

**Andreas Rauber
Stavros Christodoulakis
A Min Tjoa (Eds.)**

LNCS 3652

Research and Advanced Technology for Digital Libraries

**9th European Conference, ECDL 2005
Vienna, Austria, September 2005
Proceedings**

 **Springer**

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Andreas Rauber Stavros Christodoulakis
A Min Tjoa (Eds.)

9th European Conference, ECDL 2005
Vienna, Austria, September 18-23, 2005
Proceedings

Volume Editors

Andreas Rauber
A Min Tjoa
Vienna University of Technology
Department of Software Technology and Interactive Systems
Vienna, Austria
E-mail: {rauber,amin}@ifs.tuwien.ac.at

Stavros Christodoulakis
Technical University of Crete
Laboratory of Distributed Multimedia Information Systems and Applications
Heraklion, Greece
E-mail: stavros@ced.tuc.gr

Library of Congress Control Number: 2005932345

CR Subject Classification (1998): H.3.7, H.2, H.3, H.4.3, H.5, J.7, J.1, I.7

ISSN 0302-9743
ISBN-10 3-540-28767-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28767-4 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11551362 06/3142 5 4 3 2 1 0

Preface

Since its inception in 1997, the European Conference on Research and Advanced Technology for Digital Libraries (ECDL) has come a long way, creating a strong interdisciplinary community of researchers and practitioners in the field of digital libraries. We are proud to present the proceedings of ECDL 2005, the ninth conference in this series, which, following Pisa (1997), Heraklion (1998), Paris (1999), Lisbon (2000), Darmstadt (2001), Rome (2002), Trondheim (2003), and Bath (2004), took place on September 18–23, 2005 in Vienna, Austria.

ECDL 2005 featured separate calls for paper and poster submissions, resulting in 130 full papers and 32 posters being submitted to the conference. All papers were subject to a thorough peer-review process, with an 87-person-strong Program Committee and a further 68 additional reviewers from 35 countries from basically all continents sharing the tremendous review load, producing between three and four detailed reviews per paper. Based on these, as well as on the discussion that took place during a one-week on-line PC discussion phase, 41 papers were finally selected for inclusion in the conference program during a 1.5 day PC meeting, resulting in an acceptance rate of only 32%. Furthermore, 17 paper submissions were accepted for poster presentations with an additional 13 posters being accepted based on a simplified review process of 2–3 reviews per poster from the poster submission track. Both the full papers as well as extended abstracts of the posters presented at ECDL 2005 are provided in these proceedings.

The dense program of ECDL started on Sunday with a range of tutorials providing in-depth coverage of both introductory as well as advanced topics of digital libraries. These included a tutorial on Context-Enhanced Digital Libraries by Erich Neuhold, Claudia Niederee and Avare Stewart; Thesauri and Ontologies by Dagobert Soergel; tutorials on the creation of digital library collections with Greenstone, as well as Digital Library Interoperability Standards by Ian Witten and David Bainbridge; as well as a tutorial on the 5S Digital Library Framework by Ed Fox and Marcos Andre Goncalves.

The main conference featured two keynotes. Neil Beagrie from the British Library raised a series of challenges with respect to the growing importance of personal digital libraries, stemming from the increasingly empowered landscape of personal collections and digital memory. In the second keynote, Erich Neuhold discussed the promises and challenges of next-generation digital library architectures, analyzing the move from centralized systems to flexible content and service federations. Scientific presentations were held in two parallel sessions, interleaved with the poster sessions as well as two panel sessions addressing highly controversial topics. Tamara Sumner chaired a panel discussing the contributions that digital library technology can make to the changing nature of science and

scientific data curation. Donatella Castelli and Yannis Ioannidis raised a debate on the promises and fallacies of moving Digital Libraries onto the GRID.

Following the main conference ECDL hosted five workshops, including the long-standing CLEF Workshop of the Cross-Language Evaluation Forum, a major event on its own that was running an intensive 3-day program, as well as the fifth edition of the ECDL Workshop on Web Archiving and Digital Preservation (IWAWE 2005), the 4th NKOS Workshop on Mapping Knowledge Organization Systems (NKOS 2005), the 3rd Workshop on Digital Libraries in Health Care (HDL 2005), as well as the 1st Workshop on Knowledge Extraction and Deployment for Digital Libraries and Repositories (KED 2005)

All information relating to ECDL 2005 is available from the conference homepage at <http://www.ecdl2005.org>.

We would like to take the opportunity to thank everybody involved in making ECDL 2005 such a marvellous event. Specifically, we would like to thank all conference participants and presenters, who provided a fascinating 1-week program of high-quality presentations and intensive discussions, as well as all members from the Program Committee and the additional reviewers, who went to great length to ensure the high quality of this conference. Furthermore, we would like to thank all members of the Organizing Committee, and particularly everybody in the local organizing teams at the Austrian National Library, the Austrian Computer Society, and the Vienna University of Technology. Particularly, we would like to thank Andreas Pesenhofer for running and thoroughly monitoring the Conference Management System, Georg Pözlbauer for his diligent effort in compiling the proceedings, Thomas Lidy, Rudolf Mayer, Carl Rauch and Stephan Strodl, who, together with numerous student volunteers, assisted in all stages of organizing the conference. They all invested tremendous efforts to make sure that ECDL 2005 became an exciting and enjoyable event. It is due to them that the organization of ECDL 2005 was not just hard work, but also a pleasure.

June 2005

Andreas Rauber, Stavros Christodoulakis

Organization

Organization Committee

General chair

A Min Tjoa Vienna University of Technology, Austria

Program Chairs

Andreas Rauber Vienna University of Technology, Austria
Stavros Christodoulakis Technical University of Crete, Greece

Poster and Demo Chairs

Giuseppe Amato Consiglio Nazionale delle Ricerche, Italy
Pavel Zezula Masaryk University, Brno, Czech Republic

Workshop Chairs

Fabio Crestani University of Strathclyde, UK
Dieter Merkl University of Western Sydney, Australia

Panel Chairs

László Kovács Hungarian Academy of Sciences, Hungary
Ed Fox Virginia Tech University, USA

Tutorial Chairs

Nozha Boujemaa INRIA, France
Shin'ichi Satoh National Institute of Informatics, Japan

Best Paper Award Chair

Erich Neuhold Fraunhofer – IPSI, Germany

Doctoral Consortium Chair

Jose Borbinha INESC-ID – Information System Group, Portugal

Publicity Chairs

Julien Masanes Bibliothèque nationale de France, France
Michael Bauer Technische Universität München, Germany

Local Organizing Chairs

Eugen Muehlvenzl Austrian Computer Society, Austria
Max Kaiser Austrian National Library, Austria
Alexander Schatten Vienna University of Technology, Austria
Carl Rauch Vienna University of Technology, Austria

Program Committee

Maristella Agosti	University of Padua, Italy
Helena Ahonen-Myka	University of Helsinki, Finland
Hanne Albrechtsen	Risø National Laboratory, Denmark
Anders Andrö	Lund University, Sweden
Margherita Antona	Foundation for Research and Technology, Greece
Daniel E. Atkins	University of Michigan, USA
Ricardo Baeza-Yates	University Pompeu Fabra, Spain, University of Chile, Chile
Miroslav Bartošek	Masaryk University, Czech Republic
Alejandro Bia	Universidad Miguel Hernández, Spain
Mária Bielíková	Slovak University of Technology in Bratislava, Slovakia
Alberto Del Bimbo	Università degli Studi di Firenze, Italy
Ann Blandford	University College London, UK
Davide Bolchini	Università della Svizzera Italiana, Switzerland
Christine L. Borgman	University of California, USA
Pavel I. Braslavsky	Russian Academy of Sciences, Russia
Christian Breiteneder	Vienna University of Technology, Austria
Gerhard Budin	University of Vienna, Austria
Tiziana Cartaci	University of Roma “La Sapienza”, Italy
Donatella Castelli	Consiglio Nazionale delle Ricerche, Italy
A. Enis Cetin	Bilkent University, Turkey
Lee-Feng Chien	Academia Sinica, Taiwan
Key-Sun Choi	Korea Advanced Institute of Science and Technology, Korea
Birte Christensen-Dalsgaard	State and University Library, Denmark
Anita Coleman	University of Arizona, USA
Michel Crucianu	INRIA, France
Sally Jo Cunningham	University of Waikato, New Zealand
Michael Day	University of Bath, UK
Michael Dittenbach	E-Commerce Competence Center – EC3, Austria
Boris Dobrov	Moscow State University, Russia
Martin Doerr	Foundation for Research & Technology, Greece
Sandor Dominich	University of Veszprem, Hungary
Matthew Dovey	University of Oxford, UK
J. Stephen Downie	University of Illinois at Urbana-Champaign, USA
Cédric Dumas	Ecole des Mines de Nantes, France
Jan Engelen	Katholieke Universiteit Leuven, Belgium
Dieter Fellner	Technische Universität Braunschweig, Germany
Pablo de la Fuente	Universidad de Valladolid, Spain
Norbert Fuhr	University of Duisburg-Essen, Germany

Julio Gonzalo	Universidad Nacional de Educación a Distancia, Spain
Juha Hakala	Helsinki University Library, Finland
Allan Hanbury	Vienna University of Technology, Austria
Jane Hunter	University of Queensland, Australia
Stephen Katz	Food and Agriculture Organization of the United Nations, Italy
Wolfgang Klas	University of Vienna, Austria
Traugott Koch	UKOLN, UK
Harald Krottmaier	Graz University of Technology, Austria
Josef Küng	Johannes Kepler Universität Linz, Austria
Carl Lagoze	Cornell University, USA
Mounia Lalmas	Queen Mary University London, UK
Ray R. Larson	University of California, Berkeley, USA
Ee-Peng Lim	Nanyang Technological University, Singapore
Liz Lyon	University of Bath, UK
Gary Marchionini	University of North Carolina at Chapel Hill, USA
Vladmimír Marík	Czech Technical University, Czech Republic
András Micsik	Hungarian Academy of Sciences, Hungary
Dunja Mladenic	Josef Stefan Institute, Slovenia
Günter Mühlberger	University of Innsbruck, Austria
Marc Nanard	Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, France
Igor Nekrestyanov	Saint Petersburg State University, Russia
Heike Neuroth	Georg-August Universität Göttingen, Germany
Liddy Nevile	La Trobe University, Australia
Elias Pampalk	Austrian Research Institute for Artificial Intelligence, Austria
Christos Papatheodorou	Ionian University, Greece
Jan Paralic	University of Technology Kosice, Slovakia
Thomas Risse	IPSI Fraunhofer, Germany
Seamus Ross	University of Glasgow, UK
Nieves Rodríguez Brisaboa	Universidad de A Coruña, Spain
Lloyd Rutledge	Centrum voor Wiskunde en Informatica, Netherlands
Rudi Schmiede	Technische Universität Darmstadt, Germany
Heiko Schuldt	University of Health Sciences, Medical Informatics and Technology, Austria
Timos Sellis	National Technical University of Athens, Greece
Mário J. Gaspar da Silva	Universidade de Lisboa, Portugal
Edleno Silva de Moura	Universidade do Amazonas, Brazil
Simeon Simoff	University of Technology Sydney, Australia
Maria Sliwinska	International Center for Information Management Systems and Services, Poland

Dagobert Soergel	University of Maryland, USA
Ingeborg T. Sølvsberg	Norwegian University of Technology and Science, Norway
Nicolas Spyrtos	Université de Paris-Sud, France
Jela Steinerová	Comenius University in Bratislava, Slovakia
Shigeo Sugimoto	University of Tsukuba, Japan
Hussein Suleman	University of Cape Town, South Africa
Shalini R. Urs	University of Mysore, India
Felisa M. Verdejo	Universidad Nacional de Educación a Distancia, Spain
Paula Viana	Instituto Politécnico do Porto, Portugal
James Z. Wang	Pennsylvania State University, USA
Gerhard Weikum	Max-Planck-Institut für Informatik, Germany
Ian Witten	University of Waikato, New Zealand

Additional Reviewers

Trond Aalberg	Giorg Maria Di Nunzio	Kristina Machova
Michail Ageev	Dulce Domingos	Hubert Mara
Enrique Amigo	Georges Dupret	Mari Carmen Marcos
Pedro Antunes	Horst Eidenberger	Bruno Martins
Michela Bacchin	Kateryna Falkovych	Olena Medelyan
David Bainbridge	Renato Ferreira	Massimo Melucci
Maria Bruna Baldacci	Nicola Ferro	Dave Nichols
Peter Bednar	Gudrun Fischer	Blaz Novak
Helmut Berger	Blaz Fortuna	Ashish Parulekar
Stefano Berretti	Ingo Frommholz	Anselmo Penas
Stefano Bocconi	Silvia Gabrielli	Dimitris Plexousakis
Leonid Boitsov	Manolis Gergatsoulis	Carl Rauch
Janez Brank	Joost Geurts	Timothy Read
Peter Butka	Francois Goasdoue	Covadonga Rodrigo
Andrea Cali	Miha Grcar	Martin Sarnovsky
Pavel Calado	Maxim Gubin	Tarkan Sevilmis
Liliana Calderon	Nuno M. Guimaraes	Manuele Simi
Leonardo Candela	Mohand-Said Hacid	Jaroslav Susol
Luis Carrico	Bernhard Haslhofer	Giannis Tsakonas
Joao M.B. Cavalcanti	Robert Hecht	Torsten Ullrich
Marcirio Chaves	Aleks Jakulin	Miha Vuk
Juan Cigarran	Kevin Keenoy	Andreas Wichert
Altigran Soares Da Silva	Stefan Leitich	

Table of Contents

Digital Library Models and Architectures

Requirements Gathering and Modeling of Domain-Specific Digital Libraries with the 5S Framework: An Archaeological Case Study with ETANA

Rao Shen, Marcos André Gonçalves, Weiguo Fan, Edward Fox 1

On the Effective Manipulation of Digital Objects: A Prototype-Based Instantiation Approach

Kostas Saidis, George Pyrounakis, Mara Nikolaidou 13

LibraRing: An Architecture for Distributed Digital Libraries Based on DHTs

Christos Tryfonopoulos, Stratos Idreos, Manolis Koubarakis 25

Multimedia and Hypermedia Digital Libraries

Hierarchical Organization and Description of Music Collections at the Artist Level

Elias Pampalk, Arthur Flexer, Gerhard Widmer 37

A Comparison of Melodic Segmentation Techniques for Music Information Retrieval

Giovanna Neve, Nicola Orio 49

The Effect of Collection Fusion Strategies on Information Seeking Performance in Distributed Hypermedia Digital Libraries

Michail Salampasis, John Tait 57

XML

A Native XML Database Supporting Approximate Match Search

Giuseppe Amato, Franca Debole 69

XMLibrary Search: An XML Search Engine Oriented to Digital Libraries

Enrique Sánchez-Villamil, Carlos González Muñoz, Rafael C. Carrasco 81

From Legacy Documents to XML: A Conversion Framework
*Jean-Pierre Chanod, Boris Chidlovskii, Hervé Dejean,
 Olivier Fambon, Jérôme Fuselier, Thierry Jacquin,
 Jean-Luc Meunier* 92

SCOPE – A Generic Framework for XML Based Publishing
 Processes
Uwe Müller, Manuel Klatt 104

Building Digital Libraries

DAR: A Digital Assets Repository for Library Collections
Iman Saleh, Noha Adly, Magdy Nagi 116

Webservices Infrastructure for the Registration of Scientific Primary
 Data
Uwe Schindler, Jan Brase, Michael Diepenbroek 128

Incremental, Semi-automatic, Mapping-Based Integration of
 Heterogeneous Collections into Archaeological Digital Libraries:
 Megiddo Case Study
*Ananth Raghavan, Naga Srinivas Vemuri, Rao Shen,
 Marcos A. Goncalves, Weiguo Fan, Edward A. Fox* 139

Integrating Diverse Research in a Digital Library Focused on a Single
 Author
*Neal Audenaert, Richard Furuta, Eduardo Urbina, Jie Deng,
 Carlos Monroy, Rosy Sáenz, Doris Careaga* 151

User Studies

A Fluid Interface for Personal Digital Libraries
*Lance E. Good, Ashok C. Papat, William C. Janssen,
 Eric A. Bier* 162

MedioVis – A User-Centred Library Metadata Browser
*Christian Grün, Jens Gerken, Hans-Christian Jetter, Werner König,
 Harald Reiterer* 174

Effectiveness of Implicit Rating Data on Characterizing Users in
 Complex Information Systems
*Seonho Kim, Uma Murthy, Kapil Ahuja, Sandi Vasile,
 Edward A. Fox* 186

Managing Personal Documents with a Digital Library <i>Imene Jaballah, Sally Jo Cunningham, Ian H. Witten</i>	195
The Influence of the Scatter of Literature on the Use of Electronic Resources Across Disciplines: A Case Study of FinELib <i>Pertti Vakkari, Sanna Talja</i>	207
Information Seeking by Humanities Scholars <i>George Buchanan, Sally Jo Cunningham, Ann Blandford, Jon Rimmer, Claire Warwick</i>	218
ReadUp: A Widget for Reading <i>William C. Janssen</i>	230
Digital Preservation	
The DSpace Open Source Digital Asset Management System: Challenges and Opportunities <i>Robert Tansley, MacKenzie Smith, Julie Harford Walker</i>	242
File-Based Storage of Digital Objects and Constituent Datastreams: XMLtapes and Internet Archive ARC Files <i>Xiaoming Liu, Lyudmila Balakireva, Patrick Hochstenbach, Herbert Van de Sompel</i>	254
A No-Compromises Architecture for Digital Document Preservation <i>Thomas A. Phelps, P.B. Watry</i>	266
A Study into the Effect of Digitisation Projects on the Management and Stability of Historic Photograph Collections <i>Veronica Davis-Perkins, Richard Butterworth, Paul Curzon, Bob Fields</i>	278
Metadata	
Strategies for Reprocessing Aggregated Metadata <i>Muriel Foulonneau, Timothy W. Cole</i>	290
A Hybrid Declarative/Procedural Metadata Mapping Language Based on Python <i>Greg Janée, James Frew</i>	302

Using a Metadata Schema Registry in the National Digital Data Archive of Hungary
Csaba Fülöp, Gergő Kiss, László Kovács, András Micsik 314

Digital Libraries and e-Learning

Finding Appropriate Learning Objects: An Empirical Evaluation
Jehad Najjar, Joris Klerkx, Riina Vuorikari, Erik Duval 323

Managing Geography Learning Objects Using Personalized Project Spaces in G-Portal
Dion Hoe-Lian Goh, Aixin Sun, Wenbo Zong, Dan Wu, Ee-Peng Lim, Yin-Leng Theng, John Hedberg, Chew Hung Chang 336

Evaluation of the NSDL and Google for Obtaining Pedagogical Resources
Frank McCown, Johan Bollen, Michael L. Nelson 344

Policy Model for University Digital Collections
Alexandros Koulouris, Sarantos Kapidakis 356

Text Classification in Digital Libraries

Importance of HTML Structural Elements and Metadata in Automated Subject Classification
Koraljka Golub, Anders Ardö 368

DL Meets P2P – Distributed Document Retrieval Based on Classification and Content
Wolf-Tilo Balke, Wolfgang Nejdl, Wolf Siberski, Uwe Thaden 379

Automatic Detection of Survey Articles
Hidetsugu Nanba, Manabu Okumura 391

Searching

Focused Crawling Using Latent Semantic Indexing - An Application for Vertical Search Engines
George Almpanidis, Constantine Kotropoulos, Ioannis Pitas 402

Active Support for Query Formulation in Virtual Digital Libraries: A Case Study with DAFFODIL <i>André Schaefer, Matthias Jordan, Claus-Peter Klas, Norbert Fuhr</i>	414
Expression of Z39.50 Supported Search Capabilities by Applying Formal Descriptions <i>Michalis Sfakakis, Sarantos Kapidakis</i>	426

Text Digital Libraries

A Comparison of On-line Computer Science Citation Databases <i>Vaclav Petricek, Ingemar J. Cox, Hui Han, Isaac G. Councill, C. Lee Giles</i>	438
A Multi-document Summarization System for Sociology Dissertation Abstracts: Design, Implementation and Evaluation <i>Shiyan Ou, Christopher S.G. Khoo, Dion H. Goh</i>	450
Compressing Dynamic Text Collections via Phrase-Based Coding <i>Nieves R. Brisaboa, Antonio Fariña, Gonzalo Navarro, José R. Paramá</i>	462

Panels

Does eScience Need Digital Libraries? <i>Tamara Sumner, Rachel Heery, Jane Hunter, Norbert Lossau, Michael Wright</i>	475
Digital Libraries over the Grid: Heaven or Hell? (Panel Description) <i>Donatella Castelli, Yannis Ioannidis</i>	477

Posters

Management and Sharing of Bibliographies <i>Erik Wilde, Sai Anand, Petra Zimmermann</i>	479
Legislative Digital Library: Online and Off-Line Database of Laws <i>Viorel Dumitru, Adrian Colomitchi, Eduard Budulea, Stefan Diaconescu</i>	481
DIRECT: A System for Evaluating Information Access Components of Digital Libraries <i>Giorgio Maria Di Nunzio, Nicola Ferro</i>	483

Modular Emulation as a Viable Preservation Strategy <i>Jeffrey van der Hoeven, Hilde van Wijngaarden</i>	485
Retrieving Amateur Video from a Small Collection <i>Daniela Petrelli, Dan Auld, Cathal Gurrin, Alan Smeaton</i>	487
A Flexible Framework for Content-Based Access Management for Federated Digital Libraries <i>K. Bhoopalam, K. Maly, F. McCown, R. Mukkamala, M. Zubair</i>	489
The OAI Data-Provider Registration and Validation Service <i>Simeon Warner</i>	491
An Effective Access Mechanism to Digital Interview Archives <i>Atsuhiko Takasu, Kenro Aihara</i>	493
A Semantic Structure for Digital Theses Collection Based on Domain Annotations <i>Rocío Abascal, Béatrice Rumpler, Suela Berisha-Bohé, Jean Marie Pinon</i>	496
Towards Evaluating the Impact of Ontologies on the Quality of a Digital Library Alerting System <i>Alfons Huhn, Peter Höfner, Werner Kießling</i>	498
Building Semantic Digital Libraries: Automated Ontology Linking by Associative Naïve Bayes Classifier <i>Hyunki Kim, Myung-Gil Jang, Su-Shing Chen</i>	500
Evaluation of a Collaborative Querying System <i>Lin Fu, Dion Hoe-Lian Goh, Schubert Shou-Boon Foo</i>	502
Aiding Comprehension in Electronic Books Using Contextual Information <i>Yixing Sun, David J. Harper, Stuart N.K. Watt</i>	504
An Information Foraging Tool <i>Cathal Hoare, Humphrey Sorensen</i>	507
mod_oai: An Apache Module for Metadata Harvesting <i>Michael L. Nelson, Herbert Van de Sompel, Xiaoming Liu, Terry L. Harrison, Nathan McFarland</i>	509

Using a Path-Based Thesaurus Model to Enhance a Domain-Specific Digital Library <i>Mathew J. Weaver, Lois Delcambre, Timothy Tolle, Marianne Lykke Nielsen</i>	511
Generating and Evaluating Automatic Metadata for Educational Resources <i>Elizabeth D. Liddy, Jiangping Chen, Christina M. Finneran, Anne R. Diekema, Sarah C. Harwell, Ozgur Yilmazel</i>	513
Web Service Providers: A New Role in the Open Archives Initiative? Extended Abstract <i>Manuel Llavador, José H. Canós, Marcos R.S. Borges</i>	515
DiCoMo: An Algorithm Based Method to Estimate Digitization Costs in Digital Libraries <i>Alejandro Bia, Jaime Gómez</i>	519
Adapting Kepler Framework for Enriching Institutional Repositories: An Experimental Study <i>A. Ramnishath, Francis Jayakanth, Filbert Minj, T.B. Rajashekar</i>	521
The Construction of a Chinese Rubbings Digital Library: An Attempt in Preserving and Utilizing Chinese Cultural Heritage Materials <i>Guohui Li, Michael Bailou Huang</i>	523
Policy Model for National and Academic Digital Collections <i>Alexandros Koulouris, Sarantos Kapidakis</i>	525
A Framework for Supporting Common Search Strategies in DAFFODIL <i>Sascha Kriewel, Claus-Peter Klas, Sven Frankmölle, Norbert Fuhr</i>	527
Searching Cross-Language Metadata with Automatically Structured Queries <i>Víctor Peinado, Fernando López-Ostenero, Julio Gonzalo, Felisa Verdejo</i>	529
Similarity and Duplicate Detection System for an OAI Compliant Federated Digital Library <i>Haseebulla M. Khan, Kurt Maly, Mohammad Zubair</i>	531
Sharing Academic Integrity Guidance: Working Towards a Digital Library Infrastructure <i>Samuel Leung, Karen Fill, David DiBiase, Andy Nelson</i>	533

Supporting ECDL'05 Using TCeReview <i>Andreas Pesenhofer, Helmut Berger, Andreas Rauber</i>	535
ContentE: Flexible Publication of Digitised Works with METS <i>José Borbinha, Gilberto Pedrosa, João Penas</i>	537
The UNIMARC Metadata Registry <i>José Borbinha, Hugo Manguinhas</i>	539
Developing a Computational Model of “Quality” for Educational Digital Libraries <i>Tamara Sumner, Mary Marlino, Myra Custard</i>	541
Author Index	543

Requirements Gathering and Modeling of Domain-Specific Digital Libraries with the 5S Framework: An Archaeological Case Study with ETANA

Rao Shen¹, Marcos André Gonçalves², Weiguo Fan¹, and Edward Fox¹

¹ Digital Library Research Laboratory, Virginia Tech, USA

² Department of Computer Science, Federal University of Minas Gerais (UFMG), Brazil
{rshen, mgoncalv, wfan, fox}@vt.edu

Abstract. Requirements gathering and conceptual modeling are essential for the customization of digital libraries (DLs), to help attend the needs of target communities. In this paper, we show how to apply the 5S (Streams, Structures, Spaces, Scenarios, and Societies) formal framework to support both tasks. The intuitive nature of the framework allows for easy and systematic requirements analysis, while its formal nature ensures the precision and correctness required for semi-automatic DL generation. Further, we show how 5S can help us define a domain-specific DL metamodel in the field of archaeology. Finally, an archaeological DL case study (from the ETANA project) yields informal and formal descriptions of two DL models (instances of the metamodel).

1 Introduction

The construction of any digital library (DL) involves a number of decisions covering: 1) which types of multimedia content will be supported by the DL; 2) how the stored information is organized and structured; 3) which are the target communities; and 4) which services and capabilities will be provided [3]. The process of formally assembling such decisions and representing them in a format useful for processing by a DL system involves both requirements gathering and analytical modeling or design.

Modern software engineering has encouraged the use of formal methods, with mathematically defined syntax and semantics, to support such tasks. Formal methods and frameworks can support specification of (most of the parts of) complex systems such as DLs, while also promoting rigor and correctness. This paper focuses on the application of the 5S (Streams, Structures, Spaces, Scenarios, and Societies) formal framework [2] in the support of these tasks. More specifically, we show how 5S can help us document complex requirements and can support the modeling of domain-specific digital libraries, illustrated with a case study from the field of archaeology.

The rest of this paper is organized as follows. Section 2 provides a brief background on the 5S framework. Section 3 informally discusses requirements of archaeological DLs according to 5S. Section 4 builds on the prior section to present a formal archaeological DL metamodel. Section 5 presents a two-part case study illustrating the methodology and models. Section 6 concludes the paper and outlines future work.

2 Background on the 5S Framework

In [4] we presented a formal framework for the DL field, summarized in Figure 1. We defined “minimal digital library” (defn. 24 of [4], shown at the bottom right) as the highest level concept. Figure 1 illustrates the supporting layers of definitions: mathematical foundations (e.g., graphs, sequences, functions), the 5 Ss (Streams, Structures, Spaces, Scenarios, and Societies), and key concepts of a DL (e.g., digital object, collection). Arrows represent dependencies, indicating that a concept is formally defined in terms of previously defined concepts that point to it.

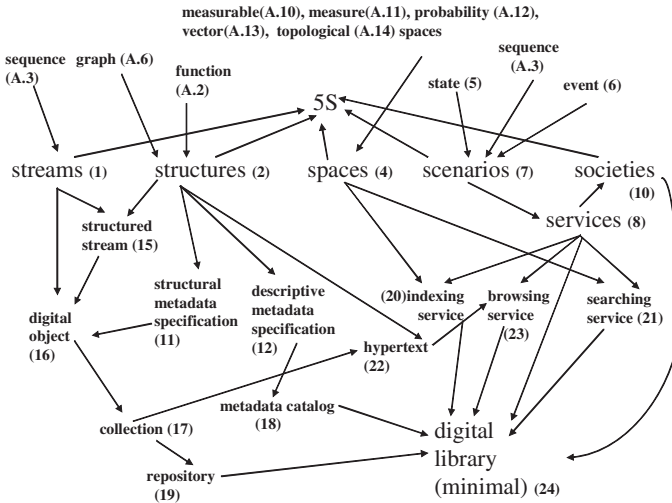


Fig. 1. 5S definitional structure (from [4])

3 Archaeological Digital Libraries: A 5S-Based Informal View

This section shows how 5S can be used to analyze the requirements of domain-specific DLs. More specifically, it informally describes the archaeological domain, and therefore archaeological DLs (ArchDLs), in the light of the 5S framework.

1. Societies

Societies can be groups of humans as well as hardware and software components. Examples of human societies in ArchDLs include archaeologists (in academic institutes, fieldwork settings, excavation units, or local / national government bodies), the general public (e.g., educators, learners), and those who lived in historic and prehistoric societies. There also are societies of project directors, field staff (responsible for the work of excavation), technical staff (e.g., photographers, technical illustrators, and their assistants), and camp staff (including camp managers, registrars, and tool stewards). Since archaeology is a multi-disciplinary subject, drawing on a wide range of skills and specialties, from the arts and humanities through to the biological and physical sciences, societies of specialists (e.g., in geology, anthropology, lithics,

ceramics, faunal and floral remains, remote sensing) are involved in ArchDLs. Societies follow certain rules and their members play particular roles. Members of societies have activities and relationships (e.g., specialists serve to assist and advise the varying field and laboratory staffs regarding field problems and other matters related to their special skills and interests). Because archaeologists in diverse countries follow different laws and customs, a number of ethical and freedom-related issues arise in connection with ArchDLs. Examples include: Who owns the finds? Where should they be preserved? What nationality and ethnicity do they represent? Who has publication rights? To address these issues, and to support the variety of needs of interested societies, DL designers have planned for numerous scenarios.

2. Scenarios

A scenario is often defined as a description of interactions between a human user and a computerized system. Scenarios also can describe interactions among software modules (as in [4]) or among humans. Further, describing scientific processes (hypothesizing, observing, recording, testing, analyzing, and drawing conclusions – used during any archaeological study) as scenarios can help with comprehending specific ArchDL phenomena, and with requirements elicitation and specification generation.

Digital recording as an archaeological process to facilitate information gathering occurs in two stages, the planning stage and the excavation stage. Remote sensing, fieldwalking, field surveys, building surveys, consulting historical and other documentary sources, and managing the sites and monuments (and related records) maintained by local and national government bodies may be involved in the planning stage. During excavation, detailed information is recorded, including for each layer of soil, and for features such as pole holes, pits, and ditches. Data about each artifact is recorded together with information about its exact find spot. Numerous environmental and other samples are taken for laboratory analysis, and the location and purpose of each is carefully recorded. Large numbers of photographs are taken, both general views of the progress of excavation and detailed shots showing the contexts of finds. Since excavation is a destructive process, this makes it imperative that the recording methods are both accurate and reliable. Unlike many other applications of information systems, it simply is not possible to go back and re-check at a later date [5]. Large quantities of archaeological data generated during the abovementioned two stages can be harvested by ArchDLs, organized, and stored to be available to researchers outside a project (site) – without substantial delay. After excavation, information stored in ArchDLs is analyzed, and helps archaeologists to test hypotheses. For example, if archaeologists retrieve records of corn artifacts from an ArchDL, they might hypothesize that the former residents were farmers, and test their hypothesis with soil sample data using statistical analysis tools provided by the ArchDL. This hypothesis is a scenario involving archaeologists, the historical community (farmers), and finds (corn samples). Other hypotheses are scenarios describing relationships among historical communities. For example, if there are large collections of jars of the same style found in two nearby sites, archaeologists might hypothesize that people in these two sites (cities) used the jars to carry things in commercial trade. Thus, primary archaeological data, managed with powerful tools in ArchDLs, help archaeologists find physical relationships between excavation contexts, develop a structural history of a site, and extend the understanding of past material cultures and environments in the

area. Data generated from the sites' interpretation then provides a basis for future work including publication, museum displays, and, in due course, input into future project planning.

Besides supporting archaeologists in their work as described above, ArchDLs provide services for the general public. A student interested in a Near Eastern site can access all the archaeological information about it by browsing or using complex retrieval criteria that take into account both intrinsic attributes of items and their extrinsic spatial and temporal interrelationships. Further, she can view the information organized in a spatial hierarchy / map that facilitates navigation among archaeological items at various spatial scales. She can click on items to show details; to display photographs, maps, diagrams, or textual documents; or to jump to other items.

3. Spaces

One important spatial aspect of ArchDLs is the geographic distribution of found artifacts, which are located in a 4D spatial continua, the fourth one being the temporal (as inferred by the archaeologists). Metric or vector spaces are used to support retrieval operations, calculate distances, and constrain searches spatially. Another space-related aspect deals with user interfaces, or with 3D models of the past.

4. Structures

Structures represent the way archaeological information is organized along several dimensions. Archaeological information is spatially organized, temporally sequenced, and highly variable. Examples include site organization, temporal order, and taxonomies of specific unearthed artifacts like bones and seeds. The structures of sites present simply and consistently the basic spatial containment relationship at every level of detail, from the broadest region of archaeological interest to the smallest aspect of an individual find. Generally, specific regions are subdivided into sites, normally administered and excavated by different groups. Each site is further subdivided into partitions, sub-partitions, and loci, the latter being the nucleus of the excavation. Materials or artifacts found in different loci are organized in containers for further reference and analysis. The locus is the elementary volume unit used for establishing archaeological relationships. Archaeological relationships between loci are from both the vertical and horizontal points of view. The first is given by reference to loci above and below a given locus, the second by coexisting loci (loci located at the same level). The archaeological relationship is related to the temporal succession of various events of construction, deposition, and destruction. Temporal sequencing of archaeological items involves linking items to form a stratigraphic diagram of the kind developed in the 1970s by Edward Harris (<http://www.harrismatrix.com/>) and now used by many archaeologists. A "Harris Matrix" is a compact diagram representing the essential stratigraphic relationships among all the items; it shows the chronological relationship between excavated layers and contexts. In general, if two layers are in contact with each other and one lies over the other, then the upper layer is chronologically later. This is the basis on which the structural history of a site is founded. The construction of this diagram and its subsequent use in the interpretation of structural phases is central to both the understanding of the site during excavation and to the post-excavation process [1]. Spatial and stratigraphic relationships among archaeological items can be regarded as extrinsic attributes (inter-item relationships) [6]; intrinsic

attributes are those describing the items themselves. Finally, since archaeological information is highly variable, items observed in a typical excavation may fall into a wide variety of different classification systems, and may exhibit many idiosyncrasies.

5. Streams

In the archaeological setting, streams represent the enormous amount of dynamic multimedia information generated in the processes of planning, excavating, analyzing, and publishing. Examples include photos and drawings of excavation sites, loci, or unearthed artifacts; audio and video recordings of excavation activities; textual reports; and 3D models used to reconstruct and visualize archaeological ruins.

4 A 5S-Based Metamodel for Archaeological Digital Libraries

With key requirements for ArchDLs summarized in the previous section, we can proceed to constructively define a minimal ArchDL metamodel. A domain-specific metamodel is a generic model which captures aspects specific to the domain at hand. We build upon the definition of a minimal DL as formally defined in [4] and extend it with concepts specific to the archaeology domain. Following our minimalist approach, we only define essential concepts without which we think a DL cannot be considered an ArchDL. The concepts and definitions are illustrated in Figure 2, where each concept is enclosed in a box labeled with the number of its formal definition (1-10 as below or starting with ‘‘A.’’ if given in [4]). The main extensions concern the fact that: 1) most archaeological digital objects are surrogates of real-world artifacts; and 2) these artifacts are found within a social-temporal-spatial context.

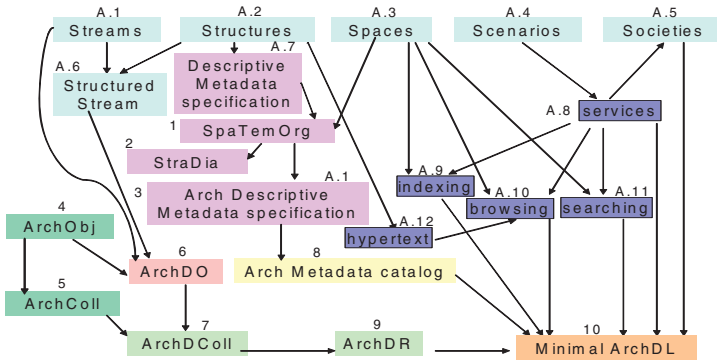


Fig. 2. Minimal archaeological DL in the 5S framework

Notation: Let $L = \cup D_k$ be a set of literals defined as the union of domains D_k of simple data types (e.g., strings, numbers, dates, etc.). Let also R represent sets of labels for resources. Let $SpaPI$ be a tree with a vertex set $\{v_i \mid i=1,2,\dots,7\}$; an edge set $E_{SpaPI} = \{(v_i, v_{i+1}) \mid i=1,2,\dots,6\}$; a labeling function $F_{edgeI}: E_{SpaPI} \rightarrow L_{SpaPI} = \{\text{‘contains’}\}$; a labeling function $F_{nodeI}: \{v_i \mid i=1,2,\dots,7\} \rightarrow V_{SpaPI} = \{F_{nodeI}(v_i) \mid i=1,2,\dots,7\}$, where $F_{nodeI}(v_1) = \text{‘region’}$, $F_{nodeI}(v_2) = \text{‘site’}$, $F_{nodeI}(v_3) = \text{‘partition’}$, $F_{nodeI}(v_4) = \text{‘sub-’}$

partition', $F_{node1}(v_5)= 'locus'$, $F_{node1}(v_6)= 'container'$, and $F_{node7}(v_7)= 'find'$. Let $SpaP2$ be a set: $SpaP2= \{ 'above', 'below', 'coexisting with' \}$. Let $Temp$ be a tree with a vertex set $\{u_1, u_2\}$; an edge set $E_{Temp}=\{(u_1, u_2)\}$; a labeling function $F_{edge2}: \{u_1, u_2\} \rightarrow L_{Temp}=\{ 'detailed by' \}$; a labeling function $F_{node2}: \{u_i \mid i=1,2\} \rightarrow V_{Temp}=\{F_{node2}(u_i)\}$, where $F_{node2}(u_1)= 'period'$, and $F_{node2}(u_2)= 'chronology'$.

Definition 1. A **Spatial Temporal Organization (SpaTemOrg)** is a descriptive meta-data specification (see def. 12 in [4] for details), $SpaTemOrg = ((V, E), RUL, F)$, such that $\forall e=(u,v) \in E$, where $u,v \in V$, $F(u) \in RUL, F(v) \in RUL, F(e) \in V_{SpaP1} \cup V_{Temp} \cup SpaP2$.

Example 1.1. Given $u, v_1, v_2 \in V, F(u)= 'Bone1', F(v_1)= 'Jordan Valley', F(v_2)= 'Nimrin', x=F((u, v_1))= 'region', y=F((u, v_2))= 'site', F_{SpaP1}((x, y))= 'contains'$, expression $('Bone1', (region: 'Jordan valley'), (site: 'Nimrin'))$ means 'Bone1' was excavated from the Jordan valley, which contains the Nimrin site (see Fig. 3).

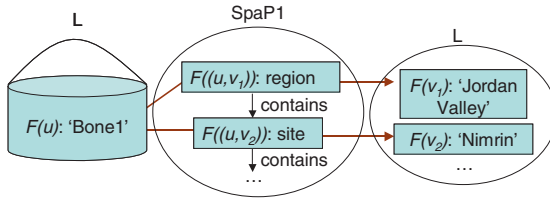


Fig. 3. Example 1.1 of *SpaTemOrg*

Example 1.2. Given $u, v_1, v_2 \in V, F(u)= 'Bone1', F(v_1)= 'Middle Bronze', F(v_2)= '2000B.C. - 1500B.C.', x=F((u, v_1))= 'period', y=F((u, v_2))= 'chronology', F_{Temp}((x, y))= 'detailed by'$, expression $('Bone1', (period: 'Middle Bronze'), (chronology: '2000B.C. - 1500B.C.))$ means 'Bone1' was excavated from a deposit made in the Middle Bronze age, which has range 2000B.C. - 1500B.C. (see Fig. 4).

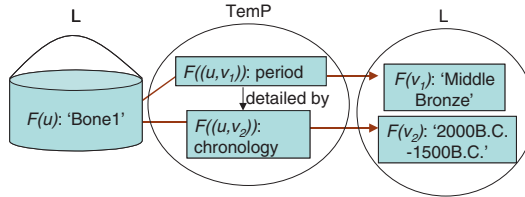


Fig. 4. Example 1.2 of *SpaTemOrg*

Definition 2. A **Stratigraphic Diagram (StraDia)** is a *SpaTemOrg* with a labeling function $F_{stra}: (V \cup E) \rightarrow L_{stra}=\{ 'contemporary with', 'later than' \}$, and two relations, \equiv and \leq , are defined on V :

1) \cong relation is: a) reflexive, b) symmetric, and c) transitive

2) \leq relation is: a) reflexive, b) anti-symmetric, and c) transitive

such that $F_{stra}(\cong) = \text{'contemporary with'}$, $F_{stra}(\leq) = \text{'later than'}$, $\forall e = (u, v) \in E$, where $u, v \in V$, $F(u) \in R \cup L$, $F(v) \in R \cup L$, and $F(e) \in L_{stra} \cup SpaP2$.

Example 2.1. Given $u, v \in V$, $F(u) = \text{'locus1'}$, and $F(v) = \text{'locus2'}$, $F_{stra}((u, v)) = F_{stra}(\leq) = \text{'later than'}$, $F((u, v)) = \text{'above'} \in SpaP2$, expression $(\text{'locus1'} \leq \text{'locus2'})$ means that *locus1* was later than *locus2*; and expression $(\text{'locus1'}, \text{'above'}, \text{'locus2'})$ means that *locus1* was above *locus2* (see Fig. 5).

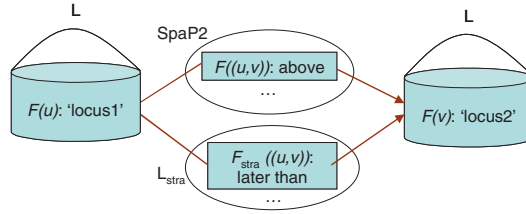


Fig. 5. Example 2.1 of *StraDia*

Definition 3. An *Archaeology Descriptive Metadata specification*: $Arch_dm \in \{SpaTemOrg\}$.

Definition 4. An *Archaeology Object in real world (ArchObj)* is a unit of observation generated by an archaeological activity (e.g., an archaeological town site, tomb, skeletal material, pottery, etc.).

Definition 5. An *Archaeology Collection (ArchColl)* is a tuple: $ArchColl = (h_{ArchColl}, \{ArchObj\})$, where $h_{ArchColl} \in H$, and H is a set of universally unique handles; $\{ArchObj\}$ is a set of archaeology objects in the real world.

Definition 6. An *Archaeology Digital Object (ArchDO)* is a tuple: $ArchDO = (h, SM, ST, StructuredStreams, SurrogateObj)$, where

- 1) $h \in H$, where H is a set of universally unique handles;
- 2) $SM = \{sm_1, sm_2, \dots, sm_n\}$ is a set of streams;
- 3) $ST = \{st_1, st_2, \dots, st_m\}$ is a set of structural metadata specifications;
- 4) $StructuredStreams = \{stsm_1, stsm_2, \dots, stsm_p\}$ is a set of *StructuredStream* functions defined from the streams in the SM set and the structures in the ST set.
- 5) $SurrogateObj$: a function $\{h\} \rightarrow \{ArchObj_1, ArchObj_2, \dots, ArchObj_k\}$ maps a handle h to an archaeology object in the real world, $SurrogateObj(h)$.

Definition 7. An *Archaeology Digital Collection (ArchDColl)* is a tuple: $ArchDColl = (h_{ArchDColl}, \{ArchDO\}, SurrogateColl)$, where $h_{ArchDColl} \in H$; H is a set of universally unique handles; $\{ArchDO\}$ is a set of archaeology digital objects with handles in H . Let $Coll \in 2^{\{SurrogateObj(h)\}}$, where h is the handle of $ArchDO$; $SurrogateColl$ is a function $\{h_{ArchDColl}\} \rightarrow Coll$ that maps handle $h_{ArchDColl}$ to a real world archaeology collection. Fig. 6 illustrates functions $SurrogateObj$ and $SurrogateColl$.

Definition 8. An *Archaeology metadata catalog* ($ArchDM_{ArchDColl}$) for an ArchDL collection $ArchDColl$ is a set of pairs $\{(h, \{Arch_dm_1, Arch_dm_2, \dots, Arch_dm_i\})\}$, where $h \in H$ and each $Arch_dm_i$ is an archaeology descriptive metadata specification.

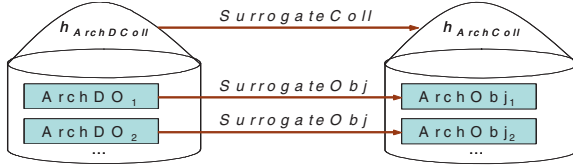


Fig. 6. Functions *SurrogateObj* and *SurrogateColl*

Definition 9. Let $DColl = \{ArchDO_1, ArchDO_2, \dots, ArchDO_k\}$ with k handles in H . An *Archaeology Digital Repository* ($ArchDR$) is a tuple: $ArchDR = (R, get, store, del)$, where $R \subseteq 2^{DColl}$, and “get”, “store”, and “del” are functions over the repository (see def. 19 in [4] for details on these functions).

Definition 10. An *Archaeological Digital Library* ($ArchDL$) is a tuple: $ArchDL = (ArchDR, ArchDM, Serv, Soc)$, where

- 1) $ArchDR$ is an archaeology digital repository;
- 2) $ArchDM = \{ArchDM_{ArchDColl_1}, ArchDM_{ArchDColl_2}, \dots, ArchDM_{ArchDColl_k}\}$ is a set of archaeology metadata catalogs for all archaeology digital collections $\{ArchDColl_1, ArchDColl_2, \dots, ArchDColl_k\}$ in the repository;
- 3) $Serv$ is a set of services containing at least indexing, searching, and browsing;
- 4) $Soc = (SM \cup Ac, R)$, where SM is a set of service managers responsible for running DL services, $Ac \subseteq \{Archeologist, GeneralPublic\}$ is a set of actors that use those services, and R is a set of relationships among SM and Ac .

5 Case Studies: Application of 5S to Archaeological DLs

In the last two sections, 5S was used to provide both an informal and a formal ArchDL model. In this section we use two archaeological information systems of ETANA projects (<http://www.etana.org/>) as case studies to 1) show the use of 5S as an analytical tool helpful in comprehending specific ArchDL phenomena; and 2) illustrate the use of 5S for requirements gathering and modeling in ArchDL development. Data contributed by ETANA projects to ETANA-DL (<http://feathers.dlib.vt.edu>) are described at http://feathers.dlib.vt.edu:8080/etana/htmlPages/etanadl_collections.htm.

5.1 Virtual Nimrin

Tell Nimrin (TN) is an archaeological town site at Shuna South, Jordan, north of the Dead Sea, in the Jordan Valley. The digital presentation of TN, virtual Nimrin (VN, <http://www.cwru.edu/affil/nimrin/>), at Case Western Reserve University, is supervised by director James W. Flanagan.

1. Societies

VN was designed for the general public as well as research specialists. Other communities addressed include: directors, core field staff (square supervisors, technical archaeologists, disciplinary specialists, assistant staff, and managers), and VN website developers/viewers from a score of museums, research institutions, colleges, and universities in Jordan and the United States.

2. Scenarios

Each of the communities involved in the VN society is engaged in various tasks and activities. Core field staffs were responsible for the actual work of excavation and recording. For example, in the field, unearthed bones were bagged separately, daily, with a feature and field specimen number which could be cross referenced with associated ceramics. These bagged bones were transported to field laboratories to be dry brushed, washed when necessary, and separated into generalized categories such as large, medium, or small mammals; fish and birds, etc. To advance and enhance digital recording, digital photography and additional programming were used. Project directors pursued geological and archaeological research by analyzing the field survey and excavated record, testing hypotheses, and publishing preliminary and final reports. For example, they found there was a reduction in percentages of bones of hogs over time at Tell Nimrin, and hypothesize that the reason probably was the introduction of religious taboos against eating pork. VN website developers built systems to allow users to interact with and interpret the site without being constrained by the director's view. General users may be interested in taking the tour of the VN website and in viewing museum quality artifacts and major finds, while specialists may want to interact with or download the databases created from the field records of excavation.

3. Spaces

TN's geographical setting marks the intersection of N-S and E-W arteries in the Jordan Valley approximately 12km north of the Dead Sea and 8km from the Jordan River due east of Jericho. It is at 35°37'30" east longitude and 31°54'00" north latitude with a Palestinian grid reference of 2094E/1451N. The mound stands 12.75m high on the south bank of Wadi Nimrin, with base 200m below sea level. GPS was used in a geological survey, such as to document the regression of the ancient freshwater Lake Lisan that once filled the Jordan Valley, and to determine how the change from a freshwater to a saline body affected the pattern of ancient settlements in the region. Other space aspects of VN are TN's coordinate system (site grid and identification of squares) displayed in the topographical drawing, and VN's user interface.

4. Structures

Structures of VN include its relational database, TN's site organization, and TN's stratigraphic diagram, from which a temporal sequence was derived. Spatial and temporal description of records in the database is specified according to TN's polar point grid site organization and site chronological order. The 00/00 point was set at the highest elevation of the mound which was central to its N/S and E/W expanse as well. From there, the site was divided into quadrants, which were subdivided into 5m × 5m squares, each labeled according to the point closest to 00/00. For instance, N40/W20 identified both the point 40m north and 20m west of 00/00 and the five meter square

to the north and west of that point. Stratigraphical relationship analysis has identified eight major strata. They are: Modern (stratum VIII), Mamluk (stratum VII), Late Byzantine/Umayyad (stratum VI), Roman/Byzantine (stratum V), Persian (stratum IV), Iron II (stratum III), Iron I (stratum II), and Middle Bronze (stratum I). They clarified TN's long history as an agricultural town site and indicated TN was a substantial settlement, inhabited continuously for the past 4 millennia, except for a 500 year period.

5. Streams

VN deals with various streams, such as drawings and photos of (parts of) TN, publications of preliminary (final) reports, and tuples of primary data in the database.

Virtual Nimrin (VN) Formal Model

Virtual Nimrin is a tuple: $VN = (VN_R, VN_DM, VN_Serv, VN_Soc)$, where

- 1) VN_R is an archaeological digital repository having Tell Nimrin's digital collections of animal bones, seeds, etc. – $DCollBone, DCollSeed, \dots, DCollObj$.
- 2) $VN_DM = \{VN_DM_{DCollBone}, VN_DM_{DCollSeed}, \dots, VN_DM_{DCollObj}\}$ is a set of archaeology metadata catalogs for all archaeology digital collections in VN, where $VN_DM_{DCollObj}$ is a metadata catalog for digital collection $DCollObj$. Let $VN_dm_{DCollObj}$ be a descriptive metadata specification for digital objects in $DCollObj$. $VN_dm_{DCollObj} \in \{SpaTemOrg\} \cup \{VN_Dobj\}$, where
 - a) $\{Jordan\ Valley, Nimrin, quadrant, square, locus, bag\} \cup \{Ottoman-Modern, Islamic, Byzantine, Late\ Hellenistic-Roman, Persian, Iron\ II, Iron\ I, Middle\ Bronze\}$ $\subset L$. (See examples in Figures 3, 4 for reference.)
 - b) $VN_Dobj = ((V, E), R, F)$ is an archaeological-object-specific descriptive metadata specification. If $DCollObj$ is a digital collection of animal bones, then $\forall e = (u, v) \in E$, where $u \in V, v \in V, F(u) \in R, F(v) \in L$, and $F(e) \in \{boneName, animalName\}$.
- 3) $VN_Serv = \{browsing, searching\}$
- 4) $VN_Soc = (\{VN_ServiceManager\} \cup \{VN_Actor\}, R) \cup \{HistoricNimrinResidence\} \cup \{PrehistoricNimrinResidence\}, R)$,
 where $\{director, fieldStaff, specialist, student, \dots\} \subset \{VN_Actor\}$,
 $\{VN_BrowseManager, VN_SearchManager\} \subset \{VN_ServiceManager\}$,
 $\{browse = (student \times VN_BrowseManager, browsing), search = (specialist \times VN_SearchManager, searching), guide = (director \times fieldStaff, \Phi)\} \subset R$

5.2 Halif DigMaster

The site of Tell Halif, located in southern Israel, is the focus of the Lahav Research Project. Halif DigMaster (HD) is an online archaeological database that offers access to a collection of Persian/Classical (and some Iron II Age) figurines recovered in excavation from Tell Halif (TH).

1. Societies

HD was developed to disseminate archaeological information to the academy and to the public. Societies of HD include the communities who excavated the figurines

from Tell Halif, provided HD with a preliminary presentation of the excavated material, collaborated with HD on resource sharing, or developed and/or employed HD.

2. Scenarios

Scenarios of HD can be those related to activities such early publication, collaborative publication, 3D publication, and other usage scenarios of HD. The overlong lag between discovery and publication is an embarrassment for archaeology of the ancient Near East. To address this embarrassment, staff of the Lahav Research Project made graphic forms of excavated objects available to the scholarly and professional world prior to final publication, while field work was still continuing. Scenarios of inviting excavators with non-public or incompletely published collections to add their materials to HD allow several excavation teams to share resources. Scenarios of using 3D technology such VRML (Virtual Reality Modeling Language) overcome the limitations of “flat” photographs on screen. Usage scenarios of HD describe services such as browsing and searching the HD database.

3. Spaces

Major spaces in HD are the geographical setting of Tell Halif (located at 34°52' east longitude and 31°23' north latitude, at coordinates 1373/0879 on the Palestinian grid) and HD's user interface. Another space is of the VRML models of artifacts.

4. Structures

Structures in HD include the relational database, Tell Halif's site organization, and TH's strata relationship. TH has been surveyed and plotted in relationship to the standard regional grid. Each of the major sections being worked is called a “field”, which is further divided into a number of more or less standard 5m×5m areas. TH consists of seventeen major occupation strata, one built atop another to a depth of more than six meters. Those strata revealed that TH has a history of occupation began in the Chalcolithic era (3500B.C. – 3200B.C.) down to the modern settlement of Kibbutz Lahav (founded in 1963 A.D.).

5. Streams

Streams in HD are full-scale color photographs, ink drawings, QuickTime VR movies, plans and soil profiles, as well as summary reports for excavation units.

Halif DigMaster (HD) Formal Model

Halif DigMaster is a tuple: $HD = (HD_R, HD_DM, HD_Serv, HD_Soc)$, where

- 1) HD_R is an archaeological digital repository having Tell Halif's digital collection of figurines, denoted as $DCollFig$.
- 2) $HD_DM = \{HD_DM_{DCollFig}\}$ is a set of archaeology metadata catalogs for ArchDL collections of figurines, where $HD_DM_{DCollFig}$ is a metadata catalog for digital collection $DCollFig$. Let $HD_dm_{DCollFig}$ be a descriptive metadata specification for digital objects in $DCollFig$. $HD_dm_{DCollFig} \in \{SpaTemOrg\}$, where $\{‘Southern Israel’, ‘Halif’, ‘field’, ‘area’, ‘locus’, ‘basket’\} \cup \{‘Persian’\} \subset L$.
- 3) $HD_Serv = \{browsing, searching\}$
- 4) $HD_Soc = (\{HD_ServiceManager\} \cup \{HD_Actor\} \cup \{PersianHalif\}, R)$, where $\{director, fieldStaff, specialist, student\} \subset \{HD_Actor\}$, $\{HD_BrowseManager,$

$HD_SearchManager\} \subset \{HD_ServiceManager\}$. We denote the community that made the Persian figurines excavated from Tell Halif as *FigMaker*, and denote the persons who those figurines represent (as surrogates) as *FigSurrogate*. Then $\{FigMaker, FigSurrogate\} \subset \{PersianHalif\}$; $\{browse = (student \times HD_BrowseManager, browsing), search=(director \times HD_SearchManager, searching), describe=(specialist \times FigSurrogate, \Phi)\} \subset R$.

6 Conclusion

DLs and archaeology have inherently interdisciplinary natures. This makes an ArchDL an even more complex information system and the task of formally defining it difficult. In this paper, we address this problem, defining a minimal ArchDL by applying and extending a DL formal framework – 5S. Our definition serves as the foundation for our enhanced ETANA-DL (<http://feathers.dlib.vt.edu>) prototype, now being refined so as to result from semi-automatic DL generation. Future work will include modeling distributed DLs, possibly including P2P approaches as in OCKHAM (<http://www.ockham.org/>), and developing assessment measurements for domain specific integrated DLs such as ArchDL.

Acknowledgements. This work is funded in part by the National Science Foundation (ITR-0325579). Marcos Gonçalves was supported by an AOL fellowship and by CNPq. We thank Douglas Clark, Joanne Eustis, James W. Flanagan, Paul Gherman, Andrew Graham, Larry Herr, Paul Jacobs, Douglas Knight, Oystein LaBianca, David McCreery, and Randall Younker for their support. We also thank all our colleagues in the Digital Library Research Laboratory at Virginia Tech.

References

1. Finkelstein, S., Ussishkin, D. and Halpern, B. Monograph Series of the Institute of Archaeology, Tel Aviv University, 2000.
2. Gonçalves, M.A. Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications, PhD Dissertation, Virginia Tech, 2004.
3. Gonçalves, M.A. and Fox, E.A., 5SL: a language for declarative specification and generation of digital libraries. In Proc. JCDL 2002, 263-272.
4. Gonçalves, M.A., Fox, E.A., Watson, L.T. and Kipp, N.A. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. ACM TOIS 22(2):270-312.
5. Ryan, D.N. Managing Complexity: Archaeological Information Systems Past, Present and Future. <http://www.cs.kent.ac.uk/people/staff/nsr/arch/baas.html>
6. Schloen, J.D. Archaeological Data Models and Web Publication Using XML. Computers and the Humanities, 35 (2): 123-152.

On the Effective Manipulation of Digital Objects: A Prototype-Based Instantiation Approach

Kostas Saidis, George Pyrounakis, and Mara Nikolaidou

Libraries Computer Center,
Department of Informatics and Telecommunications,
University of Athens,
University Campus, Athens, 157 84, Greece
saiko@di.uoa.gr, forky@libadm.uoa.gr, mara@di.uoa.gr

Abstract. This paper elaborates on the design and development of an effective digital object manipulation mechanism that facilitates the generation of configurable Digital Library application logic, as expressed by collection manager, cataloguing and browsing modules. Our work aims to resolve the issue that digital objects typing information can be currently utilized only by humans as a guide and not by programs as a digital object type conformance mechanism. Drawing on the notions of the Object Oriented Model, we propose a “type checking” mechanism that automates the conformance of digital objects to their type definitions, named *digital object prototypes*. We pinpoint the practical benefits gained by our approach in the development of the University of Athens Digital Library, in terms of code reuse and configuration capabilities.

1 Introduction

In the context of Digital Libraries, a digital object can be conceived as a human generated artifact that encapsulates underlying digital content and related information [7,13]. Although variations on representation and encoding issues may exist, this information is used to describe, annotate, link and manipulate the object’s digital content. However, the term “object” is used in a far richer context in the field of Software Engineering: in the Object Oriented (OO) model an object acts as the container of both data (its state) and behavior (its functionality) and conforms to a type definition, named class.

The Metadata Encoding and Transmission Standard (METS) [11] refers to digital objects in terms of XML documents, providing detailed specifications of its sections, comprised of descriptive and administrative metadata, files, structural maps and links. Moreover, METS supports behaviors attached on digital content, through the *Behavior* section of the METS object, “that can be used to associate executable behaviors with content” [11]. Even though METS uses the notion of Profiles to refer to “classes” of digital objects, a METS document’s Profile [10] is “intended to describe a class of METS documents in sufficient detail to

provide both document authors and programmers the guidance they require to create and process METS documents conforming with a particular profile". The absence of an effective type-conformance mechanism forces (a) programmers to write custom code to implement digital object manipulation mechanisms in an ad-hoc and not reusable fashion and (b) cataloguing staff to carry out all the digital object typing arrangements "by hand", since DL system is not able to perform them automatically in a transparent manner.

Our work, presented in this paper, refers to the design and implementation of a sufficient digital object manipulation mechanism, that facilitates the conformance of digital objects to their type definitions, named *digital object prototypes*, in an automated manner. Section 2 provides a thorough discussion on digital object manipulation requirements of University of Athens Digital Library (UoA DL) in detail, also presenting UoA DL architecture. Section 3 introduces the notion of digital object prototypes, used as a means to express type-dependent customisations on the overall structure and behavior of digital objects. Section 4 demonstrates the benefits of utilising prototypes in the development of UoA DL modules, especially for the case of cataloguing and browsing interfaces and Section 5 clarifies the use of prototypes in the context of various collections, by setting up scopes of prototypes. Finally, discussion on related and future work resides in section 6.

2 The University of Athens Digital Library

2.1 Motivation

In [15] we presented a high-level overview of the University of Athens Digital Library (UoA DL) architecture. UoA DL will host several heterogeneous collections in terms of content type, structure, metadata and user requirements, containing both digitised and born digital material. Some of them are: the University's Historical Archive, Theatrical collection, Folklore collection [9], Byzantine Music Manuscripts collection, Medical collection and Ancient Manuscript collection. UoA DL System is implemented in Java and uses Fedora [18] as a digital object repository.

The Libraries Computer Center of the University is responsible for both UoA DL System development and the management of digitisation and cataloguing processes of the aforementioned collections, under a strict period of time. Under these conditions, we did not consider viable to develop custom functionality for each collection, in terms of metadata handling, user interfaces or any other modules. On the contrary, our design approach has been based on the concept of reusing configurable functionality, in order to cope with the various constraints and requirements imposed by each collection. Thus, we focused on the development of a general-purpose, parameterisable DL System that should be easily configured to accommodate to each collection's specific requirements, exhibiting code reuse.

Our primary focus has been given to collection management, cataloguing, browsing and user interface issues. Most collections consist of heterogeneous

digitised material that require detailed cataloguing, given that free-text search facilities cannot be provided. Thus, it is of great significance to support detailed descriptions of the nature and structure of digital content in a configurable manner. Moreover, the majority of people participating in digitisation and cataloguing process will be active members of the Academic Community (such as scholars and postgraduate students) with expertise on the field of the specific collection. In order to increase productivity and facilitate cataloguing, while achieving configurability and code reuse, we focused on generating a unified configurable cataloguing interface, that should adapt to each collection's idiosyncrasies, while hiding from users internal representation, implementation and storage details. The system should "hide" underlying notions such as datastreams, XML documents, or even the use of specific metadata sets and only if the cataloguer requires a more technical cataloguing interface, the system should disclose such "internal" details.

2.2 Fedora Repository

In [15] we describe the reasons we have chosen Fedora for the development of UoA DL. Specifically, we use Fedora Repository for handling concerns related to storage, preservation and versioning, searching and indexing, along with metadata interoperability through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [8].

Within Fedora framework, digital objects are comprised of datastreams, used to store all digital object related information. Fedora imposes no restrictions on datastream content; it may be XML metadata or arbitrary external content, such as locally stored files or remote URLs. Fedora Digital Object Model (FDOM) has been based on a METS variant in 1.x versions. In Fedora 2.0, released on late January 2005, a Fedora-specific representation of digital objects is introduced, named Fedora Object XML (FOXML) [5]. The concept behind METS *Behavior* section is implemented in FDOM in terms of *disseminators*. Disseminators associate datastreams to specific behaviors, through the use of special digital objects, namely *Behavior Definition Objects* and *Behavior Mechanism Objects*. Fedora Behaviors provide one or more methods that get associated to selected datastreams of a digital object and are automatically exposed in terms of Web Services [19], providing a standard-based, service-oriented mechanism for generating distributed and interoperable systems.

2.3 Digital Object Manipulation Requirements

In a higher level of abstraction, a digital object refers to a human generated artifact that encapsulates underlying digital content and related information [7,13]. Although variations may exist on serialisation and encoding issues (METS [11], FOXML [5], RDF [16]), this information is used to describe, annotate, link and manipulate the object's internal content. Under this perspective, a digital object is conceived as an aggregation, consisting of four parts: its metadata sets, files, structure and behaviors, as presented in Figure 1.

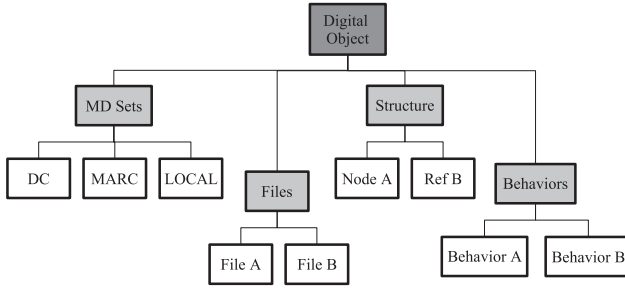


Fig. 1. An abstract representation of a Digital Object and its constitutional parts

In this context, an effective digital object manipulation mechanism should provide the following capabilities:

(a) *With regard to Metadata Sets:* (i) Allow the use of multiple Metadata sets to characterise digital objects, (ii) support mappings between fields of different Metadata Sets, in order to minimise redundancy and (iii) allow the localisation and customisation of each Metadata Set in order to cope with the special needs of different kinds of material, since, in practice, no single metadata standard can cover all possible needs, especially in the case of digitised content or digital culture content.

(b) *With regard to Files:* A digital object may contain zero or more files. File existence should not be mandatory, since a digital object could be used as a means to express the structure of "real world" objects.

(c) *With regard to Structural Information:* The representation of both structural and general-purpose linking information should be easily expressed.

(d) *With regard to Behaviors:* Facilitate DL modules and services to effectively manipulate and compose digital objects.

The detailed specification of each of these attributes depend on the digital object's nature; that is, the object's type. Consider, for example, the Theatrical collection consisting of albums that contain photos from theatrical performances of the National Theater. All Photo digital objects should behave in the same manner, *being themselves aware* about which parts comprise them and what each one represents. However, this fundamental "is-a" information is not properly expressed in neither METS nor FDOM. Fedora supplies digital objects with a *Content Model*, resembling the METS *Profile* metadata attribute. METS Profiles provide descriptions of "classes" of digital objects in order to be used by humans as a guide and not by programs as an actual type checking mechanism. In essence, digital objects are practically "typeless", since the knowledge of the types of objects is utilisable only by humans and not by the DL system. The absence of automatic type checking forces (a) programmers to write custom code to implement digital object type-conformance in an ad-hoc and not reusable fashion and (b) cataloguing staff to carry out all the digital object typing arrangements "by hand", since the system is not able to perform them automatically in a transparent manner.

According to the OO paradigm, each object corresponds to a type definition, named class. The same should stand for digital objects as well; digital objects should conform to a specification of their constitutional parts, wrapped in a separate entity. This entity, named *digital object prototype*, should contain all the corresponding specifications, in a manner independent of their realizations, such as class in the OO model provides the specification of its instantiations. Under this perspective, behaviors should not be assigned on digital objects directly; they should be assigned on the definition of their type. In the OO paradigm, functionality expressed in terms of methods is defined *once and in one place*, in the class definition. Objects, defined as class instances, are automatically supplied with this functionality as a result of their “is-a / instance-of” relation with their type definition; it is the class (that is, the type) that makes this functionality available to all its instances, it is not the user that provides it to each of them separately.

2.4 UoA DL System Architecture

From a software design perspective, the digital object type checking issues can be explained using the notion of separation of concerns [14]. There are three separate concerns that need to be resolved in order to provide an efficient digital object manipulation mechanism: (a) how the object is stored (storage and serialization concern), (b) what are the object parts and what each one represent (object specification and typing information), (c) how the object is internally represented, handled and composed according to its typing specification, in terms of coding facilities and APIs. Figure 2 depicts UoA DL architecture, where an intermediate layer is inserted between Fedora Repository and DL application logic, named *Digital Object Dictionary*, that copes with the typing and DL system internal representation concern of underlying Fedora digital objects.

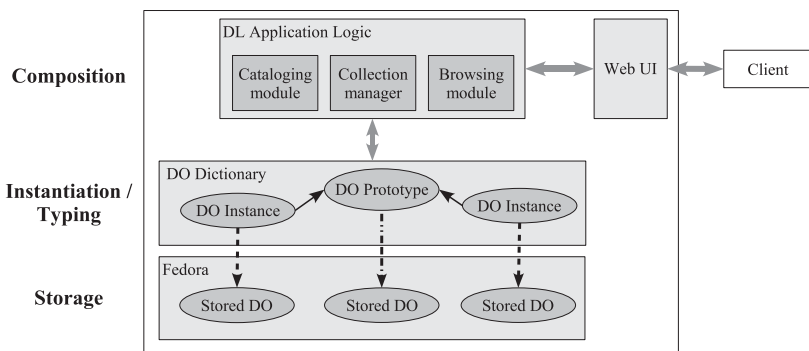


Fig. 2. Uoa DL architecture, using a Digital Object Dictionary to facilitate Compositions of Digital Objects

3 Digital Object Prototypes and Instances

A *Digital Object Prototype* provides a detailed specification of a digital object’s constitutional parts. The process of generating a digital object based on a prototype is called *instantiation* and the resulted object is called an *instance of the prototype*. The digital object instantiation process ensures that the acquired object instance will conform with the specifications residing in the prototype, making it automatically behave in the prototype-specified manner. Our implementation has focused on expressing “is-a / instance-of” relations between objects and respective prototypes, resembling the OO instantiation mechanism. Future work includes the ability to utilise OO inheritance in the context of digital object prototypes.

Figure 3 represents a Theatrical Collection Photo object as an instance of the corresponding Photo prototype. The latter is defined in terms of XML, depicted in Figure 4, providing the specifications of the constitutional parts of all the Photo digital object instances. When a Photo object instance is acquired by a DL module, the Dictionary utilises information residing in the prototype in order to load its constitutional parts from Fedora repository.

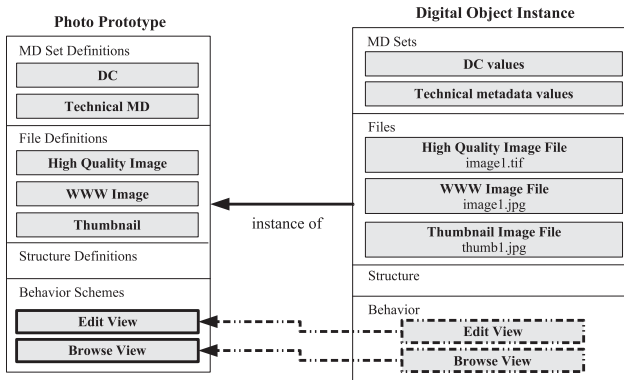


Fig. 3. A Photo Object as an Instance of the Photo Prototype

In particular, the Photo prototype specifies that:

(a) Photo object instances should contain two metadata sets, namely: (i) DC, used to hold the fields of the Dublin Core Metadata Element Set [3] for content description purposes and (ii) TechnicalMD, holding the fields of the NISO technical metadata for Digital Still Images [12] for holding digitisation information. There exist common DC and NISO fields (e.g. file format), that are mapped internally by the object instance, as specified in the Photo Prototype. Thus, even if the file format field is stored within the DC section, it is “aliased” as both a DC and NISO field by the Photo Instance. This way, field mappings are automatically handled by object’s type, such as a class in the OO paradigm uses encapsulation and information hiding to enclose “internal / private” functionality.

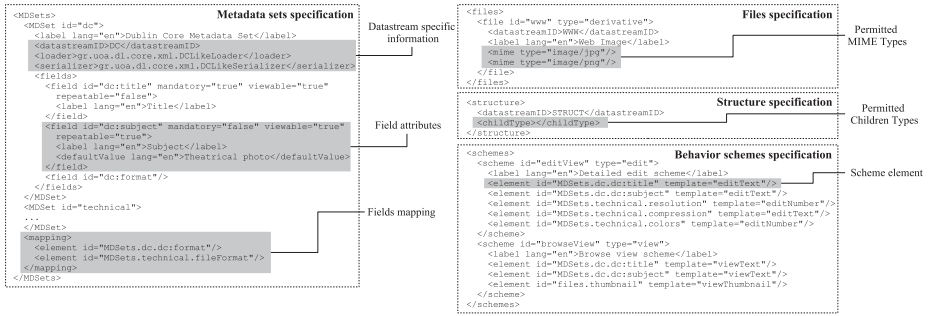


Fig. 4. Photo Prototype XML Definition

(b) Photo instances should contain a high quality TIFF image, a WWW quality JPEG image and a thumbnail. Allowed MIME types and descriptive attributes are also provided.

(c) Photo object instances do not contain other types of objects; the Structure section of the Photo instance is empty, following the specification residing in the structure definition part of the Photo Prototype. On the other hand, Albums are used to host photos and the respective structure definition of the Album Prototype (not presented herein), specifies that Photo object instances can be “inserted” into Album instances. Structural information is utilised by user interface modules, in order to inform the users of the DL System about the allowed children types of each object instance.

(d) When a Photo object instance is created, it automatically exposes the behavior schemes defined by the Photo prototype, namely *editView* and *browseView*.

A prototype may contain various behavior schemes for different purposes. However, two basic types of behavior schemes are currently supported: *edit* and *view*. The application logic indicates if the acquired digital object instance should be used for read-only or editing purposes. In the latter case, the object instance is acquired in editing mode and the user interface adjusts accordingly, by displaying the scheme elements through an editable web form. The *editView* behavior scheme depicted in Figure 4 determines the components of a low-detail cataloguing interface, comprised of the photo’s title, subject, resolution, compression ratio and number of available colours. Respectively, the *browseView* scheme defines the metadata used for browsing photo objects, comprised of the photo’s title, subject and thumbnail image. The distinction between an “editable” view or a “read-only” view is made internally by the object’s prototype, that “knows” how to guide the low-level Web UI engine to display each element for editing or presentation purposes. The result is configurability and code reuse; the Dictionary API makes no distinction between end-user and cataloguer forms, allowing the development of modules in a uniform and reusable fashion. Cataloguer interfaces are generated in the same way with “common” content display end-user interfaces.

A behavior scheme represents a composition of the digital object constitutional parts. It resides in the prototype and it is dynamically bound to all the prototype instances. This dynamic binding resembles the notion of Fedora disseminator. However, behavior schemes are based on well-known concepts of the OO paradigm, with well understood semantics. In particular, a Behavior scheme can be conceived as a method of class operating on the latter's "internal" fields. The behavior scheme is defined once in the digital object prototype and is made available to all its corresponding instances by the typing mechanism, resembling the OO model's dynamic method dispatching. Moreover, Fedora behaviors are parameterised on the datastream level of digital objects, while Prototype-based behavior schemes can be parameterised in a more fine-grained manner, supporting arbitrary compositions of digital object atomic elements, as contained in the metadata sets, files or structural / reference parts. Behavior schemes are able to identify the required atomic elements (e.g *MDSets.dc.dc:title*, *MDSets.dc.dc:subject*, *files.thumbnail* in the `browseView` behavior scheme) independent of their storage representation or datastream location. Finally, Fedora behaviors rely on the a-priori existence of the digital object and its constitutional datastreams; it is not possible to attach behavior on a digital object or its datastreams, if they do not yet exist. This means that they are not suitable for performing cataloguing effectively; all datastreams should be present and all behaviors should be defined and bound, in order to be able to utilise them. On the contrary, our approach gathers all object-related behaviors in the prototype and thus, we are able to treat a newly created prototype instantiation in a type-defined manner, before it has been ingested in Fedora Repository. Fedora Web Service behaviors will be of value after collections and related objects have been inserted into the system and DL application logic services have been developed.

4 The Benefits of Digital Object Type Conformance

The point put forward by this paper is that digital object composition and manipulation is performed more effectively if digital objects are supplied with a high-level type conformance mechanism, that resolves internal object details in a transparent manner. Fedora satisfies several of the requirements identified in section 2.3, such as the support of many metadata sets and files, along with the effective storage and indexing of structural and reference information, added in its new version. Based on its rich feature set, we have set up the digital object "type conformance" extension, that can be conceived as a realization of the *Content Model* notion, in programming and practical terms. Figure 5 depicts this functionality, performed by the intermediate Dictionary layer.

The use of digital object prototypes allowed us:

- to gather all digital object manipulation functionality in a unified programming API, exposed in terms of the digital object dictionary, that abstracts underlying serialisation and storage details of digital content. This allowed us to centralise and reuse the code that handles datastream handling and XML parsing concerns, advancing the overall system's modularity and increasing the

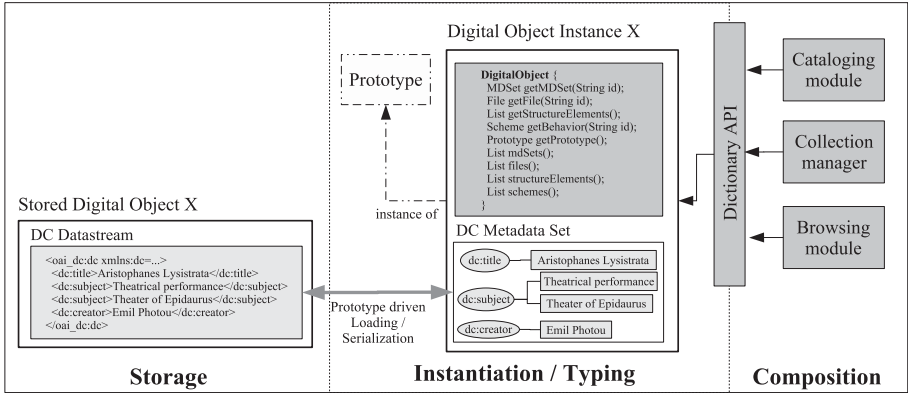


Fig. 5. A detailed overview of our proposed “type conformance” mechanism

independence of DL System modules from low-level Fedora datastream serialisation details.

- to customise digital object instances “internally” and automatically, using prototype-driven digital object instantiation. The concept of OO encapsulation is based on entities carrying out “private” functionality in a type-specific manner (e.g. metadata field mappings, validation of containment relationships and allowed files), while exposing unified external manipulation interfaces. The notion of prototype-based *behavior schemes* aims at accomplishing this functionality – each behavior scheme is executed differently by each object, albeit used in a unified manner. For instance, Album instances are also supplied with *editView* and *browseView* schemes that are executed by the Album prototype in a different manner with regard to the corresponding schemes of the Photo prototype.

- to enable type-defined introspection of digital object structure and behavior. A UoA DL module is capable of querying a digital object instance for its constitutional parts, made available to the instance through its corresponding prototype. This way, a module can supply the user with the list of supported behaviors and let him or her select the desired behavior scheme to execute. In software engineering, this concept is named *reflection*. Work presented in [4] enables introspection of underlying object structure and behaviors. A prototype can be conceived as an introspection guide for its digital object instances and we argue that prototype-driven introspection is richer in semantics terms.

5 Digital Collections and the Scope of Prototypes

A collection refers to a set of items with similar characteristics, such as type, owner, subject area and like. A digital collection aims at providing this grouping in the context of digital content. In the case of the University of Athens, most digital collections contain digitised material representing real world complex objects, as the Photo Albums of Theatrical collection. Furthermore, various metadata sets and/or local extensions or combinations of them are used to char-

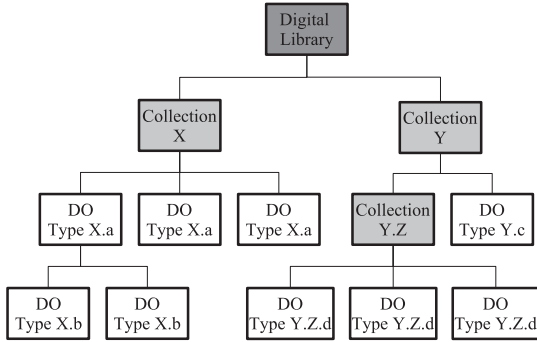


Fig. 6. A Digital Library as Hierarchy of Digital Object Instances with collection-scoped Prototypes

acterise their content. Thus, the elements of a digital collection vary significantly in terms of digital content structure, the metadata sets used to describe it and the user requirements for cataloguing and browsing. For these reasons, digital object prototypes are defined with a *collection-pertinent scope*, affecting collection-specific digital object instances. This allows us (a) to support fine-grained definitions of collection specific kinds of material and (b) to avoid type collision problems – Theatrical Collection Photo prototype may coexist with the Medical Collection Photo in the UoA DL. Each prototype is supplied with composite identifiers, such as `dl.theatrical.photo`, that differs from `dl.medical.photo`. The composite identifier is attached on respective prototype instances, through the Fedora *Content Model* metadata attribute.

For uniformity reasons, all information is stored in Fedora Repository, in terms of digital objects. This stands for collections, too. In order to be able to represent, manage and manipulate all digital content in a unified manner, collections are treated as digital object instances conforming to the *collection prototype*. By representing everything in terms of digital object instances, flexible and effective collection management capabilities can be generated. Collections, sub-collections and the Digital Library itself are represented as a hierarchy of digital object instances, as depicted in Figure 6. This representation scheme provided the following benefits:

- New collections can be easily added in UoA DL, through the creation of a new instance of the collection prototype. This has been of significant importance, providing us the ability to work out the details of each collection independently, but yet in a unified fashion.
- Support collection / sub-collection hierarchies in a configurable manner. A sub-collection is simply a collection instance added in another collection instance.
- Support the characterisation of collections and sub-collections, supplying them with metadata sets, as any other common digital object instance.
- Easily identifiable prototype definitions. The Content Model fully qualified identifier is used to supply the Dictionary with the path of the DL hierarchy

that leads to the specific collection and the definition of the specific prototype in that collection.

6 Discussion

UoA DL System is currently being used for the cataloguing process of the UoA Historical Archive collection. Several extensions are under consideration, such as the generation of a prototype construction interface, that could assist on the creation of digital object prototypes, since, currently, the XML prototype specification need to be issued by direct XML editing. Moreover, the structure part of a digital object should be supplied with general purpose linking capabilities, in order to be able to cope with relations that extend structural containment. We are taking under consideration the use of the Fedora Metadata for Object-to-Object Relationships, introduced in Fedora 2.0, for this purpose. Finally, our current effort is focused on generalising the current definition of behavior schemes in order to support prototype-driven: (a) automatic conversions of *primitive* file types to their prototype-defined *derivatives*, in order to facilitate cataloguing staff (e.g. convert a TIFF image to a low quality JPEG image and a thumbnail) and (b) cataloguing form validation (e.g. metadata fields validation).

The greatest challenge, however, relies in extending Digital Object Prototypes in order to supply them with Object Oriented inheritance capabilities. We argue that it is of great importance to bridge the gap between digital objects and OO objects even further. Approaches on formalisation of inheritance semantics [2,1,17] treat an object as a record of fields, where each field contains a method. Our representation treats objects as aggregations of their four constitutional parts. In a high level of abstraction, these two representations present significant similarities, indicating that it should be possible to incorporate inheritance in the context of digital object prototypes, albeit supplied with the appropriate semantics. Our second long-term goal is to support the concept of OO polymorphism, having digital object instances participate in more than one “is-a” relations. Definition reuse through inheritance has been discussed in [6], although targeted on information retrieval enhancements. Our aim is to use prototype inheritance for enhancing the reuse and configuration capabilities of the Dictionary digital object manipulation mechanism.

References

1. L. Cardelli. A semantics of multiple inheritance. In *Semantics of Data Types*, pages 51–68, 1984.
2. W. Cook and J. Palsberg. A denotational semantics of inheritance and its correctness. In *Proceedings of the ACM Conference on Object-Oriented Programming: Systems, Languages and Application (OOPSLA)*, pages 433–444, 1989.
3. *DCMI Metadata Terms*. Dublin Core Metadata Initiative, January 2005. Available at <http://www.dublincore.org/documents/dcmi-terms/>.
4. N. Dushay. Localizing experience of digital content via structural metadata. In *Proceedings of the Joint Conference on Digital Libraries (JCDL '02)*, pages 244–252, 2002.

5. *Introduction to Fedora Object XML*. Fedora Project. Available at <http://www.fedora.info/download/2.0/userdocs/digitalobjects/introFOXML.html>.
6. N. Fuhr. Object-oriented and database concepts for the design of networked information retrieval systems. In *Proceedings of the 5th international conference on Information and knowledge management*, pages 164–172, 1996.
7. R. Kahn and R. Wilensky. *A Framework for Distributed Digital Object Services*. Corporation of National Research Initiative - Reston USA, 1995. Available at <http://www.cnri.reston.va.us/k-w.html>.
8. C. Lagoze and H. V. de Sompel. The open archives initiative: Building a low-barrier interoperability framework. In *Proceedings of the Joint Conference on Digital Libraries (JCDL '01)*, 2001.
9. I. Lourdi and C. Papatheodorou. A metadata application profile for collection-level description of digital folklore resources. In *Proceedings of the Third International workshop on Presenting and Exploring Heritage on the Web (PEH'04), 15th International Conference and Workshop on Database and Expert Systems Applications DEXA 2004*, pages 90–94, August 2004.
10. *METS Profile Documentation*. Library of Congress.
11. *METS: An Overview & Tutorial*. Library of Congress, September 2004. Available at <http://www.loc.gov/standards/mets/METSOverview.v2.html>.
12. *Data Dictionary - Technical Metadata for Digital Still Images*. NISO Standards Committee, June 2002.
13. *Reference Model for an Open Archival Information System (OAIS)*. Consultative Committee for Space Data Systems (CCSDS), 2002. Blue Book, Issue 1.
14. D. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.
15. G. Pyrounakis, K. Saidis, M. Nikolaidou, and I. Lourdi. Designing an integrated digital library framework to support multiple heterogeneous collections. In *Proceedings of the 8th European Conference on Digital Libraries (ECDL 2004)*, pages 26–37, 2004.
16. *Resource Description Framework (RDF)*. World Wide Web Consortium. Available at <http://http://www.w3.org/RDF/>.
17. U. Reddy. Objects as closures: Abstract semantics of object-oriented languages. In *Proceedings of the ACM Conference on Lisp and Functional Programming*, pages 289–297, 1988.
18. T. Staples, R. Wayland, and S. Payette. The fedora project: An open-source digital object repository management system. *D-Lib Magazine*, 9(4), April 2003.
19. *Web Services Activity*. World Wide Web Consortium. Available at <http://www.w3.org/2002/ws/>.

LibraRing: An Architecture for Distributed Digital Libraries Based on DHTs*

Christos Tryfonopoulos, Stratos Idreos, and Manolis Koubarakis

Dept. of Electronic and Computer Engineering,
Technical University of Crete, GR73100 Chania, Crete, Greece
{trifon, sidraios, manolis}@intelligence.tuc.gr

Abstract. We present a digital library architecture based on distributed hash tables. We discuss the main components of this architecture and the protocols for offering information retrieval and information filtering functionality. We present an experimental evaluation of our proposals.

1 Introduction

We present a digital library (DL) architecture based on ideas from traditional distributed Information Retrieval and recent work on peer-to-peer (P2P) networks. Our architecture, called LibraRing (from the words *library* and *ring*), is hierarchical like the ones in [12,9] but uses a *distributed hash table* (DHT) to achieve robustness, fault-tolerance and scalability in its routing and meta-data management layer. DHTs are the second generation *structured* P2P overlay networks devised as a remedy for the known limitations of earlier P2P networks such as Napster and Gnutella [15].

There are two kinds of basic functionality that we expect this DL architecture to offer: *information retrieval* (IR) and *publish/subscribe* (pub/sub). In an IR scenario, a user can pose a *query* (e.g., “I am interested in papers on bio-informatics”) and the system returns information about matching resources. In a pub/sub scenario (also known as *information filtering* (IF) or *selective dissemination of information* (SDI)), a user posts a *subscription* (or *profile* or *continuous query*) to the system to receive notifications whenever certain events of interest take place (e.g., when a paper on bio-informatics becomes available).

We define the main components of our architecture: *super-peers*, *clients* and *providers*. Providers are used to expose the content of information sources to the network, while clients are used by information consumers. Super-peers form an overlay network that offers a robust, fault-tolerant and scalable means for routing messages and managing resource meta-data and queries. The main architectural contribution of our work is the extension of the DHT Chord protocols [15] with IR and pub/sub functionality in the context of a super-peer network.

Publications and subscriptions in our architecture could be expressed using any appropriate language (e.g., XML and XPath). Whatever language is chosen

* This work was supported in part by project Evergrow. Christos Tryfonopoulos is partially supported by a Ph.D. fellowship from the program Heraclitus of the Greek Ministry of Education.

will have a serious effect on the DHT protocols because the DHT is the layer in which publications and subscriptions live (are indexed). In the rest of this paper, we assume that publications and subscriptions will be expressed using a well-understood attribute-value model, called *AWPS* in [10]. *AWPS* is based on *named attributes* with value *free text* interpreted under the Boolean and vector models VSM or LSI. The query language of *AWPS* allows Boolean combinations of comparisons $A \text{ op } v$, where A is an attribute, v is a text value and op is one of the operators “equals”, “contains” or “similar” (“equals” and “contains” are Boolean operators and “similar” is interpreted using the VSM or LSI model).

The research presented in this paper is a continuation of our previous work on the DL information alert architecture DIAS [10] and the system P2P-DIET [7]. The main difference of the current paper from [10,7] is the definition of an architecture for DLs, brand new protocols that are extensions of DHTs and performance results that illustrate the strengths and weaknesses of our approach.

The rest of this paper is as follows. Section 2 positions our paper with respect to related work in the areas of IR, IF and P2P systems. Section 3 introduces the model *AWPS* while Section 4 presents the Chord DHT. Section 5 discusses the proposed DL architecture and an application scenario. Section 7 presents the LibraRing protocols, while Section 8 summarizes our experimental evaluation. Finally, Section 9 concludes the paper.

2 Related Work

The problem of IR and IF in DL architectures that utilize P2P networks has recently received considerable attention. Here we only discuss papers that we judge most relevant to our work (because they combine an emphasis on DLs, models and languages based on IR, and techniques from P2P networks).

The hierarchical 2-tier architecture of LibraRing is similar to the ones of file sharing systems currently deployed on the Internet, e.g., Kazaa, Gnutella2 and Fast Track, and has also been studied by P2P researchers [2,13]. From these proposals, only Edutella [13,4] uses a structured overlay network for its routing layer; all other approaches rely on techniques from unstructured P2P networks. Some recent proposals targeting DLs also rely on a similar architecture [12,11,7,4,9]. Papers coming from the IR community prefer to use the term *hub node* or *directory node* instead of super-peer or ultra-peer and the term *leaf node* instead of client [12,11,9]. However in our case super-peers have more responsibilities that just to provide directory services which is the case for hubs.

In [11] the problem of content-based retrieval in distributed DLs focusing on resource selection and document retrieval is studied. A 2-tier hierarchical P2P network is proposed, where DLs (represented by leaf nodes) cluster around directory nodes that form an unstructured P2P network in the 2nd level of the hierarchy. In a related paper [12], the authors define the concept of neighborhood in hierarchical P2P networks and use this concept to devise a method for hub selection and ranking. The PlanetP [5] system uses an unstructured P2P network where nodes propagate Bloom filter summaries of their indices to the network using a gossiping algorithm. Each peer uses a variation of *tf/idf* to decide what

nodes to contact to answer a query. In pSearch [19] the authors propose to use the CAN DHT protocol [14] and semantic document vectors (computed using LSI) to efficiently distribute document indices in a P2P network. PIRS [24] uses an unstructured P2P network and careful propagation of metadata information to be able to answer queries in highly dynamic environments. Finally, OverCite [17] is a recent proposal to build a distributed version of CiterSeer using DHTs.

Early work on IF includes SIFT [22,23] which uses the Boolean and VSM models, and InRoute [3] which is based on inference networks. Both of these systems are centralised although some issues related to distribution have been studied in SIFT [23]. Recently, a new generation of IF systems has tried to address the limitations imposed by centralized approaches by relying on ideas from P2P networks. The system P2P-DIET [7], that builds on the earlier proposal DIAS [10], is an IR and IF system that uses the model *AWPS* and is implemented as an unstructured P2P network with routing techniques based on shortest paths and minimum-weight spanning trees. pFilter [18] uses a hierarchical extension of CAN to filter unstructured documents and relies on multi-cast trees to notify subscribers. Architectural considerations regarding the development of an IF system for DLs were first studied by Hermes [6].

None of the papers cited above provides a comprehensive architecture and protocols for the support of *both* IR and IF functionality in DLs using DHTs. This is the emphasis of our work which is presented in the forthcoming sections.

3 The Data Model *AWPS*

We will use a well-understood attribute-value model, called *AWPS* in [10]. A (*resource*) *publication* is a set of attribute-value pairs (A, s) , where A is a *named attribute*, s is a *text* value and all attributes are *distinct*. The following is an example of a publication:

$$\{ (AUTHOR, \text{“John Smith”}), (TITLE, \text{“Information dissemination in P2P ...”}), \\ (ABSTRACT, \text{“In this paper we show that ...”}) \}$$

The query language of *AWPS* offers *equality*, *containment* and *similarity* operators on attribute values. The containment operator is interpreted under the Boolean model and allows Boolean and *word-proximity* queries. The similarity operator is defined as the cosine of the angle of two vectors corresponding to text values from a publication and a query. Vector representations of text values can be computed as usual using the VSM or LSI models (but only the VSM model has been used in our implementation and experiments).

Formally, a *query* is a conjunction of atomic queries of the form $A = s$, $A \supseteq wp$ or $A \sim_k s$, where A is an attribute, s is a text value, wp is a conjunction of words and proximity formulas with only words as subformulas, and k is a *similarity threshold* i.e., a real number in the interval $[0, 1]$. Thus, queries can have two parts: a part interpreted under the Boolean model and a part interpreted under the VSM or LSI model. The following is an example of a query:

$$(AUTHOR = \text{“John Smith”}) \wedge (TITLE \supseteq P2P \wedge (information \prec_{[0,0]} alert)) \wedge \\ (ABSTRACT \sim_{0.7} \text{“P2P architectures have been...”})$$

This query requests resources that have *John Smith* as their author, and their title contains the word *P2P* and a word pattern where the word *information* is immediately followed by the word *alert*. Additionally, the resources should have an abstract similar to the text value “*P2P architectures have been ...*” with similarity greater than 0.7.

4 Distributed Hash Tables

We use an extension of the Chord DHT [15] to implement our super-peer network. Chord uses a variation of *consistent hashing* [8] to map keys to nodes. In the consistent hashing scheme each node and data item is assigned a m -bit identifier where m should be large enough to avoid the possibility of different items hashing to the same identifier (a cryptographic hashing function such as SHA-1 is used). The identifier of a node can be computed by hashing its IP address. For data items, we first have to decide a key and then hash this key to obtain an identifier. For example, in a file-sharing application the name of the file can be the key (this is an application-specific decision). Identifiers are ordered in an *identifier circle (ring)* modulo 2^m i.e., from 0 to $2^m - 1$. Figure 1(b) shows an example of an identifier circle with 64 identifiers ($m = 6$) and 10 nodes.

Keys are mapped to nodes in the identifier circle as follows. Let H be the consistent hash function used. Key k is assigned to the first node which is equal or follows $H(k)$ clockwise in the identifier space. This node is called the *successor* node of identifier $H(k)$ and is denoted by $successor(H(k))$. We will often say that this node is *responsible* for key k . For example in the network shown in Figure 1(b), a key with identifier 30 would be stored at node N_{32} . In fact node N_{32} would be responsible for all keys with identifiers in the interval (21, 32].

If each node knows its successor, a query for locating the node responsible for a key k can always be answered in $O(N)$ steps where N is the number of nodes in the network. To improve this bound, Chord maintains at each node a routing table, called the *finger table*, with at most m entries. Each entry i in the finger table of node n , points to the first node s on the identifier circle that succeeds identifier $H(n) + 2^{i-1}$. These nodes (i.e., $successor(H(n) + 2^{i-1})$ for $1 \leq i \leq m$) are called the *fingers* of node n . Since fingers point at repeatedly doubling distances away from n , they can speed-up search for locating the node responsible for a key k . If the finger tables have size $O(\log N)$, then finding a successor of a node n can be done in $O(\log N)$ steps with high probability [15].

To simplify joins and leaves, each node n maintains a pointer to its *predecessor* node i.e., the first node *counter-clockwise* in the identifier circle starting from n . When a node n wants to join a Chord network, it finds a node n' that is already in the network using some out-of-band means, and then asks n' to find its position in the network by discovering n 's successor [16]. Every node n runs a *stabilization* algorithm periodically to find nodes that have recently joined the network by asking its successor for the successor's predecessor p . If p has recently joined the network then it might end-up becoming n 's successor. Each node n periodically runs two additional algorithms to check that its finger table and predecessor pointer is correct [16]. Stabilization operations may affect queries

by rendering them slower or even incorrect. Assuming that successor pointers are correct and the time it takes to correct finger tables is less than the time it takes for the network to double in size, queries can still be answered correctly in $O(\log N)$ steps with high probability [16].

To deal with node failures and increase robustness, each Chord node n maintains a *successor list* of size r which contains n 's first r successors. This list is used when the successor of n has failed. In practice even small values of r are enough to achieve robustness [16]. If a node chooses to leave a Chord network voluntarily then it can inform its successor and predecessor so they can modify their pointers and, additionally, it can transfer its keys to its successor. Any node joining or leaving a Chord network can use $O(\log^2 N)$ messages to make all successor pointers, predecessor pointers and finger tables correct [15].

5 The LibraRing Architecture

A high-level view of the LibraRing architecture is shown in Figure 1(a). Nodes can implement any of the following types of services: *super-peer service*, *provider service* and *client service*.

Super-peer service. Nodes implementing the super-peer service (*super-peers*) form the message routing layer of the network. Each super-peer is responsible for serving a fraction of the clients by storing continuous queries and resource publications, answering one-time queries, and creating notifications. The super-peers run a DHT protocol which is an extension of Chord. The role of the DHT in LibraRing is very important. First of all, it acts as a rendezvous point for information producers (providers) and information consumers (clients). Secondly, it serves as a robust, fault-tolerant and scalable routing infrastructure. When the number of super-peers is small, each node can easily locate others in a single hop by maintaining a full routing table. When the super-peer network grows in size, the DHT provides a scalable means of locating other nodes as we discussed in Section 4. Finally, by serving as a global metadata index that is partitioned among super-peers, the DHT facilitates building a distributed metadata repository that can be queried efficiently.

Client service. Nodes implementing the client service are called *clients*. A client connects to the network through a single super-peer node, which is its *access point*. Clients can connect, disconnect or even leave the system silently at any time. Clients are information consumers: they can pose one-time queries and receive answers, subscribe to resource publications and receive notifications about published resources that match their interests. If clients are not on-line, notifications matching their interests are stored by their access points and delivered once clients reconnect. Resource requests are handled directly by the client that is the owner of the resource.

Provider service. This service is implemented by information sources that want to expose their contents to the clients of LibraRing. A node implementing the provider service (*provider*) connects to the network through a super-peer

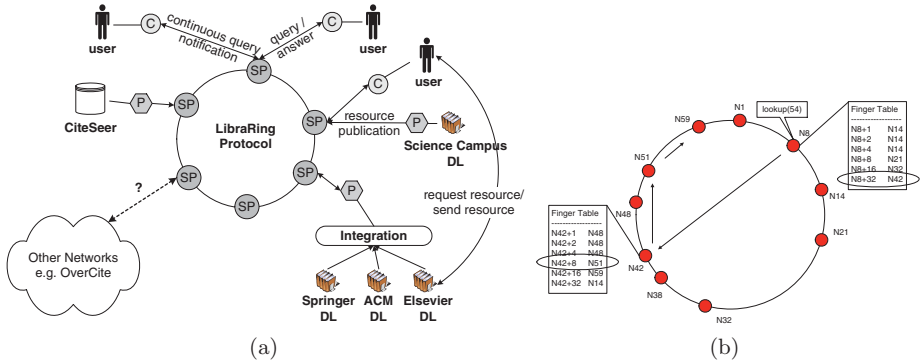


Fig. 1. The architecture of LibraRing and an example of a lookup operation over a Chord ring with $m=6$

which is its access point. To be able to implement this service, an information source should create meta-data for the resources it stores using data model *AWPS*, and publish it to the rest of the network using its access point.

As an example of an application scenario let us consider a university with 3 geographically distributed campuses (Arts, Sciences and Medicine) and a local digital library in each campus (see Figure 1(a)). Each campus maintains its own super-peer, which provides an access point for the provider representing the campus digital library, and the clients deployed by users. A university might be interested in making available to its students and staff, in a timely way, the content provided by other publishers (e.g., CiteSeer, ACM, Springer, Elsevier). Figure 1(a) shows how our architecture can be used to fulfill this requirement. An integration layer is used to unify different types of DLs. At this level, we also expect to see *observer modules* (as in [6]) for information sources that do not provide their own alerting service. This module will query the sources for new material in a scheduled manner and inform providers accordingly.

Questions of interoperability are not discussed in this paper. Figure 1(a) assumes that some of them are solved at the level of providers with well-known integration techniques, and leaves open the question of how to interoperate with other P2P networks e.g., the DHT implementing OverCite [17]. The latter question is currently under study in project Evergrow¹ that is funding this work.

6 Extensions to the Chord API

To facilitate message sending between nodes we will use the function `send(msg, I)` to send message msg from some node to node $successor(I)$, where I is a node identifier. Function `send()` is similar to Chord function `lookup(I)` [15], and costs $O(\log N)$ overlay hops for a network of N nodes. When function `send(msg, I)` is invoked by node S , it works as follows. S contacts S' , where $id(S')$ is the greatest identifier contained in the finger table of S , for which $id(S') \leq I$ holds.

¹ <http://www.evergrow.org/>

Upon reception of a `send()` message by a node S , I is compared with $id(S)$. If $id(S) < I$, then S just forwards the message by calling `send(msg, I)` itself. If $id(S) \geq I$, then S processes msg since it is the *intended recipient*.

Our protocols described in Section 7 also require that a node is capable of sending the *same* message to a group of nodes. This group is created dynamically each time a resource publication or a query submission takes place, so multicast techniques for DHTs such as [1] are not applicable. The obvious way to handle this over Chord is to create k different `send()` messages, where k is the number of different nodes to be contacted, and then locate the recipients of the message in an *iterative* fashion using $O(k \log N)$ messages. We have implemented this algorithm for comparison purposes.

We have also designed and implemented function `multiSend(msg, L)`, where L is a list of k identifiers, that can be used to send message msg to the k elements of L in a *recursive* way. When function `multiSend()` is invoked by node S , it works as follows. Initially S sorts the identifiers in L in ascending order clockwise starting from $id(S)$. Subsequently S contacts S' , where $id(S')$ is the greatest identifier contained in the finger table of S , for which $id(S') \leq head(L)$ holds, where $head(L)$ is the first element of L . Upon reception of a `multiSend()` message, by a node S , $head(L)$ is compared with $id(S)$. If $id(S) < head(L)$, then S just forwards msg by calling `multiSend()` again. If $id(S) \geq head(L)$, then S processes msg since this means that it is *one* of the intended recipients contained in list L (in other words, S is responsible for key $head(L)$). Then S creates a new list L' from L in the following way. S deletes all elements of L that are smaller or equal to $id(S)$, starting from $head(L)$, since S is responsible for them. In the new list L' that results from these deletions, we have that $id(S) < head(L')$. Finally, S forwards msg to node with identifier $head(L')$ by calling `multiSend(msg, L')`. This procedure continues until all identifiers are deleted from L . The recursive approach has in practice a significantly better performance than the iterative method as we show in Section 8.

7 The LibraRing Protocols

In this section we describe in detail the way clients, providers and super-peers join and leave the network. We also describe resource publication and query submission protocols. We use functions $key(n)$, $ip(n)$ and $id(n)$ to denote the key, the IP address and the identifier of node n respectively.

7.1 Client Join

The *first time* that a client C wants to connect to the LibraRing network, it has to follow the join protocol. C must find the IP address of a super-peer S using out-of-band means (e.g., via a secure web site that contains IPs for the super-peers that are currently online). C sends to S message `NEWCLIENT(key(C), ip(C))` and S adds C in its *clients table* (CT), which is a hash table used for identifying the peers that use S as their *access point*. $key(C)$ is used to index clients in CT , while each CT slot stores contact information about the client, its status

(connected/disconnected) and its stored notifications (see Section 7.6). Additionally, S sends to C an acknowledgement message $\text{ACKNEWCLIENT}(id(S))$. Once C has joined, it can use the connect/disconnect protocol (to be described below) to connect and disconnect from the network.

Providers use a similar protocol to join a LibraRing network.

7.2 Client Connect/Disconnect

When a client C wants to connect to the network, it sends to its access point S message $\text{CONNECTCLIENT}(key(C), ip(C), id(S))$. If $key(C)$ exists in the CT of S , C is marked as connected and stored notifications are forwarded to it. If $key(C)$ does not exist in CT , this means that S was not the access point of C the last time that C connected (Section 7.7 discusses this case).

When a client C wants to disconnect, it sends to its access point S message $\text{DISCONNECTCLIENT}(key(C), ip(C))$. S marks C as disconnected in its CT without removing information related to C , since this information will be used to create stored notifications for C while C is not online (see Section 7.6).

Providers connect to and disconnect from the network in a similar way.

7.3 Resource Indexing

A resource is indexed in three steps. In the first step a provider P constructs a publication $p = \{(A_1, s_1), (A_2, s_2), \dots, (A_n, s_n)\}$ (the resource description) and sends message $\text{PUBRESOURCE}(key(P), ip(P), key(p), p)$ to its access point S .

In step two, S forwards p to the appropriate super-peers as follows. Let D_1, \dots, D_n be the sets of *distinct* words in s_1, \dots, s_n . Then p is sent to *all* nodes with identifiers in the list $L = \{H(w_j) : w_j \in D_1 \cup \dots \cup D_n\}$. The subscription protocol guarantees that L is a superset of the set of identifiers responsible for queries that match p . Subsequently S removes duplicates and sorts L in ascending order clockwise starting from $id(S)$. This way we obtain less identifiers than the distinct words in $D_1 \cup \dots \cup D_n$, since a super-peer may be responsible for more than one words contained in the document. Having obtained L , S indexes p by creating message $msg = \text{INDEXRESOURCE}(ip(P), key(P), ip(S), key(p), p)$, and calling function $\text{multiSend}(msg, L)$.

Finally in the third step, each super-peer S' that receives this message stores p in an *inverted index* that will facilitate matching against one-time queries that will arrive later on at S' .

7.4 Submitting an One-Time Query

In this section we show how to answer one-time queries containing Boolean and vector space parts (denoted as queries of type T1) or only vector space parts (denoted as queries of type T2). The first type of queries is always indexed under its Boolean part. Let us assume that a client C wants to submit a query q (of type T1) of the form $\bigwedge_{i=1}^m A_i = s_i \wedge \bigwedge_{i=m+1}^n A_i \supseteq wp_i \wedge \bigwedge_{i=n+1}^k A_i \sim_{a_i} s_i$.

The following three steps take place. In step one, C sends to its access point S message $\text{SUBMITQ}(key(C), ip(C), key(q), q)$.

In the second step, S randomly selects a single word w contained in any of the text values s_1, \dots, s_m or word patterns wp_{m+1}, \dots, wp_n and computes $H(w)$ to obtain the identifier of the super-peer storing publications that can match q . Then it sends message $msg = \text{POSEQUERY}(ip(C), key(C), ip(S), key(q), q)$ by calling function $\text{send}(msg, H(w))$.

If q is of the form $A_{n+1} \sim_{a_1} s_1 \wedge \dots \wedge A_n \sim_{a_n} s_n$ (query type T2) then step two is modified as follows. Let D_1, \dots, D_n be the sets of *distinct* words in s_1, \dots, s_n . q has to be sent to *all* super-peers with identifiers in the list $L = \{H(w_j) : w_j \in D_1 \cup \dots \cup D_n\}$. To do so, S removes duplicates, sorts L in ascending order clockwise starting from $id(S)$ and sends message $msg = \text{POSEQUERY}(ip(C), key(C), ip(S), key(q), q)$ by calling $\text{multiSend}(msg, L)$.

In step three, each super-peer that receives an one-time query q , it matches it against its local publication store to find out which providers have published documents that match q and delivers answers as discussed in Section 7.6.

7.5 Pub/Sub Functionality

This section describes how to extend the protocols of Sections 7.4 and 7.3 to provide pub/sub functionality. To index a continuous query cq the one-time query submission protocol needs to be *modified*. The first two steps are identical, while the third step is as follows. Each super-peer that receives cq , it stores cq in its local continuous query data structures to match it against incoming publications. A super-peer S uses a hash table to index all the atomic queries of cq , using as key the attributes A_1, \dots, A_k . To index each atomic query, three different data structures are also used: (i) a hash table for text values s_1, \dots, s_m , (ii) a trie-like structure that exploits common words in word patterns wp_{m+1}, \dots, wp_n , and (iii) an inverted index for the most “significant” words in text values s_{n+1}, \dots, s_k . S' utilises these data structures at filtering time to find quickly all continuous queries cq that match an incoming publication p . This is done using an algorithm that combines algorithms BestFitTrie [21] and SQI [23].

To index a resource, the protocol of Section 7.3 needs to be *extended*. The first two steps are identical, while in the third step, each super-peer that receives p matches it against its local continuous query database using the algorithms BestFitTrie and SQI.

7.6 Notification Delivery

Assume a super-peer S that has to deliver a notification n for a continuous query cq to client C . It creates message $msg = \text{NOTIFICATION}(ip(P), key(P), pid(p), qid(cq))$, where P is the provider that published the matching resource and sends it to C . If C is not online, then S sends msg to S' , where S' is the access point of C , using $ip(S')$ associated with cq . S' stores msg , to deliver it to C upon re-connection. If S' is also off-line msg is sent to the $\text{successor}(id(S'))$, by calling function $\text{send}(msg, \text{successor}(id(S')))$. Answers to one-time queries are handled in a similar way. In case that more that one answers or notifications have to be delivered, function $\text{multiSend}()$ is used.

7.7 Super-Peer Join/Leave

To join LibraRing network, a super-peer S must find the IP address of another super-peer S' using out-of-band means. S creates message `NEWSPEER`($id(S)$, $ip(S)$) and sends it to S' which performs a lookup operation by calling `lookup`($id(S)$) to find $S_{succ} = successor(id(S))$. S' sends message `ACKNEWSPEER`($id(S_{succ})$, $ip(S_{succ})$) to S and S updates its successor to S_{succ} . S also contacts S_{succ} asking its predecessor and the data that should now be stored at S . S_{succ} updates its predecessor to S , and answers back with the contact information of its previous predecessor, S_{pred} , and all continuous queries and publications that were indexed under key k , with $id(S) \leq k < id(S_{pred})$. S makes S_{pred} its predecessor and populates its index structures with the new data that arrived. After that S populates its finger table entries by repeatedly performing lookup operations on the desired keys.

When a super-peer S wants to leave LibraRing network, it constructs message `DISCONNECTSPEER`($id(S)$, $ip(S)$, $id(S_{pred})$, $ip(S_{pred})$, $data$), where $data$ are all the continuous queries, published resources and stored notifications of off-line peers that S was responsible for. Subsequently, S sends the message to its successor S_{succ} and notifies S_{pred} that its successor is now S_{succ} . Clients that used S as their access point connect to the network through another super-peer S' . Stored notifications can be retrieved through $successor(id(S))$.

8 Experimental Evaluation

In [20] we present a detailed evaluation of DH`Trie`, a set of protocols that are essentially the LibraRing protocols in the case that all nodes in the system are equal DHT nodes (i.e., there are no super-peers). [20] deals only with the pub/sub case and evaluates the DH`Trie` protocols only under queries of type T1. In this section we continue this evaluation for the complete LibraRing protocols using the same data and query set. As in [21,20], we use a corpus of 10426 documents downloaded from CiteSeer and synthetically generated queries. In all graphs the y -axis has been truncated to show clearly the best performing algorithms.

We have implemented and experimented with four variations of the LibraRing protocols. The first one, named *It*, utilises the iterative method in the publication protocol. The second algorithm, named *ItC*, utilises the iterative method and an additional routing table called *FCache*. *FCache* is a routing table that stores the IP addresses of super-peers responsible for *frequent words* and is exploited for reaching nodes in a single hop. A detailed description and performance evaluation for *FCache* is given in [20]. The third algorithm, named *Re*, utilises the recursive method in the publication protocol. Finally, *ReC* uses the recursive method and *FCache* and outperforms the rest of the algorithms.

Figure 2(a) shows the number of messages needed by each algorithm to index a resource publication to the appropriate super-peers for different super-peer network sizes. The algorithms remain relatively unaffected by network size since a publication needs 5% more messages for a 5 times larger network. All algorithms

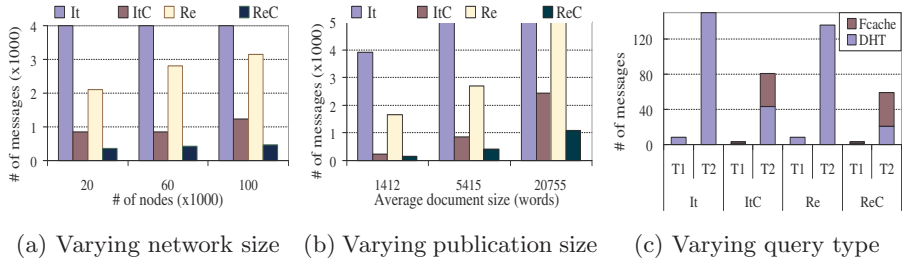


Fig. 2. Average number of messages to index a publication (a,b) and a query (c)

show a similar increase rate, with *ReC* presenting the best performance, needing 400 messages to index an incoming publication to a network of 100K super-peers.

Publication size is an important parameter in the performance of our algorithms. Figure 2(b) shows that for small publications the use of the recursive method (contrary to FCache) does not improve performance, since algorithms *ItC* and *ReC* perform similarly. For large publications though, the use of the recursive method and FCache is shown to improve performance significantly. Additionally the increase in message cost is linear to the publication size with algorithm *ReC* exhibiting the smallest increase rate, thus showing the smallest sensitivity to publication size.

Finally Figure 2(c) shows the cost indexing a query to the network. Notice that queries of type T2 are much more expensive to index needing about 60 messages to reach the responsible super-peers, while queries of type T1 can be indexed in an average of 8 messages. Again algorithm *ReC* shows a significant improvement in message cost.

9 Outlook

We are currently implementing LibraRing in the context of project Evergrow. We are also moving to more expressive languages that combine hierarchical structure and textual information (e.g., XQuery with full-text operators).

References

1. L.O. Alima, A. Ghodsi, P. Brand, and S. Haridi. Multicast in DKS(N, k, f) Overlay Networks. In *Proc. of OPODIS*, 2003.
2. B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings ICDE*, 2003.
3. J.P. Callan. Document Filtering With Inference Networks. In *Proc. of ACM SIGIR*, 1996.
4. P. A. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Publish/Subscribe for RDF-based P2P Networks. In *Proc. of ESWS*, 2004.
5. F.M. Cuenca-Acuna and T.D. Nguyen. Text-Based Content Search and Retrieval in Ad-hoc P2P Communities. In *Networking 2002 Workshops*, 2002.

6. D. Faensen, L. Faulstich, H. Schweppe, A. Hinze, and A. Steidinger. Hermes – A Notification Service for Digital Libraries. In *Proc. of JCDL*, 2001.
7. S. Idreos, M. Koubarakis, and C. Tryfonopoulos. P2P-DIET: An Extensible P2P Service that Unifies Ad-hoc and Continuous Querying in Super-Peer Networks. In *Proc. of SIGMOD*, 2004. Demo paper.
8. D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. In *Proceedings of STOC*, 1997.
9. I.A. Klampanos and J.M. Jose. An Architecture for Information Retrieval over Semi-Collaborating Peer-to-Peer Networks. In *Proc. of SAC*, 2004.
10. M. Koubarakis, T. Koutris, P. Raftopoulou, and C. Tryfonopoulos. Information Alert in Distributed Digital Libraries: The Models, Languages and Architecture of DIAS. In *Proc. of ECDL*, 2002.
11. J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proc. of CIKM*, 2003.
12. J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *Proc. of ECIR*, 2005. (to appear).
13. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. EDUTELLA: A P2P Networking Infrastructure Based on RDF. In *Proc. of WWW*, 2002.
14. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*, 2001.
15. I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. of ACM SIGCOMM*, 2001.
16. I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Frans Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, 2003.
17. J. Stribling, I.G. Councill, J. Li, M.F. Kaashoek, D.R. Karger, R. Morris, and S. Shenker. OverCite: A Cooperative Digital Research Library. In *Proc. of IPTPS*, 2005.
18. C. Tang and Z. Xu. pFilter: Global Information Filtering and Dissemination Using Structured Overlays. In *Proc. of FTDCS*, 2003.
19. C. Tang, Z. Xu, and M. Mahalingam. pSearch: Information Retrieval in Structured Overlays. In *Proc. of HotNets-I '02*.
20. C. Tryfonopoulos, S. Idreos, and M. Koubarakis. Publish/Subscribe Functionality in IR Environments using Structured Overlay Networks. In *Proc. of ACM SIGIR*, 2005.
21. C. Tryfonopoulos, M. Koubarakis, and Y. Drougas. Filtering Algorithms for Information Retrieval Models with Named Attributes and Proximity Operators. In *Proc. of ACM SIGIR*, 2004.
22. T.W. Yan and H. Garcia-Molina. Index structures for selective dissemination of information under the boolean model. *ACM TODS*, 19(2):332–364, 1994.
23. T.W. Yan and H. Garcia-Molina. The SIFT information dissemination system. *ACM TODS*, 24(4):529–565, 1999.
24. W.G. Yee and O. Frieder. The Design of PIRS, a Peer-to-Peer Information Retrieval System. In *Proc. of DBISP2P*, 2004.

Hierarchical Organization and Description of Music Collections at the Artist Level

Elias Pampalk¹, Arthur Flexer^{1,2}, and Gerhard Widmer^{1,3}

¹ Austrian Research Institute for Artificial Intelligence, Vienna, Austria

² Department of Medical Cybernetics and Artificial Intelligence,
Center for Brain Research, Vienna University of Medicine, Austria

³ Department of Computational Perception,
Johannes Kepler University Linz, Austria
{elias, arthur, gerhard}@ofai.at

Abstract. As digital music collections grow, so does the need to organizing them automatically. In this paper we present an approach to hierarchically organize music collections at the artist level. Artists are grouped according to similarity which is computed using a web search engine and standard text retrieval techniques. The groups are described by words found on the webpages using term selection techniques and domain knowledge. We compare different term selection techniques, present a simple demonstration, and discuss our findings.

1 Introduction

The value of a large music collection is limited by how efficiently a user can explore it. Portable audio players which can store over 20,000 songs and on-line music shops with more than 1 million tracks in their repositories are not uncommon. Thus, simple artist/album based organizations are insufficient.

The probably best known approach to organize music is to classify it into genres such as pop, rock, classical etc. Experts are able to name several subgenres for each genre. An example is Ishkur's guide to electronic music with 180 subgenres within electronic music alone.¹ However, classifying music into genres is not the perfect solution (for a detailed discussion see e.g. [1]). One of the main issues is that it is a very cumbersome task and in many cases requires an expert. Furthermore, large genre taxonomies tend to be inconsistent and have rather fuzzy boundaries between genres. An alternative is to use some form of similarity defined directly between artists. This allows a collection to be browsed starting with an familiar artist and exploring related ones. Successful examples include Amazon's recommendations.

In this paper we analyze the content of webpages ranked by Google to compute the similarity of artists. We investigate how this similarity can be used to automatically organize artists into overlapping hierarchical clusters. Furthermore, we investigate the advantages and disadvantages of different strategies to

¹ <http://www.di.fm/edmguide>

automatically describe these clusters with words. An HTML-based demonstration is available.²

The following sections are: (2) related work, (3) similarity computations, (4) hierarchical clustering, (5) term selection for clusters description, (6) results and discussion (including the user interface), (7) conclusions.

2 Related Work

Music information retrieval is relatively young research field. However, there are a number of publications related to extracting information from the web and approaches to structure and summarize music collections.

One of the first approaches using web-based data to compute artist similarities was presented in [2]. Co-occurrences on playlists from radio stations and compilation CD databases were used to cluster a set of 12 songs and a set of 100 artists hierarchically according to similarity. The approach was demonstrated using single-linkage agglomerative clustering. The main difference of the approach we present in this paper is that we focus on automatically describing the *contents* of these clusters. We also used a larger set of artists, a different data source, and a different clustering technique.

Another source are common webpages [3,4]. The main idea is to retrieve top ranked sites from Google queries and apply standard text-processing techniques. Using the obtained word lists, the artist similarities are computed. A drastically simplified approach is to use the number of pages found by Google for a query containing two artist names [5,6]. As the evaluation of artist similarity is quite difficult [7] it is tempting to resort to a genre classification scenario [8]. Other web-based sources include expert opinions (such as those available from the All Music Guide³), album reviews [9], or song lyrics [10].

A combination of audio-based and web-based sources is very desirable; however, that is outside of the scope of this paper. First approaches demonstrating the advantages of the combination can be found, for example, in [11,9,12].

Related work in terms of structuring and describing music collections includes the Islands of Music approach which is based on audio signal analysis [13,14]. The idea is to organize music collections on a map such that similar pieces are located close to each other. The clusters are visualized using a metaphor of geographic maps and are labeled with abstract terms describing low-level audio signal characteristics. Furthermore, “weather charts” are used to describe the value of one attribute (e.g. bass energy) across the different regions of the map. Hierarchical extensions were presented in [15,16].

The same hierarchical approach we use for this work we previously applied to organize large collections of drum sample libraries [17]. The basic idea is similar to the approach used by the search engine vivisimo.⁴

² <http://www.oefai.at/~elias/wa>

³ <http://www.allmusic.com>

⁴ <http://www.vivisimo.com>

3 Similarity Computations

In this section we describe how we compute the similarity between artists. This approach is a simplified version of the approach originally presented in [3] and is based on standard text information retrieval techniques. In previous work [8] we applied this approach successfully to classify artists into genres.

For each artist a query string consisting of the artist's name as an exact phrase extended by the keywords *+music +review* is sent to Google using Google's SOAP interface.⁵ For each artist the 50 top ranked pages are retrieved. We remove all HTML markup tags, taking only the plain text content into account. Using a stop word list we remove very frequent and unwanted terms.⁶

Let the term frequency tf_{ta} be the number of occurrences (frequency) of word (term) t in the pages retrieved for artist a . Let the document frequency df_{ta} be the number of webpages (documents) for a in which t occurs at least once, and let $df_t = \sum_a df_{ta}$.

First, for computational reasons, for each artist we remove all terms for which $df_{ta} < 3$. (The highest possible value for df_{ta} is 50). Then we merge all individual term lists into one global list. From this we remove all terms for which there is no $df_{ta} \geq 10$. (That is, for at least one artist the term must occur in at least 10 pages.)

In our experiment with 224 artists 4139 terms remained.⁷ The data is inherently extremely high-dimensional as the first 200 eigenvalues (using an eigenvalue decomposition, also known as PCA) are needed to describe 95% of the variance.

The frequency lists are combined using the term frequency \times inverse document frequency ($tf \times idf$) function (we use the *ltc* variant [18]). The term weight per artist is computed as,

$$w_{ta} = \begin{cases} (1 + \log_2 tf_{ta}) \log_2 \frac{N}{df_t}, & \text{if } tf_{ta} > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where N is the total number of pages retrieved.

This gives us a vector of term weights for each artist. We normalize the weights such that the length of the vector equals 1 (Cosine normalization) to remove the influence the document length would otherwise have. Finally, the distance between two artists is computed as the Euclidean distance of the normalized term weight vectors.

The evaluation of this approach within a genre classification context can be found in [8]. For the set of 224 artists (manually assigned to 14 genres) which we use in our experiments we get accuracies of 85% for leave-one-out evaluation using a nearest neighbor classifier.

⁵ This service is offered free of charge but is limited to 1000 queries a day per registered user. Each query returns 10 pages. <http://www.google.com/apis>

⁶ <http://www.oefai.at/~elias/wa/stopwords.txt>

⁷ A list of the artists is available from:

<http://www.cp.jku.at/people/knees/artistlist224.html>

4 Hierarchical Clustering

There is a close link between clustering and the user interface. For the user interface we use simple lists of items instead of complex graphical visualizations to convey the information. We assume that there is a certain number of items on the list which can be displayed best. We arbitrarily choose 5, which is significantly less than the 14 genres present in the collection we use for our experiments.

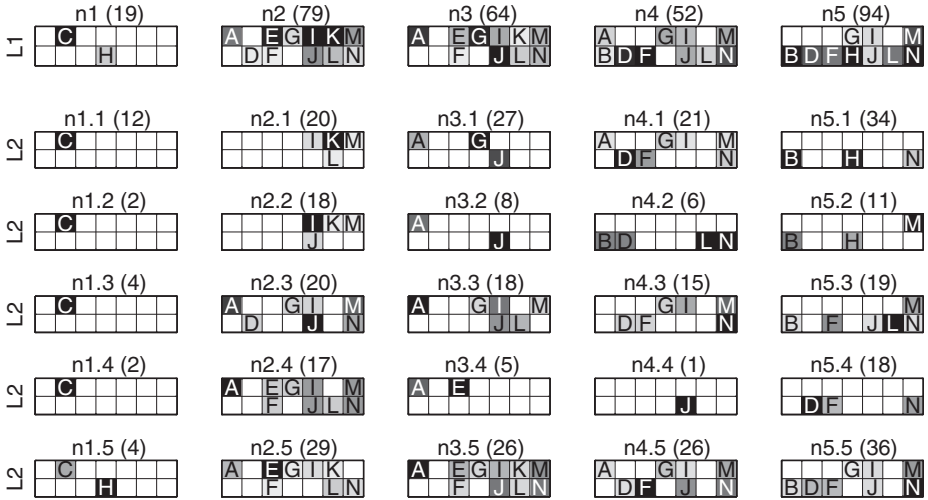


Fig. 1. Distribution of the 14 genres in the nodes of the first level (L1, first row) and second level (L2, all columns starting with the second row). For example, n2.4 (L2 node) is the fourth child of parent n2 (L1 node). Each subplot represents the distribution of genres in a node (visualized as histogram and displayed in two lines to save space). Black corresponds to high values. The boxes in the histogram corresponds to the following genres: A alternative/indie, B blues, C classic, D country, E electronic, F folk, G heavy, H jazz, I pop, J punk, K rap/hip-hop, L reggae, M R&B/soul, N rock & roll. The numbers in brackets are the number of artists mapped to the node.

A clustering technique suitable to our requirements is the one-dimensional self organizing map (SOM) [19] which can be structured hierarchically [20,21]. More flexible approaches include the growing hierarchical self-organizing map [22]. Other alternatives include, for example, hierarchical agglomerative clustering as used in [2].

The SOM groups similar items into clusters and places similar clusters close to each other. After training the SOM we increase the cluster size by 20% adding the artists closest to the border. This overlap makes it easier to find artists which are on the border of two or more clusters. Recursively, for each cluster another one-dimensional SOM is trained (for all artists assigned to the cluster) until the cluster size falls below a certain limit (e.g. only 7 artists remaining). In previous work we used the same one-dimensional overlapping clustering approach and a

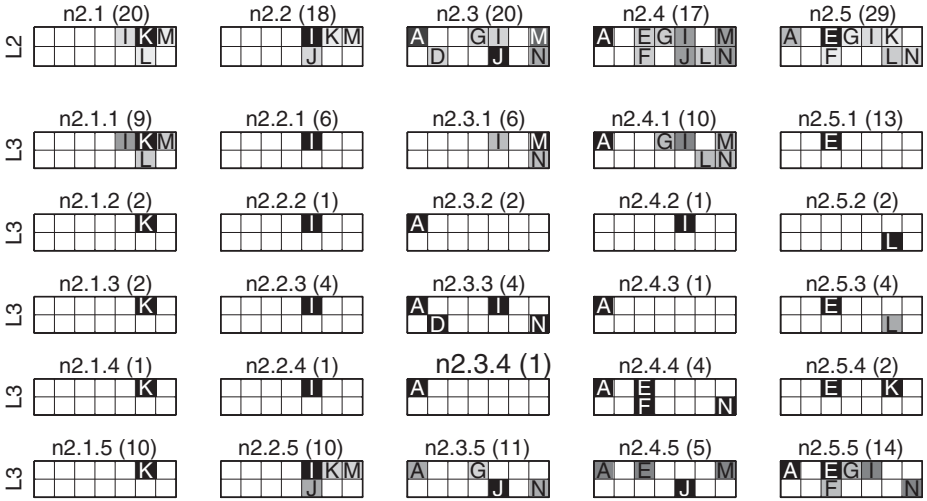


Fig. 2. Distribution of the 14 genres in the nodes of the second (L2) and third (L3) level in the subbranch of node n2

similar user interface to organize large drum sample libraries. The user feedback we got was very positive [17].

Figure 1 shows the distribution of the genres within the nodes (i.e. clusters) in the hierarchical structure. (A screenshot of how the hierarchy is displayed in the user interface is shown in Figure 3.) At the first level classical music (node n1) is well separated from all other music. The effects of the overlap are immediately visible as the sum of artists mapped to all units in the first layer is beyond 224. One example for a direct effect of the overlap is that there is jazz music in node n1, which would not be there otherwise. The nodes n1 and n5 are the only ones at the first level containing jazz music. Electronic and rap/hip-hop is only contained in n2 and n3, and blues only in n4 and n5.

At the second level most nodes have specialized. For example, n5.1 contains 34 artists mainly from jazz and blues and few from rock & roll. Another nice example is n3.2 which contains mostly punk but also some alternative. An interesting observation is that the one-dimensional ordering of the SOM (i.e. similar units should be close to each other) is not apparent. One reason for this might be the extremely high-dimensional data (as mentioned previously, 200 eigenvectors are necessary to preserve 95% of the variance in the data). Another observation indicating that the approach is not flawless is that there are some nodes which seem to contain a bit of almost every genre.

Figure 2 shows what happens at the third level (in the subbranch of node n2). For example, while node n2.3 contains artists from punk and soul/R&B none of its children mix the two. Another positive example is that node n2.5.1 captures most of the electronic music in n2.5. However, as can be seen the clustering is far from perfect and leaves a lot of room for improvement.

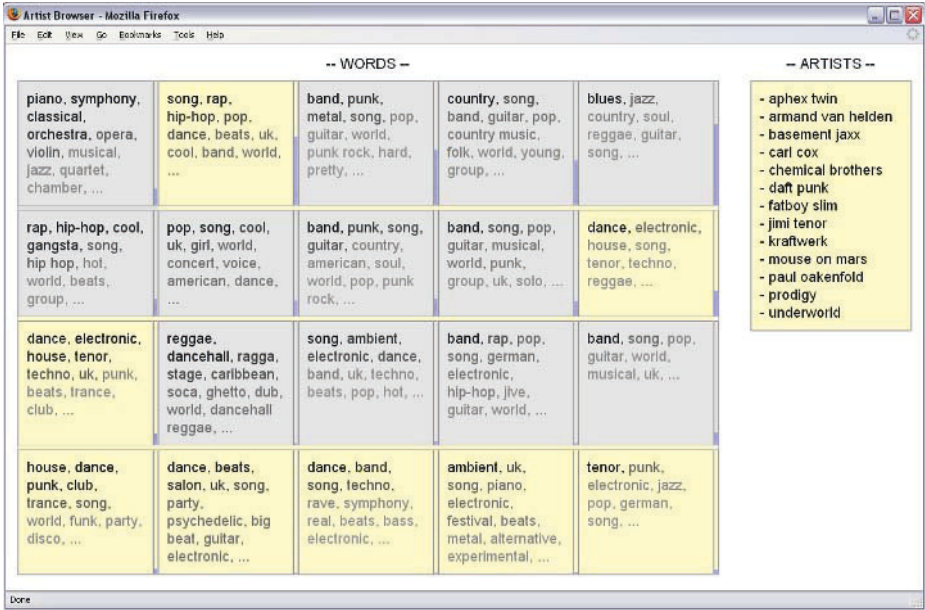


Fig. 3. Screenshot of the HTML user interface

5 Term Selection for Cluster Description

Term selection is a core component of the user interface. The goal is to select words which best summarize a group of artists. With respect to the user interaction we make three assumptions with impact on the cluster description. First, the artists are mostly unknown to the user (otherwise we could just label the nodes with the artists' names). Second, we also do not know which artists the user knows (otherwise we could use those to describe the nodes). Third, we assume that space is limited and thus we want to describe each node with as few words as possible. Dropping assumption two could lead to very interesting interactive user interfaces. However, this is beyond the scope of this paper.

In our experiments we compare five term selection techniques, and two different concepts regarding the set of terms to start with in the first place. In particular, we suggest using a domain-specific dictionary instead of the terms used to compute the similarity.

5.1 Techniques

Given are the term frequency tf_{ta} , document frequency df_{ta} , and the Cosine normalized $tf \times idf$ weight w_{ta} for each term t and artist a . A straightforward approach is to use the $tf \times idf$ computations, i.e. w_{ta} . For each node we compute the average over the cluster c of the assigned artists w_{tc} and select the terms with the highest values.

The second approach is called “LabelSOM” [23] and has successfully been applied to label large document collections organized by SOMs. LabelSOM is built on the observation that terms with a very high w_{tc} and a high variance (i.e., they are very rare in some of the documents in the cluster) are usually poor descriptors. Thus, instead of w_{tc} the variance of w_{ta} in c is used to rank the terms (better descriptors have lower variances). Since terms which do not occur in c ($w_{tc} = 0$) have variance 0, terms with w_{tc} below a manually defined threshold are removed from the list of possible candidates. This threshold depends on the number of input dimensions and how the vectors are normalized. For the 4139 dimensions we used a threshold of 0.045. For the approach with the dictionary (see below) where we have 1269 dimensions we used a threshold of 0.1. Note that the variance in a cluster consisting only of one artist is meaningless. In such cases we use $tf \times idf$ ranking instead.

Neither $tf \times idf$ ranking nor LabelSOM try to find terms which discriminate two nodes. However, emphasizing differences between nodes of the same parent helps reduce redundancies in the descriptions. Furthermore, we can assume that the user already knows what the children have in common after reading the description of the parent. A standard technique to select discriminative terms is the χ^2 (chi-square) test (e.g. [24]). The χ^2 -value measures the independence of t from group c and is computed as,

$$\chi_{tc}^2 = \frac{N(AD - BC)^2}{(A + B)(A + C)(B + D)(C + D)} \quad (2)$$

where A is the number of documents in c containing t , B the number of documents not in c containing t , C the number of documents in c without t , D the number of documents not in c without t , and N is the total number of retrieved documents. As N is equal for all terms, it can be ignored. The terms with highest χ_{tc}^2 values are selected because they are least independent from c . Note that the document frequency is very informative because df_{ta} describes the percentage of times the terms occur in the 50 retrieved documents per artist.

The fourth technique was proposed by Lagus and Kaski (LK) [25]. Like LabelSOM it was developed to label large document collections organized by SOMs. While χ^2 only uses df , LK only use tf . The heuristically motivated ranking formula (higher values are better) is,

$$f_{tc} = (tf_{tc} / \sum_{t'} tf_{t'c}) \cdot \frac{(tf_{tc} / \sum_{t'} tf_{t'c})}{\sum_{c'} (tf_{tc'} / \sum_{t'} tf_{t'c'})}, \quad (3)$$

where tf_{tc} is the average term frequency in cluster c . The left side of the product is the importance of t in c defined through the frequency of t relative to the frequency of other terms in c . The right side is the importance of t in c relative to the importance of t in all other clusters.

The fifth approach is a variation of LK. We implemented it to demonstrate the effects of extreme discrimination. In particular, in this variation tf_{tc} are normalized over the whole collection such that a word which occurs 100 times in cluster c and never in any cluster is equally important to a word that occurs

once in c and never otherwise. As we will see later his approach can only produce meaningful results when used with a specialized dictionary. We ignore all terms which do not occur in at least 10% of the documents per cluster. The ranking function (higher values are better) is,

$$f_{tc} = (tf_{tc} / \sum_{c'} tf_{tc'}) \cdot \frac{(tf_{tc} / \sum_{c'} tf_{tc'})}{\sum_{c''} (tf_{tc''} / \sum_{c'} tf_{tc'})}. \quad (4)$$

In addition we implemented two combinations. In particular combining LK with χ^2 , and the LK variant with χ^2 . In both cases the values were combined by multiplication.

5.2 Domain-Specific Dictionary

One of the main pillars of our approach is the use of a dictionary to avoid describing clusters with artist, album, and other specialized words likely to be unknown to the user. This dictionary contains general words used to describe music, such as genre names. The dictionary contains 1398 entries,⁸ 1269 of these occur in the retrieved documents. The dictionary was manually compiled by the authors in a sloppy manner by copying lists from various sources such as Wikipedia, the Yahoo directory, allmusic.com, and other sources which contained music genres (and subgenres), instruments, or adjectives. The dictionary is far from complete and contains terms which should be removed (e.g. *world*, *uk*, *band*, and *song*). However, to get a better understanding of possible drawbacks we did not modify the dictionary.

We parse each retrieved webpage and compute the term frequencies, document frequencies, and $tf \times idf$. So why did we not use the dictionary to compute the similarities? There are two reasons.

First, the classification performance using k -nearest neighbors with leave-one-out validation is only about 79% compared to the 85% of the standard approach. Considering the size of the collection this might not be significant. However, the explanation for this is that the standard approach captures a lot of the very specific words such as the artists names, names of their albums and many other terms which co-occur on related artist pages.

Second, while the dictionary is an important pillar of our approach we try not to rely too much upon it. By manipulating the dictionary it is very likely that we could achieve 100% classification accuracies on our set of 224 artists. However, such results could not be generalized to other music collections. Furthermore, in our current approach the specialized dictionary can be replaced at any time without impact on the hierarchical structure.

6 Results and Discussion

In this section we first describe the user interface we implemented to demonstrate the approach. Second, we compare different term selection approaches. Due to

⁸ <http://www.oefai.at/~elias/wa/dict.txt>

space limitations we will use simple lists of words to do so (see Table 1). Finally, we discuss our approach in general.

6.1 User Interface

To demonstrate our approach we implemented a very simple HTML interface.⁹ There are two parts of the interface: the hierarchy of clusters visualized as a grid of boxed texts and, just to the right of it, a display of a list of artists mapped to the currently selected cluster. The clusters of the first level in the hierarchy are visualized using the five boxes in the first (top) row. After the user selects a cluster, a second row appears which displays the children of the selected cluster. The selected clusters are highlighted in a different color. The hierarchy is displayed in such a way that the user can always see every previously made decision on a higher level. The number of artists mapped to a cluster is visualized by a bar next to the cluster. Inside a text box, at most the top 10 terms are displayed. However, if a term's value is below 10% of the highest value then it is not displayed. The value of the ranking function for each term is coded through the color in which the term is displayed. The best term is always black and as the values decrease the color fades out. For debugging purposes it is also possible to display the list of all ranked words for a cluster. Figure 3 shows what the user interface looks like (using LK labeling) after node n2.5.1 was selected (thus 4 levels are visible).

6.2 Comparison of Term Selection

Table 1 lists all top-ranked words for the different approaches at the first level and some examples of the second level (for the children of node n5.1). Comparing this table with Figures 1 and 2 shows how different types of genres are described. For example, in most cases describing the classical cluster (node n1) works very well. In general we have made the following observations:

First, the results using the dictionary are better in most cases at the first level. The main difference becomes clearer at the second level (for the children of node n5). In particular, using the dictionary avoids the frequent appearance of artist names.

Second, not all words in the domain specific dictionary make sense. Although not directly noticeable at the first level there are some words which appear frequently in the top-ranked words but do not convey much information: *world*, *uk*, *band*, *song*, *musical*. On the other hand, from studying the lists we noticed that words such as *love* and *hate* are missing in our dictionary. Having a few meaningless words is not such a big problem. In an interactive interface the user could just click on the words to remove and the interface could be updated immediately. However, adding missing words to the dictionary is a bit more complex and requires scanning all the retrieved documents for occurrences.

Third, the performance of the non discriminating approaches ($tf \times idf$, Label-SOM) is very poor. On the other hand, all discriminative approaches (χ^2 , LK,

⁹ <http://www.oefai.at/~elias/wa>

Table 1. List of top ranked terms for selected nodes

	with dictionary	without dictionary
<i>tf × idf</i>		
n1	classical, piano, orchestra, symphony, musical	classical, piano, orchestra, works, composer
n2	song, pop, world, uk, band	listen, color, news, pop, size
n3	song, band, pop, world, guitar	band, listen, great, pop, live
n4	song, band, guitar, pop, world	color, listen, live, pop, size
n5	song, blues, band, guitar, world	color, size, family, listen, blues
LabelSOM		
n1	world, musical, concert, song, uk	two, information, musical, recordings, great
n2	world, musical, classical, song, real	content, know, people, listen, sound
n3	musical, world, song, group, pop	great, sound, news, listen, live
n4	musical, world, classical, song, pop	content, great, information, pop, listen
n5	musical, classical, group, song, world	information, great, content, listen, pop
χ^2		
n1	piano, orchestra, symphony, concert, opera	classical, composer, musical, great, piano
n2	dance, rap, hip-hop, beats, group	news, pop, sound, track, release
n3	guitar, musical, group, punk, metal	band, sound, live, great, pop
n4	musical, guitar, country, group, blues	live, band, pop, news, policy
n5	blues, band, country, pop, jazz	blues, jazz, country, hits, policy
Lagus & Kaski		
n1	piano, symphony, classical, orchestra, opera	op, bach, piano, symphony, classical
n2	song, rap, hip-hop, pop, dance	hop, hip, rap, listen, pop
n3	band, punk, metal, song, pop	band, punk, metal, bands, great
n4	country, song, band, guitar, pop	country, alice, elvis, brooks, rate
n5	blues, jazz, country, soul, reggae	blues, jazz, color, john, size
χ^2 : LK		
n1	piano, orchestra, symphony, opera, violin	classical, piano, composer, orchestra, symphony
n2	rap, dance, hip-hop, beats, uk	news, pop, hop, hip, track
n3	punk, guitar, metal, musical, group	band, punk, live, sound, great
n4	country, guitar, musical, group, blues	country, live, band, pop, hits
n5	blues, jazz, country, band, soul	blues, jazz, country, john, hits
Lagus & Kaski variant		
n1	rondo, fortepiano, contralto, fugue, mezzo	nabucco, leopold, cycles, figaro, sonatas
n2	hardcore techno, latin rap,	pies, grandmaster, hash, tricky, pimp
n3	southern rap, east coast rap	
n4	pop-metal, melodic metal, detroit rock,	roisin, pies, hash, dez, voulez
n5	flamenco guitar, math rock	
n6	new traditionalist, british folk-rock,	csn, dez, voulez, shapes, daltrey
n7	progressive bluegrass, gabba, slowcore	
n8	rockabilly revival, new beat, progressive	hodge, precious, shanty, broonzy, dez
n9	country, vocalion, freakbeat	
χ^2 : LK variant		
n1	piano, orchestra, symphony, opera, violin	classical, symphony, composer, piano, orchestra
n2	rap, hip-hop, beats, dance, cool	hop, hip, rap, eminem, dj
n3	punk, metal, guitar, punk rock, hard	band, punk, metal, bands, live
n4	country, guitar, country music, folk, group	country, brooks, elvis, dylan, hits
n5	blues, jazz, country, soul	blues, jazz, willie, otis, john
Lagus & Kaski		
n5.1	blues, jazz, guitar, band, orchestra	blues, jazz, john, coltrane, basie
n5.2	soul, blues, song, gospel, pop	aretha, soul, redding, king, franklin
n5.3	reggae, ska, song, world, dancehall	marley, reggae, tosh, cliff, baez
n5.4	country, country music, song, bluegrass, folk	country, hank, elvis, cash, kenny
n5.5	band, song, pop, guitar, blues	elvis, roll, rate, band, bo
LK variant (with dictionary)		
n5.1	hot jazz, post-bop, vocalion, rondo, soul-jazz, classic jazz, hard bop, superstitious, octet	
n5.2	british blues, pornographic, colored, classic soul, sensual, erotic, precious, rap rock, stylish	
n5.3	vocal house, soca, british punk, gong, ragga, ska, dancehall, dancehall reggae, hard house	
n5.4	new traditionalist, yodelling, middle aged, country boogie, outlaw country, rockabilly revival	
n5.5	experimental rock, boogie rock, castanets, psychedelic pop, pagan, dream pop, crunchy	

and combinations) yield interesting results with the dictionary. However, the LK variant by itself focuses too much on the differences. Obviously, to truly judge the quality of the different variations would require user studies. Subjectively our impression was that the approach from Lagus & Kaski performed slightly better than the others.

6.3 Discussion

One of the main problems is that our approach relies on artist names. In many cases this name might have several meanings making it difficult to retrieve relevant webpages. Another problem is that many new and not so well known artist do not appear on webpages. This limits our approach to yesterday's mainstream western culture. This limitation is also underlined by the dictionary we use which contains terms mainly used in our culture. However, the dictionary could easily be replaced. Another issue is the dynamics of web contents (e.g. [26]). We studied this in [8] and the study was continued in [27]. So far we observed significant changes in the Google ranks, but these did not have a significant impact on the similarity measure.

7 Conclusions

In this paper we demonstrated possibilities to hierarchically organize music at the artist level. In particular we suggested using hierarchical clustering with overlapping clusters which are described using a domain-specific dictionary. The results are very promising, however, we fail to present a thorough evaluation. In future work we plan to conduct small scale user studies and combine this approach with other approaches based on audio signal analysis.

Acknowledgments

This research was supported by the EU project SIMAC (FP6-507142). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministries bm:bwk and bmvit.

References

1. Pachet, F., Cazaly, D.: A taxonomy of musical genres. In: Proc RIAO Content-Based Multimedia Information Access (2000)
2. Pachet, F., Westerman, G., Laigre, D.: Musical data mining for electronic music distribution. In: Proc WedelMusic Conf (2001)
3. Whitman, B., Lawrence, S.: Inferring descriptions and similarity for music from community metadata. In: Proc Intl on Computer Music Conf (2002)
4. Baumann, S., Hummel, O.: Using cultural metadata for artist recommendation. In: Proc WedelMusic Conf (2003)

5. Zadel, M., Fujinaga, I.: Web services for music information retrieval. In: Proc Intl Conf Music Information Retrieval (2004)
6. Schedl, M., Knees, P., Widmer, G.: A web-based approach to assessing artist similarity using co-occurrences. In: Proc Workshop Content-Based Multimedia Indexing (2005)
7. Ellis, D., Whitman, B., Berenzweig, A., Lawrence, S.: The quest for ground truth in musical artist similarity. In: Proc Intl Conf Music Information Retrieval (2002)
8. Knees, P., Pampalk, E., Widmer, G.: Artist classification with web-based data. In: Proc Intl Conf Music Information Retrieval (2004)
9. Whitman, B., Ellis, D.: Automatic record reviews. In: Proc Intl Conf Music Information Retrieval (2004)
10. Logan, B., Kositsky, A., Moreno, P.: Semantic analysis of song lyrics. In: Proc IEEE Intl Conf Multimedia and Expo (2004)
11. Whitman, B., Smaragdis, P.: Combining musical and cultural features for intelligent style detection. In: Proc Intl Conf Music Information Retrieval (2002)
12. Baumann, S., Pohle, T., Shankar, V.: Towards a socio-cultural compatibility of MIR systems. In: Proc Intl Conf Music Information Retrieval (2004)
13. Pampalk, E.: Islands of music: Analysis, organization, and visualization of music Archives. MSc thesis, Vienna University of Technology (2001)
14. Pampalk, E., Rauber, A., Merkl, D.: Content-based organization and visualization of music archives. In: Proc ACM Multimedia (2002)
15. Rauber, A., Pampalk, E., Merkl, D.: Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarities. In: Proc Intl Conf Music Information Retrieval (2002)
16. Schedl M.: An explorative, hierarchical user interface to structured music repositories. MSc thesis, Vienna University of Technology (2003)
17. Pampalk, E., Hlavac, P., Herrera, P.: Hierarchical organization and visualization of drum sample libraries. In: Proc Intl Conf Digital Audio Effects (2004)
18. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. *Information Processing and Management* **24** (1988)
19. Kohonen, T.: *Self-Organizing Maps*. Springer (2001)
20. Miikkulainen, R.: Script recognition with hierarchical feature maps. *Connection Science* (1990)
21. Koikkalainen, P., Oja, E.: Self-organizing hierarchical feature maps. In: Proc Intl Joint Conf Neural Networks (1990)
22. Dittenbach, M., Merkl, D., Rauber, A.: The growing hierarchical self-organizing map. In Proc Intl Joint Conf Neural Networks (2000)
23. Rauber, A.: LabelSOM: On the labeling of self-organizing maps. In: Proc Intl Joint Conf Neural Networks (1999)
24. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proc Intl Conf Machine Learning (1997)
25. Lagus, K., Kaski, S.: Keyword selection method for characterizing text document maps. In: Proc Intl Conf Artificial Neural Networks (1999)
26. Lawrence, S., Giles, C. L.: Accessibility of information on the web. In: *Nature* (1999)
27. Knees, P.: Automatic classification of musical artists based on web-data (in German). MSc thesis, Vienna University of Technology (2004)

A Comparison of Melodic Segmentation Techniques for Music Information Retrieval

Giovanna Neve and Nicola Orio

Department of Information Engineering – University of Padova,
Via Gradenigo, 6/b – 35131 Padova, Italy
{mcic, orio}@dei.unipd.it

Abstract. The scientific research on accessing and retrieval of music documents is becoming increasingly active, including the analysis of suitable features for content description or the development of algorithms to match relevant documents with queries. One of the challenges in this area is the possibility to extend textual retrieval techniques to music language. Music lacks of explicit separators between its lexical units, thus they have to be automatically extracted. This paper presents an overview of different approaches to melody segmentation aimed at extracting music lexical units. A comparison of different approaches is presented, showing their impact on indexes size and on retrieval effectiveness.

1 Introduction

Users of digital libraries may have different knowledge on the particular application domain for which the library has been created. This is particularly true for music language, where the level of music education may vary remarkably among users, who may range from the totally untrained listeners of Italian opera to the renown performer and composer of jazz music. The user's knowledge is related to the ability to describe his information needs, because he could be unable to use bibliographic values or take advantage of metadata when searching for music information. Hence, music library access is usually content-based.

The main idea underlying content-based approaches is that a document can be described by a set of features that are directly computed from its content. Usually, content-based access to multimedia data requires specific methodologies that have to be tailored to each particular media. Yet, the core information retrieval techniques, which are based on statistics and probability theory, may be more generally employed outside the textual case, because the underlying models are likely to describe fundamental characteristics being shared by different media, languages, and application domains [11]. For the particular case of music language, already in 1996 McLane stated that a challenging research topic would have been the application of some standard principles of text information retrieval to music representation [9].

As it is well known, textual information retrieval is based on the concept that *words*, which form a document, can be considered as a good descriptor of its content. Following this idea, documents can be efficiently indexed using words

as *index terms*, and retrieval can be performed using a measure of similarity between query and documents indexes. If we follow the hypothesis that these principles can be extended to indexing and retrieving music documents, then the design of ad-hoc segmentation algorithms to produce musical *lexical units* is required. This paper reports a study on the effects of different segmentation techniques on content-based retrieval of music documents.

2 Content-Based Access to Music Digital Libraries

Most of the approaches to content-based access to music documents focus on the melody as the only content descriptor. Melody is an important feature to describe a music information need because, for untrained users, it is the simplest and more comprehensible dimension. This choice also depends on the typical interaction paradigm that is used to query a system, the *query by example* paradigm, which requires the user to give an example of his information need by singing part of a song. Query by example is not the only approach to music accessing. Other paradigms include documents similarity, music recommendation, browsing music collections. Yet the query by example approach allows for extensive testing using well known evaluation techniques.

The research work on music information retrieval can be roughly divided in two categories: *on-line searching techniques*, which compute a match between the string representing the query and the ones representing the documents each time a new query is submitted to the system, and *indexing techniques*, which extract off-line all the relevant information that is needed at retrieval time and perform the match directly between query and documents indexes. Only the second category is related to the present work.

Approaches to indexing usually differ in the way lexical units are computed and how they are eventually normalized to overcome sources of mismatch between documents and queries. In [3] work melodies have been indexed through the use of N-grams, each N-gram being a sequence of N pitch intervals, while note duration was not used as a content descriptor. Another approach to document indexing has been presented in [8], where indexing has been carried out by automatically highlighting lexical units using an automatic segmentation algorithm based on music theory. Units could undergo a number of different normalization, from the complete information of pitch intervals and duration to the simple melodic profile. Melodic and rhythmic patterns have been used in [10] as lexical units, without using knowledge on music structure or cognition.

3 Approaches to Melodic Segmentation

Differently from text, music is a continuous flow of events without explicit separators. It is therefore necessary to automatically detect the lexical units of a music document to be used as index terms. Different strategies to melodic segmentation can be applied, each one focusing on particular aspects of melodic information. In the present work some approaches are considered.

Fixed-Length Segmentation (FL). The simplest segmentation approach consists in the extraction from a melody of subsequences of exactly N notes, called N-grams. N-grams may overlap, because no assumption is made on the possible starting point of a theme, neither on the possible repetitions of relevant music passages. The strength of this approach is its simplicity, because it is based neither on assumption on theories on music composition or perception, nor on analyzes of the complete melody.

The exhaustive computation of FL units is straightforward, and can be carried out in linear time. The idea underlying this approach is that the effect of musically irrelevant N-grams will be compensated by the presence of all the musically relevant ones. It is common practice to choose small values for N , typically from 3 to 7 notes, because short units give higher recall, which is considered more significant than the subsequent lowering in terms of precision. Fixed-length segmentation can be extended to polyphonic scores, with the aim to extract all relevant monophonic tokens from concurrent multiple voices [2].

Data-Driven Segmentation (DD). Segmentation can be performed considering that typical passages of a given melody tend to be repeated many times. The repetitions can simply be due to the presence of different choruses in the score or can be related to the use of the same melodic material along the composition. Each sequence that is repeated at least K times – normally twice – is usually defined a *pattern*, and is used for the description of a music document. This approach is called data-driven because patterns are computed only from the document data without exploiting knowledge on music perception or structure.

This approach can be considered as an extension of the N-grams approach, because DD units can be of any length, with the limitation that they have to be repeated inside the melody – subpatterns that are included in longer patterns are discarded, if they have the same multiplicity. Patterns can be computed from different features, like pitch or rhythm, each feature giving a different set of DD units to describe a document content. Patterns can be truncated by applying a given threshold, because it is unlikely that a user will remember long sequences of notes. The extension to polyphonic scores can be carried out similarly to the N-grams approach.

Perception-Based Segmentation (PB). Melodies can be segmented accordingly to theories on human perception. Listeners have the ability to segment the unstructured auditory stream into smaller units, which may correspond to melodic phrases, motifs, or musical gestures. Even if listeners may disagree on the exact location of boundaries between subsequent units [7], it is likely that perceptually-based units are good descriptors of a document content, because they capture melodic information that appears to be relevant for users.

The ability of segmenting the auditory stream may vary depending on the level of musical training of listeners and their knowledge of rules on music theory. Yet, a number of strategies can be generalized for all listeners, in particular the ones related to the detection of clear changes in the melodic flow such as large pitch intervals or note durations. This behavior can be partially explained by the

principles of Gestalt psychology. Computational approaches have been proposed by music theorists for the automatic emulation of listeners behavior [12]. PB units do not overlap and are based on information on note pitch and duration of monophonic melodies.

Musicological-Oriented Segmentation (MO). Another approach to segmentation is based on knowledge on music theory, in particular for classical music. According with music theorists, music is based on the combination of musical structures [5], even if its actual notation may lack of clear representations of such structures. Yet, they can be inferred by applying a number of rules, and part of the analysis of compositions consists in their identification. It is likely that the same approach can be extended also to less structured music, like popular or ethnic music.

It is assumed that a hierarchical relationship exists among music structures, from musical phrases at the lower level to movements at the higher level. MO units are computed by analyzing the musical score, applying rules for structure identification, and segmenting the score in units that correspond to low-level structures. The computation of MO units should be carried out using the global information of the score, but it has been proposed an algorithm which uses only local information and gave results comparable to more complex ones [1]. Structures may overlap in principle, but the current implementations do not take into account this possibility.

Mixed Segmentation and Data Fusion (FUS). Each segmentation technique allows for describing the melody with a set of note sequences. As it is well known, the melody contains different music dimensions, in particular pitch and rhythm, and it is necessary to choose which dimension has to be used for the representation of lexical units. Data fusion techniques can then be exploited to merge different representations, as proposed in [10]. That is, the documents are indexed using units with only rhythmic or pitch information, or the combination of the two. Each indexing scheme retrieves a different rank list, and the individual results can then be merged in a single rank list that is presented to the user. This approach may improve retrieval performances, and thus in our experiments it is used as a reference.

A set of preliminary tests highlighted the characteristics that are more related to improvements on retrieval effectiveness. Results showed that the use of short and overlapping units improved recall, while the use of longer units improved precision. The two positive effects can be obtained by using overlapping units with different lengths and not removing subpatterns. This approach to segmentation is halfway between FL and DD.

Query Segmentation (QS). Content-based retrieval requires that both documents and queries are processed correspondingly – the same segmentation is applied to documents and queries – in order to compute a measure of similarity between them. Yet, query processing has some limitations: users normally

provide very short examples, which from the one hand do not allow for the computation of patterns, and from the other hand do not guarantee that the query boundaries correspond to patterns, perceptual units, or musicological structures.

With the aim of partially overcome this problem, the extraction of lexical units from queries can be carried out taking all possible sequences of notes with the same lengths of segmented units. Most of these sequences will be completely irrelevant musically, giving no contribution to the similarity computation. On the other hand, this exhaustive approach guarantees that all possible lexical units are used to query the system.

4 Experimental Comparison of Segmentation Techniques

The comparison has been carried out according to the Cranfield model for information retrieval. A music test collection of popular music has been created with 2310 MIDI files as music documents. The channels containing the melody have been extracted automatically and the note durations have been normalized; the highest pitch has been chosen as part of the melody for polyphonic scores. After preprocessing, the collection contained complete melodies with an average length of 315.6 notes. A set of 40 queries, with average length of 9.7 notes, has been created as recognizable examples of both choruses and refrains of 20 randomly selected songs. Only the theme from which the query was taken was considered as relevant. The robustness to errors has been tested by modifying notes pitch and duration, while the effects of query length has been tested by shortening the original queries. All the tests have been carried out using the same retrieval engine, which is based on the Vector Space Model and implements the *tf · idf* weighting scheme.

Tab. 1 shows the main characteristics of lexical units extracted with the segmentation approaches. FL has been computed with N-grams of three notes, DD has been computed applying a threshold of five notes, while PB and MO have been computed using the algorithms presented in [4]. For these four approaches, units were sequences of couples of values, pitch and duration, and the index is built with one entry for each different sequence. FUS has been computed using overlapping units from two to four notes, and fusing the results of the three indexing scheme based on only rhythm, only pitch, and the combination of the two. Given the use of data fusion on alternative representations of the same documents, FUS cannot be directly compared with the others. Yet it is presented

Table 1. Main characteristics of the lexical units for the different segmentations

	FL	DD	PB	MO	FUS
Average length	3.0	4.8	4.0	3.6	3
Average units/document	52.1	61.9	43.2	45.0	132.4
Number of units	70093	123654	70713	67893	253783

Table 2. Retrieval effectiveness of the different approaches

	FL	DD	PB	MO	FUS
Av.Prec.	0.98	0.96	0.80	0.83	1.0
= 1	97.5	92.5	72.5	77.5	100
≤ 3	97.5	100	87.5	87.5	100
≤ 5	97.5	100	87.5	92.5	100
≤ 10	100	100	90.0	95.0	100
not found	0.0	0.0	10.0	2.5	0.0

as a reference, as the best results that have been obtained in our experiments with document indexing.

The approaches gave comparable results in terms of average length of lexical units, which is about three-four notes, and also in the average number of different units per document. Tab.1 gives a preliminary idea on how each segmentation approach describes documents. The last row reports the number of different units that corresponds to the number of entries in the index file. As it can be seen, segmentation with overlapping units with different lengths (DD and FUS) has the drawback of an increase of the index size. Obviously FUS had a very high number of units, because they are the sum of three different indexing schemes, and hence it has the drawback of higher memory requirements.

The results in terms of retrieval effectiveness are presented in Tab. 2, where the average precision (Av.Prec.), the percentage queries that gave the r-doc within the first k positions (with $k \in \{1, 3, 5, 10\}$), and the ones that did not retrieve the relevant document at all (“not found”), are reported.

FL and DD had comparable results, because they have close average precision and both always retrieved the relevant document within the first ten positions (FL within the first three, but with a lower percentage of queries that retrieved the relevant document at top rank). Also PB and MO had comparable results in terms of average precision, with slightly better performances of MO, in particular because PB did not retrieve at all the relevant document in 10% of the queries. This is a negative aspects of PB, due to the fact that its units do not overlap and, for short queries, it may happen that none of the note sequences match with the segmented units. A similar consideration applies also to MO, but this effect seems to be bounded by the fact that MO units are shorter. FUS gave very good results, with the relevant document always retrieved at top rank.

The performances of the different approaches depending on the presence of errors in the query are presented in Fig. 1, which shows the average precision of the approaches. Apart from FL and FUS, the other segmentation techniques had a clear drop in the performances, also when a single error was introduced. In particular, PB and MO showed a similar negative trend, almost linear with the number of errors. It is interesting to note that DD, even if its performances are almost comparable to FL in the case of a correct query, had a faster degradation in performances. FUS seemed to be the most robust to errors, probably because of the alternative representations that are used together.

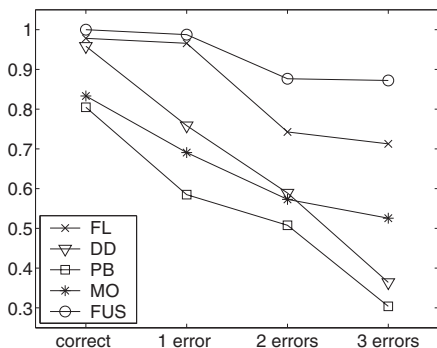


Fig. 1. Av. Prec. of queries with errors

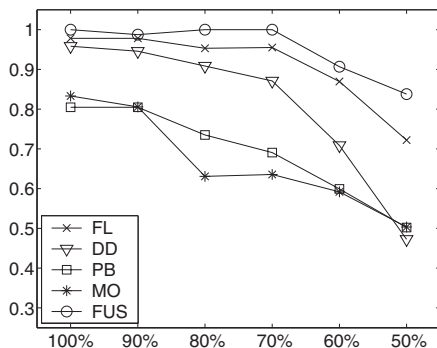


Fig. 2. Av. Prec. of short queries

The average precision depending on query length is shown in Fig. 2. Similar considerations can be made on the trends of the different approaches. PB and MO had a similar behavior, and also in this case FL was the one with the best performances. It can be noted that, when the queries are moderately shortened, the average precision of FL and DD is almost constant. The drop in performances appears earlier, and more remarkably, for DD than for FL. Again, FUS showed to be robust also to this modification.

5 Conclusions

Content-based music accessing and retrieval is still in its early age. A number of methodologies and techniques are the subject of scientific research, from the identification of suitable features to be used as content descriptors to the development of efficient algorithms to match potentially relevant documents with users queries. In this scenario, the exploration of how classical textual retrieval techniques can be applied seems promising, providing that index terms can be automatically extracted from music documents.

This paper presents an overview of different approaches to segmentation for extracting music lexical units, from fixed-length and data-driven segmentations to perceptually or theoretically based segmentations. Moreover, a mixed approach is proposed for segmenting the documents in units related to alternative features, which are merged together with a data fusion approach.

According to our experiments, simple approaches to segmentation with overlapping units gave better performances than approaches based on music perception or music theory. Moreover, fixed-length segmentation was more robust to errors in the queries and to short queries than data-driven segmentation. From these results, it seems that for music a simple approach, which does not filter out any information, improves recall without degrading precision. The good performances of a mixed approach, combined with data fusion techniques, seem to confirm this characteristic. That is, for a music retrieval task the replication of information helps improving consistently the system performances.

Acknowledgments

The work is partially supported by the DELOS Network of Excellence on Digital Libraries, as part of the Information Society Technologies (IST) Program of the European Commission (Contract G038-507618).

References

1. Cambouropoulos, E.: Musical Rhythm: a Formal Model for Determining Local Boundaries In Leman, M. (Ed.) *Music, Gestalt, and Computing*, Springer Verlag, Berlin (1997) 277–293
2. Doraisamy, S., R uger, S.: A Polyphonic Music Retrieval System Using N-Grams In *Proc. of the International Conference on Music Information Retrieval*, Barcelona, ES (2004) 204–209
3. Downie, S., Nelson, M.: Evaluation of a Simple and Effective Music Information Retrieval Method In *Proc. of the ACM-SIGIR Conference*, Athens, GR (2000) 73–80
4. Eerola, T., Toiviainen, P.: MIR in Matlab: The Midi Toolbox In *Proc. of the International Conference on Music Information Retrieval*, Barcelona, ES (2004) 22–27
5. Lerdhal, F., Jackendoff, R. *A Generative Theory of Tonal Music* MIT Press, Cambridge, MA (1983)
6. Meek, C., Birmingham, W.: Automatic Thematic Extractor *Journal of Intelligent Information Systems*, Vol. 21, Issue 1 (2003) 9–33
7. Melucci, M., Orio, N.: Evaluating Automatic Melody Segmentation Aimed at Music Information Retrieval In *Proc. of the ACM/IEEE Joint Conference on Digital Libraries*, Portland, OR (2002) 310–311
8. Melucci, M., Orio, N.: Combining Melody Processing and Information Retrieval Techniques: Methodology, Evaluation, and System Implementation *Journal of the American Society for Information Science and Technology*, Wiley, Vol. 55, Issue 12 (2004) 1058–1066
9. McLane, A.: Music as information In Williams, M. (Ed.) *Annual Review of Information Science and Technology (ARIST)*, American Society for Information Science, Vol. 31, chapter 6 (1996) 225–262
10. Neve, G., Orio, N.: Indexing and Retrieval of Music Documents through Pattern Analysis and Data Fusion Techniques In *Proc. of the International Conference on Music Information Retrieval*, Barcelona, ES (2004) 216–223
11. Sparck Jones, K., Willett, P.: *Readings in Information Retrieval* Morgan Kaufmann, San Francisco, CA (1997)
12. Tenney, J., Polansky, L.: Temporal Gestalt Perception in Music *Journal of Music Theory*, Vol. 24, Issue 2 (1980) 205–241

The Effect of Collection Fusion Strategies on Information Seeking Performance in Distributed Hypermedia Digital Libraries

Michail Salampasis¹ and John Tait²

¹Department of Informatics, Technological Educational Institute of Thessaloniki, Greece
cs1msa@it.teithe.gr

²School of Computing and Technology, University of Sunderland, St Peters Campus,
SR6 ODD, Sunderland, UK

Abstract. This paper reports the results of a user-centered experiment which examined the effect of parallel multi-database searching using automated collection fusion strategies on information seeking performance. Three conditions were tested in the experiment. Subjects in the first condition performed search tasks in a WWW-based distributed hypermedia digital library which did not support parallel, concurrent searching of multiple collections, and did not offer any automated mechanism for source selection. Subjects in the second and the third conditions performed parallel multi-database search tasks in the same library with the support of two automated collection fusion strategies (uniform and link-based), each solving the collection fusion problem using a different approach. The results show that information-seeking performance tends to be positively affected when the eclectic link-based method was used. On the other hand, the uniform collection fusion method which treats all the sub-collections in the same manner, does not present any benefit in comparison to information seeking environments in which users must manually select sources and parallel multi-database searching is not provided.

1 Introduction

Hypermedia Digital Libraries¹ (HDLs) are digital libraries based on a hypermedia paradigm, in other words collections of directly inter linked multimedia documents as opposed to collections of documents linked only indirectly by catalogues or indices. Taking the definition of DLs as networked information systems, the first DL which was extensively used is the World Wide Web (WWW). The WWW also illustrates the problems of distribution and scale which must be addressed in real HDLs.

The feature that differentiates hypermedia digital libraries from other types of DLs (i.e. digital libraries based on Information Retrieval systems or on distributed databases), is that HDLs explicitly support intuitive, opportunistic, across-document

¹ In this work we will use the definition of digital libraries which considers them to be primarily networked information systems perhaps at the expense of ignoring issues like acquisition control, cataloguing, copyright & pricing issues, navigational structure or acquiring knowledge.

browsing strategies for information seeking. In addition to browsing, HDLs usually support analytical (i.e. query-based) strategies because these are more efficient in large electronic environments. Hence, users in HDLs can employ different information-seeking strategies and engage in rich and complex interactions to achieve their goals (Bonder et al, 2001). The continuing move towards environments which effectively support different search strategies in combination has been apparent for some time (e.g. Marchionini, 1995 pp 8; Lai, Tait and McDonald, 2000; Börner, 2000; Wu et al. 2001; Zhang & Marchionini, 2004). Furthermore, there are indications that information seekers prefer electronic environments which support both analytical searching and browsing or generally support multiple strategies (Salampasis & Diamantaras, 2002). From the same experiments we have observed that information seekers use a combination of *roughly* locating information by query searching, and then, *specifically* accessing by browsing.

The short discussion above suggests that although hypermedia systems use browsing to satisfy the requirement of ease-of-use and higher interactivity, analytical searching in large electronic environments is also important for effective and efficient information seeking. In fact, query-based analytical searching was used for many years as the primary means for information seeking in large collections of unstructured documents. However, digital libraries are very different from conventional electronic environments and therefore analytical searching should be approached carefully. For example, conventional Information Retrieval (IR) methods and algorithms deal mostly with the problem of searching a single collection. On the other hand, in distributed digital libraries multiple collections coexist and this has some important implications and uncovers problems and challenges that do not exist in conventional IR. The goal of Distributed Information Retrieval (DIR) is to produce effective and efficient solutions to the problems highlighted above.

The term *collection fusion problem* (Voorhees et al, 1995) denotes the problem of DIR residing, first, in the selection of sources (from the many available) likely to provide relevant information, and second, in the production of a single, combined search result which can be effectively examined by information seekers. These two problems are very important because they directly relate to two (selecting a source and examining results) of Marchionini's (1995, p.49) seven basic subprocesses of the information seeking process. Consequently they directly influence the effectiveness and the efficiency of information seeking.

Several collection fusion strategies have been reported in the literature (e.g. Voorhees et al, 1995; Salampasis & Tait, 1999; Calvé and Savoy, 2000; Deok-Hwan and Chung, 2003 Rasolofu et al, 2003). Their performance was evaluated using the "traditional" method of ad-hoc system-centered experiments. The results showed that the retrieval results from multiple collections (i.e. multiple indexes) using collection fusion strategies are worse than the results obtained using a single collection (i.e. single index). On the other hand, these and other experiments also demonstrate that it is possible to devise a collection fusion strategy which allows multiple collection searches to outperform those for single collections (e.g. Voorhees et al, 1995). Indeed, the optimal fusion method which gives the upper bound for any collection fusion strategy, consistently performed better than the single collection run in the above experiments.

The main objective of this paper is to report the results of a *user-centered* experiment which aimed to assess the effect of collection fusion strategies in HDLs and to compare

them with a similar HDL in which users must perform the source selection manually. The fusion strategies which were used were the uniform method and a link-based strategy. As the name implies the uniform method assumes that the relevant documents corresponding to a particular query are identically distributed across all document collections. Hence, an equal number of documents are requested from each collection. The link-based fusion strategy utilises links between hypermedia documents to solve the collection fusion problem. More precisely, it operates in three stages and tries to approximate the distribution of relevant documents in disparate collections by using the distribution of links from a sampling collection to each remote collection. The experiment compares the performance of parallel searching, in which users can search multiple collections and examine a single, merged result, with the performance of collection searching in which users must manually select and search a single collection.

2 Distributed vs. Centralised Searching

There are two different system architectures which can be used to conduct analytical searches in digital libraries. The first is to maintain a centralised index built by systematically and exhaustively downloading (crawling) and indexing the documents from remote collections to a central repository. Examples using this approach include all the well known WWW search engines like Google (www.google.com) and AllTheWeb (www.allthe-web.com).

The second approach is the distributed index design. In this design each participating collection in a DL maintains its own index and uses its own methods to conduct the analytical searches, for example Dienst (Davies & Lagoze, 1994). This approach is sometimes called metasearching (Yu et al, 1999). A number of metasearchers have been available in the WWW for some time now with the MetaCrawler (Selber & Etzioni, 1997) being probably the best well know example.

In this paper we are concerned with electronic environments based on a distributed index design. Although centralised systems have advantages in areas like consistency and search time efficiency, they also have significant limitations (Rasolofotry et al, 2003). First many digital libraries (web sites) may deny access to spiders crawling documents. Second, centralised indices are difficult to maintain and update. For example, a complete crawl is very expensive and in the case of the WWW impracticable. Distributed systems have many advantages including simpler index maintenance, availability, support for different indexing strategies based on the content of the collection and so on.

One problem posed by searching multiple distributed collections is the source selection problem, i.e. how to select the sources, from the many available, to which to submit the queries and on which to conduct the searches. The second problem is how to combine the individual results to produce a single result so that the most relevant documents as high as possible in the single ranked list which will be examined by the information seekers.

There are collection fusion strategies which uniformly retrieve documents from all the collections available. On the other hand, there are collection fusion strategies which operate in a more eclectic way, i.e. they seek to retrieve documents from a small proportion of the available collections. The underlying hypothesis in these eclectic methods is that relevant documents will not usually be equally distributed in

all collections, but will be collocated in a few of them. In the next section we detail two such collection fusion strategies before presenting a user-centered experiment which may be useful in examining some of the above questions.

3 Collection Fusion Strategies

The experiments described in the next section compare single index searching using manual selection of which source to search, with two collection fusion strategies: a link-based strategy and a so-called uniform strategy.

The uniform strategy is fairly straightforward. The query is broadcast to each collection (or individual HDL) with an indication of the number of the maximum number apparently relevant documents to be returned. Each HDL then returns this number (unless too few are found) and the results are merged using a result merging algorithm (e.g. simple round robin). That is, in the final merged result (using a simple round robin merging method), the first document from collection one appears first, then the first document from collection two, and so on until the first document from each collection appears, then each the second document from the first collection and so on.

The Link-based strategy is more complex and has not been previously described so we will provide a detailed description of it in the following sections. It relies on what Salampasis has previously called the *link hypothesis* (Salampasis, 1999), a development of the *cluster hypothesis* of Jardine and van Rijsbergen (1971). The link hypothesis is that closely interlinked documents tend to be relevant to the same information need.

3.1 First Phase: Extraction of Linkage Information

We begin by retrieving an initial set of documents from a *sampling collection* which will form a seed for subsequent phases. In practice it doesn't matter which individual HDL (or collection) is used as a sampling collection but it must have adequate number of outgoing links to other sub-libraries.

The next step is to extract and calculate the number of links having as their starting points one of the documents retrieved from the sampling collection and to pass this to the approximation function in the second phase of the fusion method. Clearly, the number of links is proportional to the number of documents retrieved from the sampling collection and may affect the performance of the method. Extraction of a large number of links increases the possibility of more collections finally being selected for the distributed searching. In principle, more libraries means increased effectiveness, but also decreased efficiency.

3.2 Second Phase: Approximation of Relevant Documents Distribution

The second phase of the link-based fusion technique takes as an input the distribution of links starting from documents retrieved from the sampling collection and ending at other hypermedia documents. Essentially this is measure of the number of links from the sampling collection to other HDLs. This information is passed to an approximation function which determines the number of documents that should be requested from each other HDL, given that T documents should be retrieved in total. For example, a simple approximation function can allocate to each collection C_i a number of

documents which is proportional to the number of links pointing to documents in C_i . The expectation (according to the link-hypothesis) is that the number of relevant documents in C_i is proportional to the number of links pointing to documents in C_i .

A suitable approximation function is then as follows. If T is the total number of documents that should be finally retrieved from all collections, the set of link frequencies are used to apportion the retrieved set such that when T documents are to be returned and L_i is the link frequency for collection i , the number of documents to be retrieved by each collection is determined by the formula:

$$\frac{L_i}{\sum_{i=1}^N L_i} * T = \text{Number of documents to retrieve from collection } i \text{ (rounded appropriately)}$$

Using this approximation formula the number of documents N_i to be retrieved from each collection C_i can be determined. The query Q is consequently submitted to all collections C_i with a request to return back N_i documents. Of course, if $N_i=0$ collection C_i will not be involved in the distributed searching. The methods used to conduct the individual searches and to produce the results are not the concern of the fusion strategy. Designers of remote hypermedia collections can decide and apply different retrieval methods.

3.3 Third Phase: Merging Results

In this phase individual results returned back are merged to produce a single result. Multiple methods can be used to merge the individual results and their detailed description is beyond the scope of this paper. In the experiment which is presented here the same merging method was used in both conditions (uniform and link-based).

4 Experiment

4.1 Aim

The primary aim of the experiment was to compare distributed parallel multi-database searching with "single" searching using manual selection in a realistic information-seeking environment. A secondary aim was to compare uniform and link-based fusion strategies in a user-centered experiment.

4.2 Systems and Materials

Three different HDLs were implemented for the experiment. These three HDLs were developed over exactly the same CACM test collection.

CACM is a medium size collection which was chosen because it contains documents having links between them. The documents are generally abstracts of technical papers in computer science, and include titles, author information and some key words, although a very few of them do not include the abstract or keywords. Some of the documents in the CACM collection do not have links with other documents and therefore could not be used in the experiments. A subset of the CACM collection

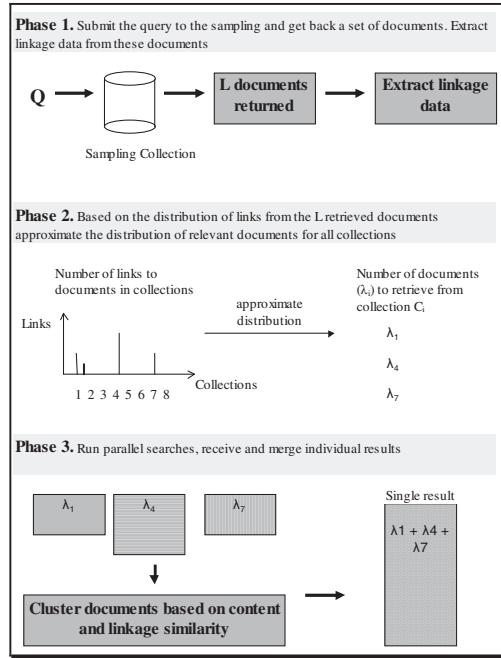


Fig. 1. The three phases of the link-based collection fusion strategy

(CACM_B) which contains all the 1751 documents which have links to other documents in the collection has thus been used.

This standard test collection was combined with a hierarchical clustering process² to produce 8 sub-collections, each playing the role of a distributed autonomous collection. The HDLs used in the experiment were based on the WWW therefore the CACM documents, which were originally ASCII files, had to be converted into HTML. The links between documents in the CACM collection were originally anchored to whole documents (i.e. the start and end points of a link were whole documents, not part of a document). For this reason the corresponding links were embedded into the CACM documents together with the titles of the destination documents to give a prospective view of the latter and therefore to facilitate selection of a link.

Although the three WWW-based HDLs were based on exactly the same raw material, the analytical search method used in each was different, as explained below:

1. Subjects in the first HDL (condition 1) could search for relevant documents using simple across-document browsing, or "single" searching strategies. i.e. they could search only one sub-library (from the eight available) at a time. So, if they wanted to search all sub-libraries for a given query, they had to repeat the query for each sub-library and examine the results separately. They had also manually to select the sources in which they believed they were more likely to find relevant docu-

² The clustering process used is called the Agglomerative Hierarchic Clustering and it is described by Voorhees (1986).

ments. Finally, the subjects could also use a browsable table of contents, which had a link for each document member of a sub-collection, as a navigational aid.

2. The second WWW HDL (condition 2) supported exactly the same information seeking strategies as the first, but subjects could perform distributed, parallel searches over the eight sub-libraries. This meant that subjects could submit just one query to search all the sub-libraries in parallel and could examine one single merged result. The uniform collection fusion strategy was used to solve the collection fusion problem, so all CACM subcollections were equally considered when a multi-database search was made.
3. The third HDL (condition 3) was identical to the second but the link-based collection fusion strategy described earlier was used to solve the collection fusion problem.

Finally, it should be said that except the analytical search methods described above, in all three HDLs there were available to users additional search methods based on browsing strategies. Table 1 shows all the possible states of an information seeker, description of each state and possible explicit movements from one state to another. Of course, state 7 (search multiple collections in parallel) was only available in conditions 2 & 3 (these with parallel searching-automated source selection).

4.3 Method

Thirty-six subjects voluntarily participated in the experiment. Subjects were tested individually. A written description of the WWW-based HDLs was given to the subjects before the tests, to help them gain an overview of the system to be used. Also, a brief, fifteen-minute training session was conducted with each subject before testing, to ensure that s/he could search the HDL and also understood the nature of the task that s/he would be asked to perform. Subjects were divided into three groups. Each group used one of the above HDLs, with each subject being informed about the available information seeking strategies. They were then given an information problem (i.e. a query, expressed as a list of keywords) and asked to find as many relevant documents as possible in 30 minutes using their preferred strategy or a combination of strategies from those which were available. For example one query used was "time sharing operating systems". As a motivation factor, the number of relevant documents was given to the subjects. If a subject found all the relevant documents in less than 30 minutes the search ceased at this point. The subjects were also asked to write down the IDs of the documents that they viewed and judged as being relevant to their query. This list of documents is called the "judgement list" of a subject.

The search sessions were logged and for each search session values for the following measures were calculated:

1. Minutes, the time in minutes that the session actually lasted;
2. First found, the time at which the first relevant document was found;
3. JR, the judged recall at the end of the session;
4. JP, the judged precision at the end of the session;
5. States, the total number of different states (movements) that a searcher moved through during the search session.

Note that the metrics JR and JP refer to "judged" recall and precision, based on the "judgement list" that the subjects produced during the experiment. Of course, it is possible, that subjects have viewed a relevant document without judging it to be so. This document would not be put in the "judgement list". Therefore we also measure the *viewed recall* (VR). Finally, it was possible that relevant documents had been retrieved from an analytical search, but never viewed by the subjects. Therefore, we also measure the *retrieved recall* (RR). Given that a document must be retrieved before it can be viewed, and it must be also viewed before it is written in the "judgement" list, the following equation should always hold:

$$\text{retrieved recall} \geq \text{viewed recall} \geq \text{judged recall}$$

Table 1. States, description and possible explicit movements

State	Description	Movements to state
1. Entry point - list of available HDLs	View all the distributed sub-collections	1, 7
2. View a single collection	View a single collection and the clusters available	3, 4
3. View clusters of documents	See documents which belong to a single collection and to the same cluster	2, 4, 6, 8
4. Search single collection	Submit a query against a single collection	2, 5
5. View	View a ranked list of retrieved documents	4, 6
6. View One document	View one document. This is the "state" in which documents are recognised as relevant	2, 3, 4, 5, 6
7. Parallel searching (available only in conditions 2&3)	Construct a query and submit against multiple collections. Apply collection fusion algorithm	5
8. Browse hierarchical	Browse hierarchically organised themes	3

4.4 Results

Tables 2 and 3 give the results for the search, including the proportion of "analytical" states (both single and parallel distributed searches) and the proportion of browsing states.

Before discussing the results of our experiment in detail we need to make some observations on these results. The experiments presented some difficulties in terms of the availability of subjects and the amount of researcher time required to carry out these experiments. There are a number of factors which were difficult to control, resulting in relatively high standard deviation values, and data which were not always amenable to proper statistical analysis. For example we are requiring subjects to read documents and make a judgement on those documents. All subjects were graduates or studying for a first degree. Nevertheless their reading rates and times taken to make decisions will vary, as will their judgement of the relevance of documents and the strategies they use for searching.

Table 2. Basic results of the search sessions

	Minutes		First Found		States		% Analytical	% Browsing
	mean	sdev	mean	sdev	mean	sdev	mean	mean
Condition 1	26.17	4.06	9.00	5.12	155.75	57.96	32	68
Condition 2	26.64	5.95	8.00	6.26	126.64	49.80	28	72
Condition 3	25.60	6.59	6.10	7.13	129.20	51.67	28	72

Table 3. Performance results of the search sessions

	JR		JP		VR		RR	
	mean	sdev	mean	sdev	mean	sdev	mean	sdev
Condition 1	0.20	0.17	0.33	0.30	0.42	0.26	0.53	0.31
Condition 2	0.18	0.12	0.37	0.32	0.37	0.23	0.42	0.23
Condition 3	0.40	0.28	0.48	0.26	0.64	0.28	0.68	0.26

Given the above reservations we proceed to make some tentative suggestions. If we compare the mean and standard deviation values for the first two conditions we can see that any differences are relatively minor i.e. we cannot state, with any degree of certainty, that there are any differences between the two sets of results. Indeed, the limited statistical analysis we are able to perform would indicate that there are no differences between these two conditions.

The results tend to indicate that parallel searching using the link-based fusion strategy is consistently more effective than searching a single library with manual selection. On the other hand, the results also indicate that some parallel searching strategies may not perform better than the single searching strategy and therefore multi-database searching should be approached with caution. For instance, multi-database searching which treats all the sub-collections the same (i.e. the uniform approach) may not be more effective than single searching. The uniform approach will probably be inefficient. Indeed, searching all the available collections for every information need is likely to be inefficient. Also, the uniform approach will probably increase the information overload for the information seekers who will receive and have to examine results from many, possibly heterogeneous, sources. These reasons possibly explain why parallel searching based on a uniform collection fusion strategy, may be less effective and efficient than a "single" collection searching architecture. The results in Table 2 show that for information seekers in the distributed parallel searching conditions (i.e. 2 & 3) the first document relevant to their information problem is found marginally earlier, although we must note the proviso stated above. Finally, they illustrate that the users in the parallel distributed searching condition tend to go through less states (i.e. make less transitions) during the information seeking process than the users in the single search condition. Finding the first relevant document sooner is useful in information tasks where one or only a few relevant documents can satisfy the information need (precision oriented problems). Also the fact that users in the distributed condition move through fewer states may indicate that these users require less cognitive overhead during the search process than the users in the single searching condition.

From the results outlined above we can generally conclude that parallel distributed searching tends to have a positive effect on an information seeking environment. It can also be concluded that the positive effect is greater in the case of using the eclectic link-based collection fusion strategy. The positive effect applies both to metrics such as R and P which generally reflect the effectiveness of a system, and to other measurements such as the ability to find the first relevant document sooner or to use less states during an information seeking process. These are some of the factors which partially determine the efficiency of an information-seeking environment.

In respect of the second aim of this part of our experiment (i.e. to compare the uniform and the link-based strategies in user-centered evaluations), the results in Table 3 show that the link-based fusion strategy performs better than the uniform fusion strategy. In all cases the link-based strategy has produced better R and P results and users in this condition have generally found their first relevant document earlier, albeit with a large scatter in the results obtained.

5 Discussion and Conclusions

There are a number limitations to the experiments undertaken in this study. They principally concern issues of methodology and issues of scale.

The main limitation of methodology has to do with the artificiality of the methods that have been used to create a distributed HDL from the single CACM collection. Documents were taken from the CACM collection, clustered, and allocated to different sub-libraries using an automatic clustering method. In other words, sub-libraries were not produced manually as would happen in a real environment. We believe that automatically produced sub-libraries do not invalidate the results of the experiments. On the other hand, it might be that the fusion strategy is better at exploiting the properties of the clustering. We have made some effort to distinguish between these hypotheses, but results on the data presented here are inconclusive. More extensive and subtle experimentation would be required to confidently distinguish between these hypotheses. However, in real environments documents are not randomly distributed, documents with similar contents tend to be collocated and our artificially-created distributed libraries are based precisely on this principle. Thus it may be that even if the link-based collection fusion strategy is exploiting the clustering, it may be doing so in a way which is useful for practical, real world searching.

It is also unclear whether we would obtain different results if the experiments had used full text rather than abstracts, or using a wider variety of link types especially combined with full text. Again these remain matters for future work.

The CACM system is not large by modern standards. We hope in the future that similar work can be undertaken using larger collections. In particular the last few years in recent Text Retrieval Conferences (TREC) a new large collection namely WT10g (Bailey et al, 2003) based on a subset of the WWW was used for IR experiments exploiting linkage information such as the experiment presented in this paper. Although the IR community has used extensively this collection recent evidence (Gurrin and Smeaton, 2004) suggests that due to small number of in-degree off-site links WEB-TREC collection can not be used for the type of experiments we present in this paper. In fact, the experiments in recent TREC conferences (TREC 1999,

TREC 2000, TREC 2001, TREC 2002, TREC 2003) trying to exploit linkage information didn't succeed any improvement using linkage information.

However, despite these limitations, and the variability in the data which we have discussed a previously, we believe that the experiment presented here is useful, because it involved a relatively large number of users and minimised (as much as possible) external factors. We are not aware of other user-centered experiments which have evaluated in a such direct manner the effect of distributed parallel searching on information seeking performance, or which have compared two different collection fusion strategies in a realistic environment.

Given the limitations outlined above, these experiments suggest the following:

1. Searching strategies which search multiple collections in parallel to produce a single merged result, have a positive effect on information seeking performance. Information seekers using search strategies simultaneously searching multiple sub-libraries tended to perform more effectively than information seekers who could only perform "local" single-library searches. They also tended to perform more efficiently as indicated by measures such as the time in which the first document is found and the number of states produced during the search process. However, it must be also said that in some cases parallel searching using the uniform collection fusion strategy was no better, and possibly worse, than the single searching condition.
2. The link-based collection fusion strategy tends to perform better than the uniform fusion strategy both in terms of effectiveness and efficiency.
3. Parallel searching is generally useful and increases the performance of information seekers.

Effective distributed metasearch requires good solutions to the collection fusion problems, but the collection fusion problem must be dealt with in any partitioned set of document collections which may be searched in parallel. Although our conclusions focus on the issue of parallel searching in HDLs in general, the implication for meta-searching in the WWW must not be lost.

In summary, we believe that results presented suggest that parallel multi-collection searching may be more effective and efficient than single collection searching, if an appropriate fusion strategy is used for solving the collection fusion problem. On the other hand, we are aware that examining information seeking in large distributed electronic environments is extremely difficult, and further work is required. As computer networks proliferate and more users find themselves searching for information in dynamic, large, heterogeneous, distributed electronic environments, the examination of parallel multi-collection searches will become increasingly more demanding. However, we believe that our experiment and the results which are presented and discussed here are a first step towards this.

References

1. Peter Bailey, Nick Craswell and David Hawking. Engineering a multi-purpose test collection for Web retrieval experiments. *Information Processing & Management*, Volume 39, Issue 6, pp 853-871, November 2003.
2. Bonder R., Chingell M., Charoenkitkarn N., Golovchinsky G. & Kopak R. The impact of text browsing on text retrieval performance. *Information processing & management*, 37, pp. 507-520, 2001.

3. Börner, K. Extracting and Visualizing Semantic Structures in Retrieval Results for Browsing. *Proceedings of the fifth ACM Conference on Digital Libraries*. San Antonio, Tx., USA. 234-5, 2000.
4. Calvé Anne and Jacques Savoy. Database merging strategy based on logistic regression. *Information Processing & Management*, Volume 36, Issue 3, pp 341-359 1, 2000.
5. Gurrin C. & Smeaton A. Replicating Web Structure in Small-Scale Test Collections. *Information retrieval*, vol 7, pp. 239-263, 2004.
6. Deok-Hwan Kim and Chin-Wan Chung. Collection fusion using Bayesian estimation of a linear regression model in image databases on the Web. *Information Processing & Management*, Volume 39, Issue 2, pp 267-285, 2003.
7. Jardine N., and van Rijsbergen C.J. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7, pp. 217-240, 1971.
8. Lai, T.-S., Tait, J.I., and McDonald, S. A User-Centered Evaluation of Visual Search Methods for CBIR. *Proceedings of CIR2000 – Third UK Conference on Image Retrieval*. John P Eakins and Peter G B Enser (eds). University of Brighton, 2000.
9. Marchionini G. *Information Seeking in Electronic Environments*. Cambridge University Press, New York, 1995.
10. Rasolofo, Y., Hawking, D. and Savoy, J., (2003). *Result merging strategies for a current news metasearcher*, *Information Processing & Management*, Volume 39, Issue 4, pp 581-609, 2003.
11. Salampasis M., and Tait, J.I. A Link-Based Collection Fusion Strategy. *Information Processing and Management*, Vol 35(5), pp. 691-711, 1999.
12. Salampasis M. & Diamantaras K. Rich interactions in digital libraries: Short review and an experimental user-centered evaluation of an Open Hypermedia System and a World Wide Web information-seeking environment. *Journal of Digital Information*, May 2002.
13. Voorhees H. Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval. Cornell University, Computer Science Department technical report TR86-765, March 1986.
14. Voorhees E., Gupta N., Johnson-Laird B. Learning Collection Fusion Strategies. *Proc. 18th Annual International ACM SIGIR Conference on Research and Development in Information retrieval*, pp. 172-179, 1995.
15. Wu Mingfang, Michael Fuller and Ross Wilkinson. Using clustering and classification approaches in interactive retrieval. *Information Processing & Management*, Volume 37, Issue 3, pp 459-484, 2001.
16. Zhang J. and Marchionini G. Coupling browse and search in highly interactive user interfaces: a study of the relation browser++. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, Tuscon, AZ, USA June 07 - 11, pp 384 – 384, 2004.

A Native XML Database Supporting Approximate Match Search*

Giuseppe Amato and Franca Debole

ISTI - CNR

Pisa, Italy

{Giuseppe.Amato, Franca.Debole}@isti.cnr.it

Abstract. XML is becoming the standard representation format for metadata. Metadata for multimedia documents, as for instance MPEG-7, require approximate match search functionalities to be supported in addition to exact match search. As an example, consider image search performed by using MPEG-7 visual descriptors. It does not make sense to search for images that are exactly equal to a query image. Rather, images similar to a query image are more likely to be searched. We present the architecture of an XML search engine where special techniques are used to integrate approximate and exact match search functionalities.

1 Introduction

XML is becoming one of the primarily used formats for the representation of heterogeneous information in many and diverse application sectors, such as multimedia digital libraries, public administration, EDI, insurances, etc. This widespread use has posed a significant number of technical requirements to systems used for storage and content-based retrieval of XML data, and many others is posing today. In particular, retrieval of XML data based on content and structure has been widely studied and it has been solved with the definition of query languages such as XPath [3] and XQuery [4] and with the development of systems able to execute queries expressed in these languages. However, many other research issues are still open.

There are many cases where users may have a vague idea of the XML structure, either because it is unknown, or because is too complex, or because many different structures – with similar semantics – are used across the database [5]. In addition there are cases where the content of elements of XML documents cannot be exactly matched against constants expressed in a query, as for instance in case of large text context or low-level feature descriptors, as in MPEG-7 [6] visual or audio descriptors.

In the first case structure search capabilities are needed, while in the second case we need approximate content search (sometime also referred as similarity search).

In this paper we present the architecture of XMLSe a native XML search engine that allows both structure search and approximate content match to be combined with

* This work was partially supported by DELOS NoE [1], funded by the European Commission under FP6 (Sixth Framework Programme) and by the ECD project (Enhanced Content Delivery) [2], funded by the Italian government. We would like to thank Paolo Bolettieri for its contribution to the implementation of XMLSe.

traditional exact match search operations. Our XML database can store and retrieve any valid XML document without need of specifying or defining their schema. Our system stores XML documents natively and uses special indexes for efficient path expression execution, exact content match search, and approximate match search.

The paper is organized as follows. In Section 2, we set the context for our work. In Section 3 we present the overall architecture of the XMLSe system. In Section 4 we describe the query algebra at the basis of the query processor. Section 5 shows some example of query execution in terms of the query algebra, while Section 6 discusses the use of XMLSe in a Digital libraries application. Section 7 concludes.

2 Motivation and Related Work

In the Digital Libraries field three different approaches are typically used to support document retrieval by means of XML encoded metadata. The first consists in using relational database to store and to search metadata. In this case metadata should be converted into relational schemes [7] [8] [9] and this is very difficult when complex and descriptive metadata schemes such as ECHO [10] and MPEG-7 [6] should be managed: even simple XML queries are translated into complex sequences of joins among the relational tables. The second approach consists in using full text search engines [11] to index metadata records, and in general this applications are limited to relatively simple and flat metadata schemes. Besides, it is not possible to search by specifying ranges of values. The third and last approach consists in doing full sequential scan of metadata records. In this case no indexing is performed on the metadata and the custom search algorithms always scans the entire metadata set to retrieve searched information.

A relatively new promising approach is to store metadata in native XML databases as for instance Tamino [12], eXist [13], Xindice [14]. However, these systems, in addition to some simple text search functionality, exclusively support exact match queries. They are not suitable to deal with metadata of multimedia documents, such as color histograms, and to provides users with structure search functionalities.

With the continuous increase of production of multimedia documents in digital format, the problem of retrieving stored documents by content from large archives is becoming more and more difficult. A very important direction toward the support of content-based retrieval is feature based similarity access. Similarity based access means that the user specifies some characteristics of the wanted information, usually by an example image (e.g., find images similar to this given image, represents the query). The system retrieves the most relevant objects with respect to the given characteristics, i.e., the objects most similar to the query. Such approach assumes the ability to measure the distance (with some kind of metric) between the query and the data set images. Another advantage of this approach is that the returned images can be ranked by decreasing order of similarity with the query. The standardization effort carried-out by MPEG-7 [6], intending to provide a normative framework for multimedia content description, has permitted several features for images to be represented as visual descriptors to be encoded in XML.

In our system we have realized the techniques necessary to support XML represented feature similarity search. For instance, in case of an MPEG-7 visual descriptor, the system administrator can associate an approximate match search index to a specific XML element so that it can be efficiently searched by similarity. The XQuery language has been extended with new operators that deal with approximate match and ranking, in order to deal with these new search functionality.

3 System Architecture

In this section we will discuss the architecture of our system, explaining the characteristics of the main components: the data storage and the system indexes. A sketch of the architecture is given in Figure 1.

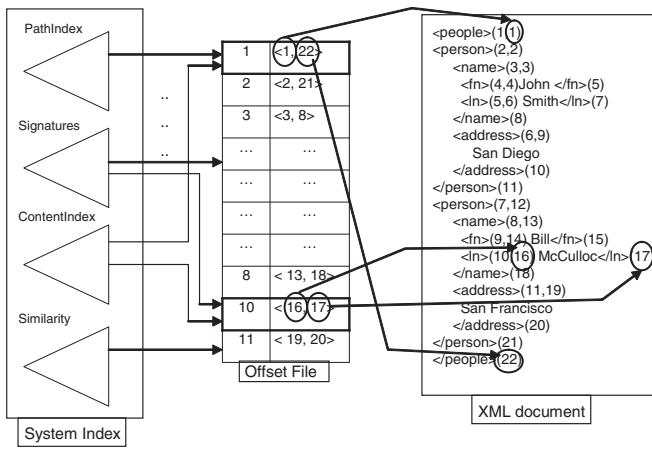


Fig. 1. The components of data storage.

3.1 Data Storage

In recent years various projects [8] have proposed several strategies for storing XML data sets. Some of these have used a commercial database management system to store XML documents [7], others have stored XML documents as ASCII files in the file system, and others have also used an Object storage [15]. We have chosen to store each XML document in its native format and to use special access methods to access XML elements. XML documents are sequentially stored in a file, called *repository*. Every XML element is identified by a unique *Element Instance Identifier* (*eid*). As depicted in Figure 1, we use an offset file to associate every *eid* with a 2-tuple $\langle start, end \rangle$, which contain respectively a reference to the *start* and *end* position of the element in the repository. By using structural containment join techniques [16] containment relationships among elements can be solved. The mapping between XML element names and the corresponding list of *eid* is realized through an element name index.

3.2 System Index

Special indexes are needed to improve the efficiency of XML queries. For this reason we have studied and realized some indexes to efficiently resolve the mapping between element and its occurrences and to process content predicates, similarity predicates, and navigation operations throughout the XML structure.

Path Index. Processing a path expression (es: *//person/ln*), with optional wildcard, involves two steps: first, the occurrences of elements specified in the path expression (es: *person* and *ln*) should be found and second, hierarchical relationships, according to the path expression being processed, should be verified with containment joins. Processing a path expression is much more efficient using ad hoc indexes, like those proposed in [17] [18] [19], which associate entire pathnames with the list of their occurrences in XML documents.

In our system we have proposed a new path index to resolve efficiently the path expressions. The advantage of our approach with respect to the others, is that also path expressions containing wildcards in arbitrary position can be efficiently processed. This approach, discussed in [20], is based on the construction of a *rotated path lexicon*, consisting of all possible rotations of all element names in a path. It is inspired by approaches used in text retrieval systems to processing partially specified query terms. In our system the concept of term is substituted by path: each path is associated with the list of its occurrences and for this reason we call *path lexicon* the set of occurring paths (see Figure 2). Let *path*, *path₁* be pure path expressions, that is path expressions containing just a sequence of element (and attribute) names, with no wildcards, and predicates. We can process with a single index access the following types of path expressions: *path*, *//path*, *path//path₁*, *path//*, and *//path//*. For more details on technique see [20].

Content Index. Processing the queries that, in addition to structural relationships, contains the content predicates (es: */people/person//ln='McCulloch'*), can be inefficient. In order to solve this problem we have extended our path index technique to handle simultaneously the content predicates and structural relationships. The content of an element is seen as a special child of an element so it is included as the last element of a

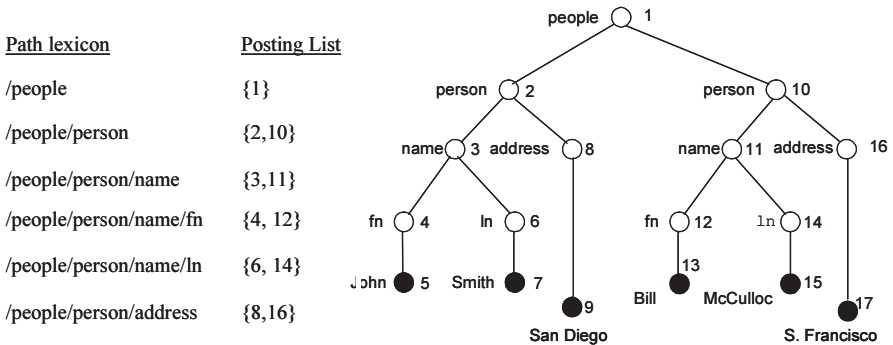


Fig. 2. The paths and their inverted lists associated

path. Of course, it does not make sense to index content of all elements and attributes. The database administrator can decide, tacking into account performance issues, which elements and attributes should have their content indexed. By using this extension, an expression of comparison can simply be processed by a single access to the path index [20].

Tree Signature. Efficient processing of path expressions in XQuery queries requires the efficient execution of navigation operations on trees (ancestor, descendant, parent etc ...), for this reason in our system we have used the *signature file* approach. Signatures are a compact representations of larger structures, which allow the execution of queries on the signatures instead of the documents. We define the tree signature [21] as sequences of tree-node entries to obtain a compact representation of the tree structures. To transform ordered trees into sequences we apply the *preorder* and the *postorder* numbering schema. The *preorder* and *postorder* sequences are ordered lists of all nodes of a given tree T. In a *preorder* sequence a tree node is traversed and assigned its rank before its children are assigned their rank and traversed from left to right, whereas in the *postorder* sequence a tree node is traversed and assigned its rank after its children are assigned their rank and traversed from left to right.

The general structure of tree signature for a document tree T is

$$sig(T) = \langle t_1, post_1, ff_1, fa_1; t_2, post_2, ff_2, fa_2; \dots; t_n, post_n, ff_n, fa_n \rangle$$

where $ff_i(fa_i)$ is the preorder value of the first following (first ancestor) node of the node with the preorder number i . The signature of an XML file is maintained in a corresponding signature file consisting of a list of records. Through this tree signature the most significant axes of XPath can be efficiently evaluated, resolving any navigation operation.

Exploiting the capability of the tree signature is it is also possible to process *structure search queries*, as discussed in [5]. In fact, there are many cases where the user may have a vague idea of the XML structure, either because it is unknown, or because it is too complex. In these cases, what the user may need to search for are the relationships that exist among the specified components. For instance, in an XML encoded bibliography dataset, one may want to search for relationships between two specific persons to discover whether they were co-authors, editors, editor and co-author.

Approximate Match Index. Recently published papers [22] [23] investigate the possibility to search for XML documents not only with the exact-match paradigm but also with the approximate match paradigm. An exact-match approach is restrictive, since it limits the set of relevant and correlated results of queries. With the continuous increase of multimedia document encoded in XML, this problem is even more relevant. In fact rarely a user express exact requests on the features of a multimedia object (e.g., color histogram). Rather, the user will more likely express queries like "Find all the images similar to this".

For supporting the approximate match search in our system, we have introduced a new operator \sim , which can be applied to content of XML elements. To be able to resolve this type of query we have used suitable index structures. With regard to the generic similarity queries the index structure which we use is the AM-tree [24]. It can

be used when a distance function is available to measure the (dis)-similarity among content representations. For instance it can be used to search by similarity MPEG-7 visual descriptors.

For the text search, we make a use of the functionalities of the full-text search engine library. Specifically we have used Lucene [25].

4 Query Algebra

An XQuery query is translated into a sequence of simple operations to be executed (the logical query execution plan). Operators of our query algebra take as arguments, and return, lists of tuples of *eiid* (see Section 3.1). We call these lists *Element Instance Identifier Result (EIIR)*. For instance, given an *EIIR* R , the evaluation of $R_O = \text{Parent}(R, \text{article})$ gives back the *EIIR* R_O that is the set of elements named *article*, which are parents of elements contained in R .

InstanceElements. To initiate processing a query, the first step is finding the occurrences of the element names specified in the query. We define the operator $R_O = \text{instanceElements}(EN)$ that returns R_O , which contain all the *eiid* corresponding to the element name EN . It returns all occurrences of the element name EN in the repository.

Selection. The selection $R_O = \text{select}_P(R_I)$ is applied to R_I to return $R_O \subseteq R_I$ that satisfy a selection predicate P . In addition to the standard set of operators ($=, \leq$, etc.), the elementary conditions supported by XML include the approximate match (or similarity) operator \sim , which is used as a binary operator as $\text{Exp} \sim \text{Const}$. When the elements of Exp are indexed using the AM-tree index the selection operator returns all the elements similar Const , according to the similarity function associated with the AM-Tree. When the Exp is indexed using the full-text index, the selection returns all the elements whose content is pertinent to the text given.

Join. The join operator ($R_O = R_I \bowtie_P R_E$) take as input two *EIIR*, respectively external R_E and internal R_I , and returns the *EIIR* output R_O , which contain the elements of $R_I \times R_E$ that satisfy the predicate P , which is defined on both the *EIIR*.

Navigation Operators. The navigation operators are described in the following. These operators are typically evaluated using the signatures (as described in [26] and resumed in Section 3.2). Several common combinations of these operators can also be processed with the path index (see Section 5 and [21])

- a **child** operator $R_O = \text{child}(R_I)$, which given the *EIIR* R_I returns for every element of R_I its children. For instance the node i has as the first child the node with index $i + 1$ and all the other children nodes are determined recursively until the bound ff_i is reached.
- a **parent** operator $R_O = \text{parent}(R_I)$, which given the *EIIR* R_I returns for every element of R_I its parent. The parent node is directly given by the pointer fa_i in tree signature of every element of R_I .

- a **descendant** operator $R_O = descendant(R_I)$ which given the *EIIR* R_I returns for every element of R_I its descendants. The descendants of node i are the nodes with index $i + 1$ up to nodes with index $ff_i - 1$.
- an **ancestor** operator $R_O = ancestors(R_I)$ which for every element of *EIIR* R_I returns its ancestors. The ancestors nodes is calculated like a just recursive closure of **parent**.

Structure Join. The structure join operator is used to support structure search queries. It is useful when the structure of XML data is unknown and the specific objective of the query is to verify the existence of relationships (in terms of XML hierarchies) among specific elements. Basically this operators, given a tuple of elements, verifies if they have a common ancestor below a specified level, considering that the root of an XML document has level 0. For instance, in Figure 2, nodes *John* and *San Diego* have a common ancestor of level 1. On the other hand, *John* and *S. Francisco* does not have an ancestor of level 1, but they have one of level 0.

The structure join operator $R_O = structureJoin_l(R_1, \dots, R_k)$ takes as input *k EIIR*, and returns the *EIIR* $R_O \subseteq R_1 \times \dots \times R_k$ which have a common ancestors at least a level l in the document structure: all tuples for which there is not a common ancestor of level l are eliminated from the result.

The cost of producing first the Cartesian product of the k lists and then eliminating those tuples that do not satisfy the predicates, can be very high. In [5] we propose a new structure join algorithm, able to perform this step of query execution efficiently.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" ...>
<Description xsi:type="ContentEntityType">
<MultimediaContent xsi:type="ImageType">
<Image>
<MediaLocator>
<MediaUri>D:\ANSAnumb\104.JPG</MediaUri>
</MediaLocator>
<VisualDescriptor xsi:type="ScalableColorType" numOfBitplanesDiscarded="0"
numOfCoeff="64">
<Coeff>-16 34 127 94 5 14 -5 -14 27 15 -11 -28 -11 12 0 1 ... </Coeff>
</VisualDescriptor>
....
<VisualDescriptor xsi:type="EdgeHistogramType">
<BinCounts>2 4 5 6 5 5 1 4 5 4 4 1 2 3 5 3 2 7 7 5 4 3 2 6 5 3 1 4 5 4 4 3 6 6 4 3 1 2 3
</BinCounts>
</VisualDescriptor>
<VisualDescriptor xsi:type="HomogeneousTextureType">
<Average>94</Average><StandardDeviation>144</StandardDeviation><Energy>238
215 186 200 189 209 210 171 179 180 179 170 174 151 122 163 123 151 144 115 98
138 128 141 139 69 53 110 61 71</Energy>
....
</Image>
</MultimediaContent>
</Description></Mpeg7>

```

Fig. 3. An example of a MPEG7 document encoded in XML

5 Query Execution

In this section we discuss the translation of some XQuery queries into our algebra. In the following we assume that the document considered are those of Figure 2 and Figure 3. We resolve the path expression with path index. We suppose that the elements *VisualDescriptor* are indexed by an AM-Tree and *ln* by a full-text index. For each query we give both, the query execution plan in terms of operations of the algebra, and the optimized physical execution plan that conveniently exploits available indexes.

Example 1. Considering the following query:

```
(A)  for $a in /people/person,
      where $a//ln ~ 'Culloc'
      return $a//address
```

We look for the address of person which have *Culloc* in their lastname *ln*. This query is translated in our algebra as follow:

- a) $R_1 = \text{instanceElements}(\textit{people})$
- b) $R_2 = \text{child}(R_1, \textit{person})$
- c) $R_3 = \text{descendant}(R_2, \textit{ln})$
- d) $R_4 = \text{select}(R_3, \sim \textit{'Culloc'})$
- e) $R_5 = \text{ancestor}(R_4, \textit{person})$
- f) $R_6 = \text{descendant}(R_5, \textit{address})$

Whereas using the indexes we have this execution plan:

```
A1  $R_1 = \text{PathIndex}(\textit{/people/person//ln})$ 
A2  $R_2 = \text{FullTextIndex}(\textit{ln}, \textit{'Culloc'})$ 
A3  $R_3 = \text{Intersect}(R_1, R_2)$ 
A4  $R_4 = \text{Ancestor}(R_3, \textit{person})$ 
A5  $R_5 = \text{Descendant}(R_4, \textit{address})$ 
```

We have processed (A1) the path expressions */people/person//ln* with a single access to the index (*PathIndex*), whereas in the logical plan the same expressions is processed with three operations. Second (A2), since full-text index is available on the last element (*ln*) of path, we resolve the select operator with an access to full-text index. Then (A4) the tree signatures are used to navigate through the structure and taken first the *person* ancestor of R_3 and then the *address* descendants of R_4 .

Example 2. Considering the following query related to XML document of Figure 3:

```
(B)  for $a in /Mpeg7, $b in /Mpeg7
      where $a//MediaUri = 'D:\ANSAnumb\104.jpg' and
          $a//VisualDescriptor ~ $b//VisualDescriptor
      return $b
```

It returns all the elements *Mpeg7* whose visual descriptors are similar to that of image ('D:\ANSAnumb\104.jpg'). The logical query plan is:

- a) $R_1 = \text{instanceElements}^{\$a}(\text{Mpeg7})$
- b) $R_2 = \text{instanceElements}^{\$b}(\text{Mpeg7})$
- c) $R_3 = \text{descendant}(R_1, \text{MediaUri})$
- d) $R_4 = \text{select}(R_3, = 'D:\ANSAnumb\104.jpg')$
- e) $R_5 = \text{ancestor}(R_4, \text{Mpeg7})$
- f) $R_6 = \text{descendant}(R_5, \text{VisualDescriptor})$
- g) $R_7 = \text{descendant}(R_2, \text{VisualDescriptor})$
- h) $R_8 = \text{select}(R_7, \sim' R'_6)$
- i) $R_9 = \text{ancestor}(R_8, \text{Mpeg7})$

In the previous plan we use the notation $\text{instanceElements}^x(E)$ to indicate the retrieval of all the *eiid* corresponding to E , and the binding with the variable x . This is an example of a possible physical execution plan:

- B1** $R_1 = \text{PathIndex}(/Mpeg7//MediaUri)$
- B2** $R_2 = \text{Select}(R_1, = 'D:\ANSAnumb\104.jpg')$
- B3** $R_3 = \text{Ancestor}(R_2, \text{Mpeg7})$
- B4** $R_4 = \text{Descendant}(R_3, \text{VisualDescriptor})$
- B5** $R_5 = \text{AM-Tree}(\text{VisualDescriptor}, R_4)$
- B6** $R_6 = \text{Ancestor}(R_5, \text{Mpeg7})$

As in Example 1 we have processed in B1 the path expressions $/Mpeg7//MediaUri$ with an access to index (*PathIndex*). Second (B2), we have selected from the elements of R_1 , the one corresponding to the image *104.jpg*. With the navigation operations (B3, B4) we have accessed the corresponding element *VisualDescriptor*. Then (B5), since image-similarity index is available on the elements (*VisualDescriptor*), we use it to take the elements similar to the selected one R_4 . Finally (B6) the tree signatures are used to navigate through the structure to access the *Mpeg7* ancestor of R_5 .

6 Use of Xmlse for Digital Library Applications

Our XML search engine has been successfully employed to support metadata management in the MILOS [27] [28] multimedia content management system, which in turns has been used for implementing multimedia digital libraries. In Figure 4 we show the search and retrieval interface of the ECHO [10] video digital library application built using Milos. This application allows users to find videos by combining full text, image similarity, and exact/partial match search. Users can browse among scenes of videos, and access corresponding metadata.

In Figure 4 the user searches for German videos related to 'worker strike'. Milos correspondingly generates and submit to XMLSe the following XQuery query:

```
for $a in /echo/AVDocument
where $a/DescriptionLanguage='DE'
and $a/EnglishAbstract ~ 'worker strike'
return $a
```

where the element *EnglishAbstract* is indexed by a full-text index (the exact match radio button is not checked). The user interface, on the left side, shows the results of the

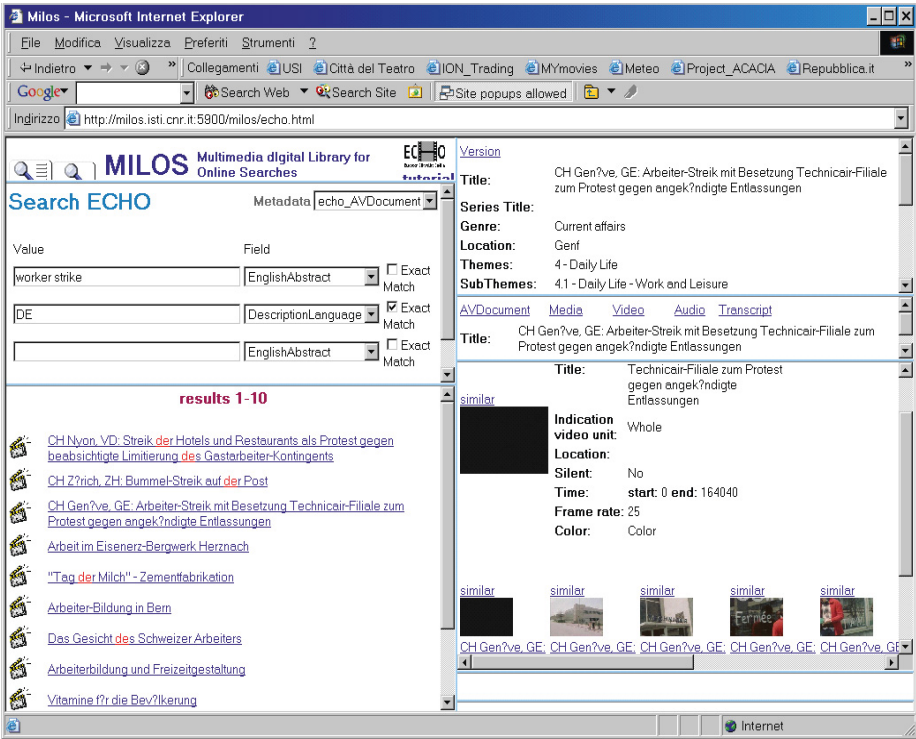


Fig. 4. An example of combined use of exact match and similarity search functionality in MILOS, as supported by XMLSe

query. On the right side, the user can display, for each retrieved document, the related metadata and structural information, which basically consists of the key frames of the scenes contained in the video. The user can select a specific key frame and search for other similar key frames in the repository (see the “similar” link above each key frame in Figure 4).

As a consequence of this, Milos generates and asks XMLSe for processing the following query:

```
for $a in /echo/Video, $b in /Mpeg7, $c in /Mpeg7
where $a/Keyframe = 'urn:milos:echo_video:000000000025653'
and $a/Keyframe = $b//MediaUri
and $b//VisualDescriptor ~ $c//VisualDescriptor
return $c
```

where we suppose that the user is searching for key frames similar to the one identified by 'urn:milos:echo_video:000000000025653', the MPEG-7 *VisualDescriptor* element is used for assessing similarity between key frames, and the *VisualDescriptor* is indexed by an AM-tree (Section 3.2).

7 Conclusion

We have presented the architecture of XMLSe, a native XML search engine that offers XML approximate content search and structure search in addition to traditional exact match search. We have introduced the various index structures that are used to efficiently process XML queries and we have presented the query algebra at the basis of the query processor. This XML search engine is particularly indicated to manage metadata for multimedia digital libraries, where approximate match queries are particularly frequent. The XML search engine has been successfully employed to support metadata management in the MILOS multimedia content management system.

References

1. Delos: (<http://www.delos.info/>)
2. ECD, Enhanced, Content, Delivery: <http://ecd.isti.cnr.it> (2002)
3. XPath1.0: <http://www.w3.org/tr/xpath> (1999)
4. XQuery1.0: <http://www.w3.org/tr/xquery> (2005)
5. Amato, G., Debole, F., Rabitti, F., Savino, P., Zezula, P.: Signature-based approach for efficient relationship search on xml data collections. In: XSym 2004, XML Database Symposium in Conjunction with VLDB 2004. (2004) 82–96
6. MPEG: <http://www.chiariglione.org/mpeg/> (2004)
7. Florescu, D., Kossmann, D.: Storing and querying xml data using an rdbms. In: IEEE Data Engineering Bulletin Vol. 22 No 3. (1999) 27–34
8. Shanmugasundaram, J., Tufte, K., He, G., Zhang, C., DeWitt, D., Naughton, J.: Relational databases for querying xml documents: limitations and opportunities. In: Proceedings of the 25th VLDB Conference, Edinburgh, Scotland (1999)
9. Shimura, T., Yoshikawa, M., Uemura, S.: Storage and retrieval of xml documents using object-relational databases. In: DEXA '99: Proceedings of the 10th International Conference on Database and Expert Systems Applications, Springer-Verlag (1999) 206–217
10. ECHO: <http://pc-erato2.iei.pi.cnr.it/echo/> (2000)
11. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1984)
12. Tamino: <http://www1.softwareag.com/corporate/products/tamino/default.asp> (2001)
13. Meier, W.: exist: An open source native xml database. In: NODe 2002 Web- and Database-Related Workshops, Springer LNCS Series 2593 (2002)
14. Xindice, A.: <http://xml.apache.org/xindice/> (2001)
15. Carey, M.J., DeWitt, D.J., Franklin, M.J., Hall, N.E., McAuliffe, M.L., Naughton, J.F., Schuh, D.T., Solomon, M.H., Tan, C.K., Tsatalos, O.G., White, S.J., Zwilling, M.J.: Shoring up persistent applications. (1994) 383–394
16. Zhang, C., Naughton, J., DeWitt, D., Luo, Q., Lohman, G.: On supporting containment queries in relational database management systems. In: SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data, ACM Press (2001) 425–436
17. Goldman, R., Widom, J.: Dataguides: Enabling query formulation and optimization in semistructured databases. In Jarke, M., Carey, M.J., Dittrich, K.R., Lochovsky, F.H., Loucopoulos, P., Jeusfeld, M.A., eds.: VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece, Morgan Kaufmann (1997) 436–445

18. Chung, C.W., Min, J.K., Shim, K.: Apex: An adaptive path index for xml data. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, June 3-6, 2002, ACM Press (2002)
19. Cooper, B., Sample, N., Franklin, M.J., Hjaltason, G.R., Shadmon, M.: A fast index for semistructured data. In Apers, P.M.G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., Snodgrass, R.T., eds.: VLDB 2001, Proceedings of 27th International Conference on Very Large Data Bases, September 11-14, 2001, Roma, Italy, Morgan Kaufmann (2001) 341–350
20. Amato, G., Debole, F., Zezula, P., Rabitti, F.: Yapi: Yet another path index for xml searching. In: ECDL 2003, 7th European Conference on Research and Advanced Technology for Digital Libraries. (2003)
21. Amato, G., Debole, F., Zezula, P., Rabitti, F.: Tree signatures for xml querying and navigation. In: XSym 2003, XML Database Symposium in Conjunction with VLDB 2003. (2003)
22. Fuhr, N., Großjohann, K.: XIRQL: An extension of XQL for information retrieval (2000) In ACM SIGIR Workshop On XML and Information Retrieval, Athens, Greece.
23. Guha, S., Jagadish, H.V., Koudas, N., Srivastava, D., Yu, T.: Approximate xml joins. In: SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data, ACM Press (2002) 287–298
24. Amato, G., Rabitti, F., Savino, P., Zezula, P.: Region proximity in metric spaces and its use for approximate similarity search. *ACM Trans. Inf. Syst.* **21** (2003) 192–227
25. Lucene: <http://lucene.apache.org/java/docs/index.html> (2000)
26. Zezula, P., Amato, G., Debole, F., Rabitti, F.: Tree Signatures for XML Querying and Navigation. *Lecture Notes in Computer Science* Springer (2003)
27. Milos: <http://milos.isti.cnr.it> (2002)
28. Amato, G., Gennaro, C., Rabitti, F., Savino, P.: Milos: A multimedia content management system for digital library applications. In: European Conference on Digital Libraries, ECDL 2004, Bath, UK, September 12-17 2004. (2004) <http://milos.isti.cnr.it/>.

XMLibrary Search: An XML Search Engine Oriented to Digital Libraries*

Enrique Sánchez-Villamil, Carlos González Muñoz, and Rafael C. Carrasco

Transducens , Departamento de Lenguajes i Sistemas Informáticos,
Universidad de Alicante, E-03071, Alicante

Abstract. The increase in the amount of data available in digital libraries calls for the development of search engines that allow the users to find quickly and effectively what they are looking for. The XML tagging makes possible the addition of structural information in digitized content. These metadata offer new opportunities to a wide variety of new services. This paper describes the requirements that a search engine inside a digital library should fulfill and it also presents a specific XML search engine architecture. This architecture is designed to index a large amount of text with structural tagging and to be web-available. The architecture has been developed and successfully tested at the Miguel de Cervantes Digital Library.

Keywords: Passage Information Retrieval Systems, XML Search Engines, Digital Libraries.

1 Introduction

During the last years, the amount of digital information available has grown quickly. Then, it is compulsory the development of tools that would allow a user to get easily and quickly the needed information. A digital library owns a large work collection that is made available to readers. However, the user does not often enter the library to *read* a work but to *look up* in it. In classical libraries only browsing or catalogue searching are possible.

Once the information is digitized, a wide range of exploitation possibilities is available. The next stage in accesibility is the development of search engines, that allow to find information inside the works (content and structure), to free the user from the task of searching manually the required information.

The next step is to take advantage of the structure of the digitized works. When digitizing works, it is interesting to store not only the text, but also some structural description. For this purpose, XML tagging is used. The arborescent structure of the XML documents allows higher level searches.

At present, there is a wide variety of XML searching tools and some of them are distributed as open source, such as `Fxgrep` [1], `Xset` [2], `Sgrep` [3], `XQEngine`

* Work partially funded by the Spanish Government through grant TIC2003-08681-C02-01.

[4], Lucene [5], TSep [6] or eXist [7]. However, in general, they are document information retrieval systems (TeraXML [8], XIndex [9], DataparkSearch [10]) rather than passage information retrieval systems [11] like XMLibrary Search. Besides, we are not aware of any non-commercial search engine that generates links to the document passage in which the occurrences are located; they generate links to the top of the document instead. There are some search engines that use morphological information to maximize their recall (Convera RetrievalWare [12]), which obtain interesting results for research purposes. Our engine offers some query suggestions¹ to the user instead, which allow them to choose the level of coverage. The use of index based search engines, which in general are faster than database based search engines, is appropriate since digital libraries do not change very often the content of their XML documents.

Search engines can be used as a basic tool for a wide range of services that could be useful for the users of digital libraries, such as summary generation applications, document analysis reporting tools or the development of a collection of learning objects for language and literature courses.

Some of the benefits of the architecture proposed in this paper are:

- The handling of a large text collection.
- The processing of a relatively complex set of queries.
- The speed to solve the queries.

The XML search engine that is presented in this paper is integrated in the Miguel de Cervantes Digital Library.² The paper is divided into the proposed distributed system architecture and the internals of the search engine.

2 Requirements

During the design of a search engine, some incompatible goals or features have to be compromised to achieve the needed requirements. For instance, a digital library needs fast searches while in an information extraction system the speed is not the priority although they usually deal with complex queries.

The desirable features could be classified into two groups: functionality features (introducing constraints to the set of documents, sorting the results by their relevance, showing topics related with the query...) and system features (as speed, accuracy, robustness, safety...). Croft [13] proposed a different classification that emphasized the integration of information retrieval systems (search engines) with other systems and also stressed the distributed architectures. A different classification according to digital libraries needs, follows.

- Efficacy: As Croft stressed, a good efficacy level can only be achieved by a properly designed user interface. This interface should be both powerful enough to allow complex queries and easy enough to be used by unexperienced users.

¹ Based on morphological information, synonyms and word similarity.

² <http://www.cervantesvirtual.com/herramientas/textos/buscador.i.shtml>

- Speed: The user should be able to see the results very quickly even if many users are using simultaneously the system. The more complex the queries are, the slower their solving will be. Furthermore, the speed should not decrease significantly when increasing the size of the text collections.
- Information Retrieval: The system should not only redirect the user to the information but also preview what the user is looking for. If the user is looking for a sentence, it will not be enough to show in which work the sentence is found. The system should also show the context of the phrase and also redirect to the exact point of the work where the phrase was found.
- Safety: The search engine uses some private information that has to be preserved. Safety is directly related to information retrieval because any search engine needs full access to the works.
- Multimedia: The multimedia digital libraries have extra requirements, due to the fact that the handled data is much larger in size and, therefore, processing time is longer. The search engine presented in this paper does not handle this kind of data. However the proposed architecture would easily allow the integration of this feature.
- Query expansion: The query expansion consists in the generation of variations of the user's query. That variations can be generated after the search, as a suggestion of related queries, or before the search to offer higher quality coverage results. These expansions are usually based on thesauri and natural language processing heuristics.

Any architecture for a search engine should take into account all of these topics, so that its quality along with its functionality could be properly measured.

3 Search Engine Architecture

The architecture presented in this paper is a distributed system with three different layers, as can be seen in figure 1. These three layers are fully independent and communicate each other using TCP/IP protocols. To maximize the performance it is recommendable to execute each layer in different computers. The performance of the system is given by the maximum number of searches that the system can process per second.

Users access the search engine through the web server, which executes the search engine client to send queries to the user server. The user server is the responsible of query distribution to the available query servers. Once the query server has finished attending to a query the user server sends its result to the corresponding client. All query servers run independently, so that they need a local copy of the search index.

This architecture offers a very good performance, and if the average load of the system is small then all servers can run on the same computer. The different layers are explained in detail below.

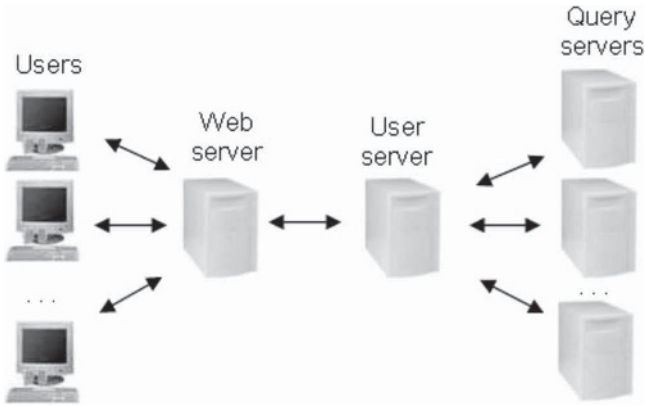


Fig. 1. Distributed architecture of the search engine

Form for advanced search in texts

Search

The WORDS in

The WORDS in
All

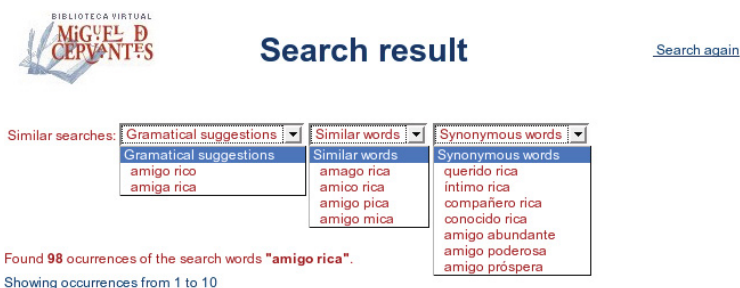
The WORDS in different from the work's one
All the words

Limit to	Options
TITLE of the work <input type="text"/>	<input type="checkbox"/> Search exact phrase
AUTHOR of the work <input type="text"/>	<input checked="" type="checkbox"/> Search orderly words
<input type="checkbox"/> written	<input type="checkbox"/> Search disorderly words
PERIOD the work was <input type="text"/> between <input type="text" value="Before 14th c."/> and <input type="text" value="21st century"/>	<input type="checkbox"/> Search inside editor's notes
<input type="checkbox"/> published	<input type="checkbox"/> Case sensitive
	<input checked="" type="checkbox"/> Arrange results

Show

Context for each coincidence Results per page linguistic expansions

Fig. 2. User interface of the XML search engine running at the Miguel de Cervantes Digital Library



The screenshot shows the search results page for the query "amigo rica". At the top left is the logo for "BIBLIOTECA VIRTUAL MIGUEL DE CERVANTES". The main heading is "Search result" with a "Search again" link to the right. Below the heading, there are three dropdown menus for "Similar searches": "Gramatical suggestions" (showing "amigo rico" and "amiga rica"), "Similar words" (showing "amico rica", "amigo pica", and "amigo mica"), and "Synonymous words" (showing "querido rica", "intimo rica", "compañero rica", "conocido rica", "amigo abundante", "amigo poderosa", and "amigo próspera"). Below these menus, it states "Found 98 occurrences of the search words 'amigo rica'." and "Showing occurrences from 1 to 10".

1 **TITLE:** Ángel Guerra **AUTHOR:** Pérez Galdós, Benito

[. . .] -Pero no ha llegado todavía el momento de dejar libre y horro a nuestro grande **amigo** y consejero -agregó la **rica**-hembra-, y he venido a suplicarle que se pase por allá y eche unos exorcismos a la niña, porque desde anoche se me ha puesto muy triste [. . .]

2 **TITLE:** Eugenia Grandet **AUTHOR:** Honoré de Balzac

[. . .] en el bolsillo de su chaleco y que tentaba de vez en cuando, mandaba que le trasladasen a su sitio ordinario y permanecia allí silencioso. Por lo demás, su antiguo **amigo** el notario, comprendiendo que la **rica** heredera se casaría necesariamente con su sobrino el presidente, si Carlos Grandet no volvía, redobló sus cuidados y sus [. . .]

3 **TITLE:** Ensayos de Montaigne seguidos de todas sus cartas conocidas hasta el día **AUTHOR:** Michel de Montaigne

[. . .] al tirano o a sus cómplices, sin emplear las formalidades de la justicia; juzgaba perverso a un hombre, por ordinario ciudadano que fuera, si en la batalla no era humano con su **amigo** y con su huésped. Alma de **rica** composición, casaba con las acciones humanas más rudas y violentas la humanidad y la bondad, hasta las más exquisitas que [. . .]

Fig. 3. A sample query result when searching the words "amigo rica"

3.1 Search Engine Client

The search engine client is the volatile part of the architecture, provided that each client processes a single query. The functionality of the client is restricted to the designed interface,³ which is shown in figure 2. It builds a data packet containing the query and information related with the user and sends it to the user server. Additionally, the client can provide an adaptation between different encodings.

The query syntax is specified by a grammar designed to allow only possible queries that can be resolved quickly with the index.⁴ This protocol optimizes the querying of the index to speed up the solving process. The query does not specify only what to look for but also how to search in the index.

The data packet is sent to the user server and, after that, the client waits for the result. The result consists of a webpage fragment, so that the client adds the header and the footer, which allows the rest of the search engine to be completely independent of the webpage design. Once the result has been received, the client

³ Each digital library should build its own interface.

⁴ Depending on the architecture some queries are easy to process and others are not (see Baeza-Yates [14]).

builds the webpage to show it to the user and finishes its execution. In figure 3 a sample query result is shown.

3.2 User Server

The user server coordinates the whole system. It receives requests from all clients and distributes them between the available query servers.⁵ After receiving the results from the query servers it reroutes them to the corresponding clients. The user server can directly resolve some of the requests because it manages a query cache.

The system is coordinated by a heterogeneous database, which contains data of several kinds:

- Query servers: Availability, load, remaining query queue, idle time, resolved requests.
- Users: IP addresses, queries requests, waiting time.
- Queries: Users that requested it, query server in charge of solving it, status, number of occurrences found, time spent in solving it, result of the query.
- Statistics: Working time, number of queries, average time per query, total size of the query results in bytes, etc.

The user server, acting as a cache, keeps the data of the queries for some time, storing temporarily their result, so that another client posting the same query would be directly answered with the result. Furthermore, these data allow queries that have been enqueued for a long time to be discarded (only in case of an overloaded system).

User data is not only used for the redistribution of the result of the queries, but also as a password controlled access. Moreover, the user server distinguishes between several privilege levels, so that some queries may be restricted to authorized users.

The user server generates a log with all requests served, discarded or denied, and any occurred incidence. Furthermore, it manages an on-line statistics generation system, which allows administrators to follow the execution of the whole search engine.

The search engine is controlled only by a single user server that centralizes the generation of statistics and log modules. This centralization allows for a high performance and easeness of maintenance of the whole system. However, the only limit of the performance is the number of requests that the user server can process. In our experiments, only 0.06% of the time needed to solve a query was spent in the user server. Therefore, in practice, the performance will be limited by the number of query servers available.

3.3 Query Server

The query server attends to the query requests received from the user server. The queries are queued and solved individually. After solving a query, its result

⁵ Except for some requests that can be answered directly by the user server.

is sent back to the user server, along with some statistics such as the time spent in solving it.

The process of attending to the queries is divided into two different phases. In the first one, the query is processed to generate a list of occurrences. In the second one, the XML files are accessed according to the list of occurrences to extract their context⁶ and to obtain the result of the query.

A certain range of occurrences to show is always specified, i.e., the occurrences from 1 to 50, so that the query server needs to access a limited number of files without slowing down the process. The search engine implements a passage information retrieval system [11]. Given the structure of the XML documents, the content shown is the same passage⁷ in which the occurrence was found.

The division between these two phases allows the query server to store the results of the first phase into a cache, so that when the same query with different range of occurrences is requested, only the second phase has to be executed again. The second phase takes a significant amount of time, due to the fact that the XML files are stored in the hard disk. After retrieving the contexts, the query expansion generation, which will be explained in section 4.3, is executed.

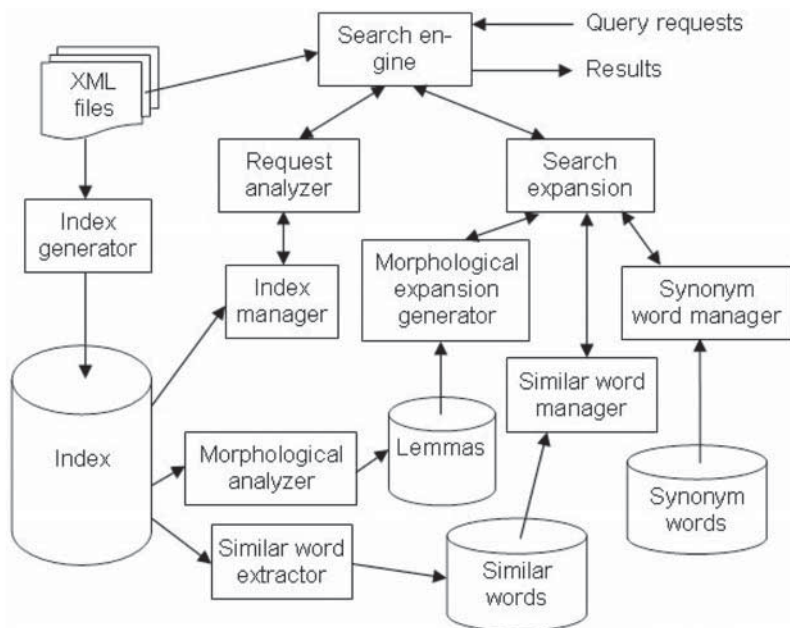


Fig. 4. Search engine internals scheme with all its modules as well as the databases

⁶ The context is retrieved as plain text to preserve the XML format.

⁷ In our case, the search engine considers some TEI elements (paragraphs, verse lines and citations) as passages (see <http://www.tei-c.org>).

4 Search Engine Components

The search engine system, which is integrated in the query server, is the responsible of attending to query requests. Its performance relies in a previously generated index obtained from the collection of XML files. This index provides direct access to the words, tags and attributes included in the collection. This kind of index ensures that query solving time grows atmost linearly with the size of the collection of XML files.

Figure 4 shows the structure of modules that compose the internals of the search engine system. The search engine analyzes each query request to create an execution plan. Once the plan is created, queries are processed and expansions are also analyzed to suggest them to the user. The following paragraphs describe in detail the different modules that compose the engine.

4.1 Index Generator and Index Manager

The index generator module is the responsible of indexing the information from the XML file collection, which is related to the exact position of each item type (word, tag or attribute) within the collection. The generation process takes about 15 minutes to finish using a Pentium IV-1.5GHz machine for an XML files collection of 300 megabytes.

The data structure of the index, contains a hash table for each of the different item types, as well as one index file for all of them. This index file contains for each item all the positions within the documents where they appear. Each one of the hash tables will contain a different entry for each item, so that, when one of these items is searched, it is possible to obtain both the exact number of occurrences of the item and the position within the index file where the related information is located.

The size of the set of index files is comparable to the size of the whole XML file collections; no compression algorithm is used. Due to this, it is not feasible to maintain the whole index in memory. This design allows to maintain in memory only the hash tables. Therefore, to access the index file at most a few disk reads (of contiguous pages) are needed.

The index manager module exposes a simple interface to search words, tags or attributes, offering also mechanisms to perform nested searches exploiting the XML arborescent structure.

4.2 Request Analyzer

This module is the responsible for studying the query requests sent by the client. It requires access to the index and the XML files to generate the results that satisfy the query. To generate the results, the collection of XML files has to be accessed, and this slows response time.

Digital libraries may have their works available on-line. In such case, it is possible to provide a link to the exact position in the work for each of the query results, so that direct access to the required information is enabled.

4.3 Search Expansion

The task of this module is to suggest further queries similar to the posted ones. This search expansion module relies in three modules: the morphological expansion generator, similar word manager and synonym word manager.

The morphological expansion generator provides access to the morphological analysis of the words in the XML documents. The generation of these analysis is performed right after the index generation process, so that delays are avoided when solving queries. The morphological analysis module that the system uses is part of the Spanish-Catalan machine translation system *interNOSTRUM* [15].

The similar word manager returns a list of the similar words, that is, words that are ortographically close. Information about all the similar words among the words found in the XML documents is also automatically generated right after the index generation phase. This task improves user experience as it provides semiautomatic correction of typographic errors.

The synonym word manager obtains the synonyms of the most common words contained in the XML documents. These synonyms are stored in a hand-crafted database, which is accessed during the query solving phase.

The search expansion works differently depending on the number of words contained in the query. When the user is searching a single word, similar and synonym words are suggested, whereas when more words are contained in the query, these are analyzed to check their gender, number and case (if it is a verb), and the suggested expansions keep morphological concordances.

5 Experiments

It is not easy to automatically evaluate the grade of accomplishment of some of the requirements proposed in section 2, such as efficacy or safety. However, we can analyze if the system meets or not requirements such as information retrieval, query expansion and multimedia features. The speed of the system, which is one of the most important requirements, can be measured in terms of queries solved per second.

In our experiments, we have tried to estimate the speed of the system by emulating a real case. We have built a test set of 10,000 queries, containing the following types of queries:

- 2,500 single word cached queries with large result sets.
- 2,500 complex cached queries with empty result sets, that are not cached.
- 2,500 complex cached queries with small result sets.
- 2,500 single word uncached queries with large result sets.

The experiments were performed running all the servers in the same computer and the system was able to answer approximately 5.5 queries per second using an XML file collection of 300 megabytes in a Pentium IV 1.5GHz with 1GByte of memory.

6 Conclusions

The global analysis of the architecture reveals that the required objectives are achieved. The architecture offers a high efficiency, given that the user queries are always processed and suggestions are made. However, the efficacy is very dependant on the web interface given to the application. So, if the users can not express precisely what they want, the efficacy is decreased.

The context of the occurrences of the searched words is obtained using a system based on passages, thus satisfying the information retrieval prerequisite. Furthermore, unlike other XML search engines, direct links are generated to the exact positions within the published documents, which allows the user to access directly to all the information related with the query.

Execution time is very dependant on the implementation, but in theory, it is very low, as new query servers can be easily added to the system. The process time needed by the user server is minimal. Therefore, it will be seldom overloaded regardless the number of users accessing at it. Network speed is essential as it determines the maximum number of requests per second that can be received; if the user server is able to listen to all of them, the global performance will never be limited by the user server.

Safety of the information is quite high, because queries pass through the web server, the user server, and eventually, also through the query server. The user server contains a request register and can hence restrict the access to the resources. Additionally, the query server accesses the XML collection and returns only plain text to preserve the original XML files.

Finally, the search expansion, based on grammatical criteria, words similarity and synonymies, improves the user interaction with the system. Besides, the proposed architecture is ready to easily integrate new services needed in the future, i.e., working with multimedia data that will become usual in the next years.

7 Future Work

At present, we are working on an open source version of the search engine. This new version will be easy to install and integrate into any digital library. The aim is to allow webserver administrators to successfully install `XMLibrary search`, but still allowing them to customize it to fit their needs. Furthermore, the open source version would also allow the indexation of HTML files, after being turned into XHTML files.

In addition, we are researching index compression algorithms that would allow the search engine to be faster, provided that smaller indexes are read faster from the hard disk and decompression time is minimal. Moreover, we are studying the new services that a suffix array based index [16] would allow.

Eventually, the integration of several tools of natural language processing like part-of-speech taggers and machine translation tools would improve the system.

References

1. Neumann, A., Berlea, A., Seidl, H.: Fxgrep: A XML querying tool. In: <http://www.informatik.uni-trier.de/~aberlea/Fxgrep/>. (2000)
2. Zhao, B.Y., Joseph, A.: Xset: A lightweight XML search engine for internet applications. <http://www.cs.berkeley.edu/~ravenben/xset/html/xset-saint.pdf> (2000)
3. Jaakkola, J., Kipeläinen, P.: Using sgrep for querying structured text files. <http://www.cs.helsinki.fi/TR/C-1996/83/> (1996)
4. Katz, H.: XQEngine - XML query engine. <http://xengine.sourceforge.net/> (2003)
5. Goetz, B.: The Lucene search engine: Powerful, flexible and free. <http://www.javaworld.com/javaworld/jw-09-2000/jw-0915-lucene.html> (2000)
6. Noehring, O., Jedlicka, M.: TSep: The search engine project. <http://tsep.sourceforge.net/> (2004)
7. Meier, W.: eXist: An Open Source Native XML Database. In: Web, Web-Services, and Database Systems. (2002) 169–183
8. Doclinx: TeraXML enterprise search. <http://www.doclinx.com/products/ftxml.html> (2002)
9. Liota, M.: Apache's XIndice organizes XML data without schema. <http://www.devx.com/xml/article/9796> (2002)
10. Zakharov, M.: DataparkSearch engine. <http://www.dataparksearch.org/> (2004)
11. Salton, G., Allan, J., Buckley, C.: Approaches to Passage Retrieval in Full Text Information Systems. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1993) 49–58
12. Convera: Convera Retrievalware. <http://www.convera.com/> (2004)
13. Croft, W.: What do people want from information retrieval? <http://www.dlib.org/dlib/november95/11croft.html> **D-Lib Magazine volume 1** (1995)
14. Baeza-Yates: Proximal nodes: a model to query document databases by content and structure. *ACM Transactions on Information Systems (TOIS)* **Volume 15, Issue 4** (1997) 400–435
15. Canals-Marote, R., Esteve-Guillén, A., Garrido, A., Guardiola-Savall, M., Iturraspe-Bellver, A., Montserrat-Buendia, S., Ortiz-Rojas, S., Pastor-Pina, H., Pérez-Antón, P., Forcada, M.: The Spanish-Catalan machine translation system interNOSTRUM. 0922-6567 - *Machine Translation VIII* (2001) 73–76
16. Manber, U., Myers, G.: Suffix arrays: A new method for on-line string searching. In the first Annual ACM-SIAM Symposium on Discrete Algorithms (1990) 319–327

From Legacy Documents to XML: A Conversion Framework

Jean-Pierre Chanod, Boris Chidlovskii, Hervé Dejean, Olivier Fambon,
Jérôme Fuselier, Thierry Jacquin, and Jean-Luc Meunier

Xerox Research Centre Europe
6, chemin de Maupertuis, F-38240 Meylan, France
{firstname.lastname}@xrce.xerox.com

Abstract. We present an integrated framework for the document conversion from legacy formats to XML format. We describe the *LegDoC* project, aimed at automating the conversion of layout annotations layout-oriented formats like PDF, PS and HTML to semantic-oriented annotations. A toolkit of different components covers complementary techniques the logical document analysis and semantic annotations with the methods of machine learning. We use a real case conversion project as a driving example to exemplify different techniques implemented in the project.

1 Introduction

The eXtended Markup Language (XML) is the modern industry standard for data exchange across service and enterprise boundaries. It became a common practice to use XML as the underlying data model for the information capture, exchange and reuse. A large spectrum of activities led by the Web Consortium, OASIS and other actors around XML, including XML schema, querying and transformation, has led to an increasing availability and exchange of data and documents in XML format, the proliferation of user-defined XML schema definitions (DTDs, XML Schemas, Relax NG), the integration of XML components in large-scale content management solutions.

The migration from legacy formats to XML has two important branches, *data-oriented XML* and *document-oriented XML*. Data-oriented XML refers to well-structured data and storage systems like databases or transactional systems. The migration of this data toward XML poses no serious problems, as data is already well-structured and ready for machine-oriented processing. Instead, the migration of *legacy documents* toward XML raises important issues. Documents that often form corporate and personal knowledge bases, are unstructured or semi-structured objects; they are stored in generic or specialized file systems, in a multitude of formats and forms. A large majority of documents are created for humans and not machines, with various implicit assumptions and choices which are obvious for a human reader, but difficult and ambiguous for computer programs. The migration of legacy documents toward XML addresses the document transformation into a form that eases the machine-oriented processing and reuse of documents, through a process that makes all implicit assumptions and choices explicit, and is guided by a common sense or by a specific domain knowledge.

In the *mass document migration* toward XML, source documents are available in rendering-oriented formats like Adobe PDF, PostScript or Microsoft Word. The migration result is expected to be XML documents that fit a user-defined or domain-specific XML schema. It is frequent in the conversion that the target documents preserve an important part of the source content but disregard all information relevant to the document presentation, such as pagination, headings, etc. Structurally, source and target documents are often very different, as they follow two opposite paradigms of *layout-oriented* and *semantic-oriented* document annotations. The former refers to the traditional, human-oriented paradigm of document annotation; the later is associated with a relatively new paradigm of the semantic-oriented annotation of documents for the machine processing.

Currently, the conversion of legacy documents into dense semantic XML is performed by domain experts and remains essentially manual and expensive. XML and Web communities offer various tools for transforming data into and from XML, including XSLT, XQuery and their graphical extensions [17,18]. However, writing accurate transformation rules for the mass document conversion appears difficult if even possible, because of the size and complexity of both source documents and target schema.

The current state-of-art in the domain of semantic annotation leaves not much hope for achieving the fully automated and accurate converters in any observable future. Nevertheless, the conversion cost can be considerably reduced by deploying different and complementary methods. One well-established approach is the analysis of the logical structure of documents. Another approach can be based on data mining or machine learning techniques that can attempt to infer accurate transformation rules using a subset of annotated source documents or their fragments.

We adopt an approach of managing the conversion complexity by decomposing the entire problem into a sequence of smaller and easier-to-handle conversion or transformation steps, where each sub-problem can be solved with an appropriate method. Any step performs a specific processing of the document, enriches it and thus puts it closer to the target XML format.

2 Legacy Document Conversion Project

At Xerox Research Centre Europe, we are conducting the Legacy Document Conversion (*LegDoC*) project aimed at automating different tasks of the mass document conversion to XML. The typical conversion task starts with a large collection of legacy documents available in PDF, PostScript or Microsoft Word formats. The schema of target documents are provided in the form of DTD or W3C XML Schema descriptions. The conversion goal is to migrate source documents or their components into the XML files structured according to the target schema.

The generic view of the conversion flow is presented in Figure 1. According to this figure, we distinguish among three types of document annotations. The first type refers to **layout** annotations that cope with the document presentation in terms of the physical rendering of elements (x and y positions, width, height, font, etc.). The second type refers to a more abstract, **logical** structure of the document, it expresses spatial relationships between elements in a page, such as columns, headings, paragraphs and lines. The

third type of annotations is **semantic** one; it refers rather to the meaning of elements than to their appearance on a page. Semantic annotations may be of different granularity with two well-known examples being the metadata and entities. *Metadata* refers to elements that describe the whole document, like title, authors, creationDate, etc.; all such elements are routinely indexed and used in content management applications. *Entities* are content elements of a low granularity, like person names, tool names, index entry points, etc.

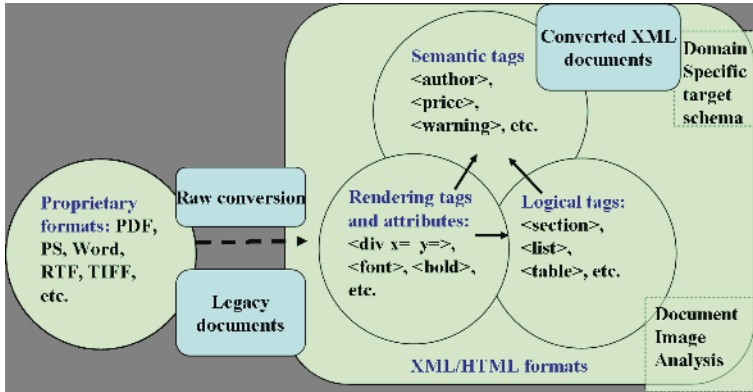


Fig. 1. Three types of annotations and the conversion flow

To achieve the target of converting legacy documents to XML, the *LegDoC* project offers a framework for modeling, evaluating and executing various conversion cases. The project framework is composed of the following components:

Raw XML: the conversion starts with rewriting documents from proprietary formats into a raw XML. This deploys off-the-shelf converters for Adobe PDF and other formats. The output of a converter is XML files that preserve the rendering of documents. All converters allow an accurate recognition of characters, lines and their rendering attributes (x and y positions, fonts, etc.). However, they are fairly limited in the recognition of logical or semantic annotations.

Preprocessing: this component cleans up and indexes the raw XML files. The index entry points get associated with all XML nodes and remain persistent during all the conversion process thus enabling the easy traceability and debugging of the different conversion steps.

Logical analysis: it includes methods for the spacial analysis and extraction from the raw XML, the detection of headers and footers, determination of the reading order, the document structuring using the Table of Content where available, etc.

Semantic annotation: it covers methods for recognizing entities in the document content. The methods are both hand-crafted regular expressions and a collection of machine learning algorithms that allow one to build learning models from corpora of annotated documents and apply the models to non-annotated documents.

Visualization and annotation: it includes an assistance for the visualization and validation of outputs of intermediate and final conversion steps.

Conversion management: this component offers a support for building chains of transformation and enrichment steps, gradually migrating from the raw XML to the target XML. It represents an explicit set of agreements and requirements for a transformation chain and validity of output of intermediate steps, including XML schema definitions for any step.

The core components of the logical analysis, semantic annotations, visualization and annotations are presented in more detail in Sections 4-6.

3 Conversion Example

The components in the *LegDoC* framework are developed in a generic manner, they however require an adaptation to any specific conversion task. In the following, we use one case of technical documentation conversion as a driving example for presenting how the components contribute to the conversion process. The selected case is given by a collection of *truck repairing manuals*. The target schema is a complex W3C XML Schema description which scrupulously describes all notions and entities relevant to the truck repairing world, including tools, operations, steps and items of demounting and mounting processes, etc. The gross volume of PDF documents is dedicated to the repairing operations, one such page is shown in Figure 2.a. Figure 2.b gives the SVG

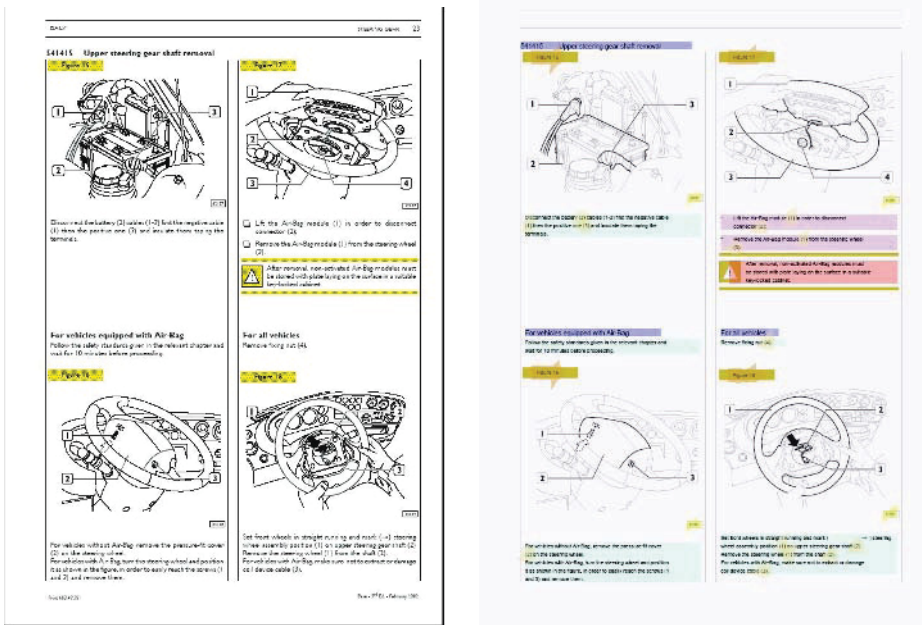


Fig. 2. Conversion example: left) Source PDF file, right) Target XML with annotations

<pre> <TEXT x="340.0" y="76.0 height="9.0" width="38.0" font="Times">Figure 17</TEXT> <TEXT x="544.0" y="269.0" ...>62108</TEXT> <TEXT x="324.0" y="291.0" ...></TEXT> <TEXT x="341.0" y="291.0" ...>Lift the Air-Bag module (1) in order to disconnect</TEXT> <TEXT x="341.0" y="302.0" ...>connector (2);</TEXT> <TEXT x="324.0" y="318.0" ...></TEXT> <TEXT x="341.0" y="319.0" ...>Remove the Air-Bag module (1) from the steering wheel</TEXT> <TEXT x="341.0" y="330.0" ...>(3).</TEXT> <TEXT x="357.0" y="352.0" ...>After removal, non-activated Air-Bag modules must</TEXT> <TEXT x="357.0" y="363.0" ...>be stored with plate laying on the surface in a suitable</TEXT> <TEXT x="335.0" y="362.0" ...>!</TEXT> <TEXT x="357.0" y="374.0" ...>key-locked cabinet.</TEXT> </pre> <p style="text-align: center;">(a)</p>	<pre> <Figure-Number>17</Figure-Number> <Figure-Ref>62108</Figure-Ref> <List> <Item>Lift the Air-Bag module (1) in order to disconnect connector (2);</Item> <Item>Remove the Air-Bag module (1) from the steering wheel (3).</Item> </List> <Warning>After removal, non-activated Air-Bag modules must be stored with plate laying on the surface in a suitable key-locked cabinet.</Warning> </pre> <p style="text-align: center;">(b)</p>
--	--

Fig. 3. Conversion example: a) fragment of the raw XML, b) fragment of the target XML

representation of the same page with all target XML annotations highlighted with different colors.

Figure 3 offers a closer look on the problem. Figure 3.a shows a fragment of raw XML corresponding to the upper part of the left column in Figure 2.a. The fragment shows layout annotations, after the reading order detection (see Section 4). Figure 3.b shows the target annotation of the selected fragment. The differences between the two fragments exemplify the main issue of the conversion task: *the layout annotations with line-level granularity get converted into logical, paragraph-level annotations and semantic annotations of entities, like Figure-Ref, List, Warning, etc.*

4 Logical Structure Analysis

The conversion chain starts with converting the source PDF documents into raw XML format. After the raw XML is cleaned up and indexed, we call for the logical analysis component. Its purpose is to logically organize the document. By logical structures, we mean generic structures occurring frequently in documents, such as columns, headings, paragraphs and lists. The first step of the component refers to rebuilding appropriately the text flow of the document, which corresponds to finding its reading order. The second step logically structures the document according to the Table of Contents.

4.1 Rebuilding the Reading Order

The reading ordering problem consists in inducing such an order between layout objects on a page that reflects how the humans used to read them. The problem has been already addressed in the page image analysis [4,10,14]. However, the use of off-the-shelf PDF converters adds a new spin to the problem, as it requires considering objects of different granularity, such as lines, words, syllables, even letters. Since the layout annotations in the raw XML allow multiple valid orderings, the correct order can be detected by exploiting other features, like the font values or the textual content. In our case, we adopt a geometry-based approach which proved its efficiency on the considered class of documents.

In its origin, the XY-cut is a page segmentation method [9] working on a page image. The method consists in finding the widest empty rectangle, or *valley*, entirely crossing the page image either vertically or horizontally. The page is then segmented in two blocks, which are shrunk to fit closely their content (a threshold gives some robustness to image noise). The method applies then recursively to each block. It stops when no large-enough valley can be found in any of the created blocks, which become the final image segments. [8] implemented the XY-cut by using the bounding boxes of connected components of black pixels instead of using image pixels. Once the connected components are obtained, the recursive XY-cut becomes much faster to compute.

Ishitani [10] proposed to leverage the hierarchy of blocks generated by the XY-cut algorithm to induce an order among blocks, in particular among leaf blocks. For a western conventional top-bottom left-to-right reading order, ordering the hierarchy of blocks in the same, top-bottom left-to-right way leads often to the expected order.

A local order can be induced inside blocks by looking for horizontal cuts with a lower threshold, that allows for thin valleys or even some overlapping. This is a way of forming lines in an elegant manner, with the same cutting model. Finally, the block ordering for all pages defines the complete document ordering.

Since the sequence in which blocks operationally appear determines the block order, we need to control the formation of the hierarchy of blocks. Following the spirit of Ishitani's suggestions [10], the chosen cutting strategy and the score function favor a reading by columns, since this is the conventional reading in most technical documents. But rather than relying on a bottom-up approach based on local geometric features, our method consists in selecting among all possible cuttings of a block the one that leads to the best set of columns.

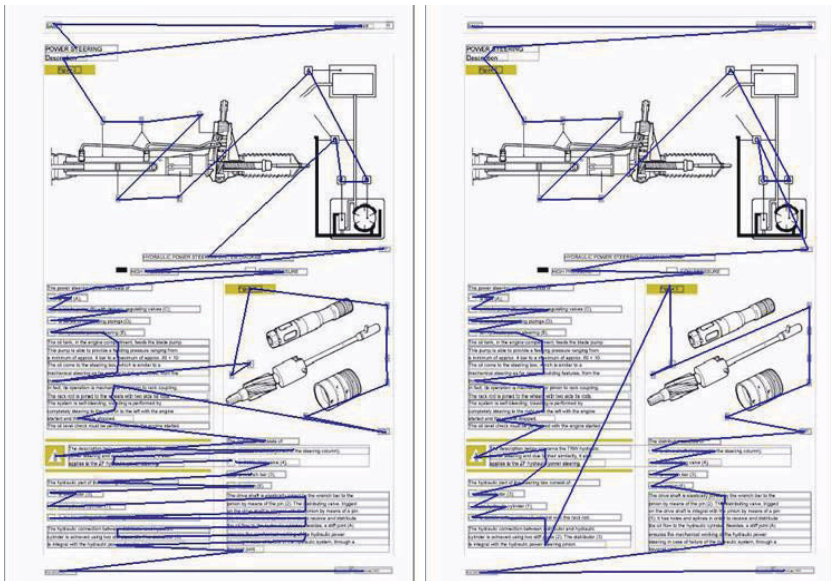


Fig. 4. Reading order detection: left) the original PDF text stream; right) the detected reading order

The cutting of each block is determined by optimizing a score function, which rewards the creation of columns. Figure 4 shows an example of the reading order detection. The blocks formed by the method are visualized with surrounding bounding boxes and their centers are shown by a dot. The line linking consecutive blocks in the page forms a polygon that naturally presents the data flow in the document.

4.2 Document Structuring with the Table of Contents

Semi-structured documents often contain a table of contents that provides a logical organization for their content. Based on this observation, we are concerned in this section with the detection and reconstruction of such tables of contents. Unlike all previous approaches [3,9,12,13], we propose generic characteristics of a table of contents (hereafter ToC) together with a set of associated methods in order to detect a ToC from any given document, to recognize its logical organization and to structure the document accordingly. Because of the large variation in the shape and content a ToC may display, we believe that previous description-based approaches are limited to a series of specific collections. We instead choose a functional approach that relies on the functional properties that any ToC intrinsically respects. These properties are:

1. *Contiguity*: a ToC consists of a series of contiguous items that serve as references to other parts of the document.
2. *Textual similarity*: the reference itself and the referred part either are identical or expose a high level of textual similarity.
3. *Ordering*: the references and the referred parts appear in the same order in the document.
4. *Optional elements*: a ToC entry may include (a few) elements whose role is not to refer to any other part of the document.
5. *No self-reference*: all references refer outside the contiguous list of references forming the ToC.

Our method for detecting the ToC consists of three major steps. First, we define links between each pair of text blocks in the whole document that satisfies a textual similarity criterion. Each link includes a source text block and a target text block. The similarity measure we currently use is the ratio of words shared by the two blocks, considering spaces and punctuation as word separators. Any time the ratio achieves a predefined similarity threshold, a pair of symmetric links is created.

Second, all possible ToC candidate areas are enumerated. Each text block is considered as a possible ToC start and is extended until it is no longer possible to comply with the five properties identified above. A ToC candidate is then a set of contiguous text blocks, from which it is possible to select one link per block so as to provide an ascending order for the target text blocks.

Third, we employ a scoring function to rank the candidate tables of contents. The highest ranked candidate for the Table of Contents is then selected for further processing. Currently, the scoring function is the sum of entry weights, where an entry weight is inversely proportional to the number of outgoing links. This entry weight characterizes the certainty of any of its associated links, under the assumption that the more links

initiate from a given source text block, the less likely that any one of those links is a "true" link of a table of contents.

Once the highest ranked table of contents candidate has been selected, we select the best link for each of its entries by finding a global optimum for the table of contents while respecting the five ToC properties. A weight is associated with each link, which is proportional to the similarity level that led to the link creation. A Viterbi shortest-path algorithm is adequate to effectively determine the global optimum.

5 Semantic Annotations

The logical analysis component represents an important step toward the target XML format. Often, the logical annotations should be further extended to the annotation of semantic entities, accordingly to the target schema. In cases where the semantic entities has a simple form, writing hand-crafted rules in the form of regular expressions can be sufficient for capturing entities in the source documents. As an example, figure references in the example collection (see Figure 3) are 5-digit numbers which are easily recognizable by a simple regular expression.

In more complex cases, methods of machine learning can be deployed to infer entity annotation rules. The principle of supervised learning requires an effort of annotating a subset of samples, building a model and training the model parameters from the available corpus. Given a corpus of annotated instances from source files, we can train a model of semantic annotations and to apply it to non-annotated documents. In the *Leg-DoC* project, the annotation model is a probabilistic model for labeling leaf nodes in tree structure of XML with annotation classes from a set C .

We follow the *maximum entropy* principle, according to which the best model for estimating probability distributions from data is the one that is consistent with certain constraints derived from the training data, but otherwise makes the fewest possible assumptions. In the probabilistic framework, the distribution with the "fewest possible assumptions" is one with the highest entropy, and closest to the uniform distribution. Each constraint expresses some characteristic of the training data that should also be present in the learned distribution [2]. The constraint is based on a binary feature, it constrains the expected value of the feature in the model to be equal to its expected value in the training data.

One important advantage of maximum entropy models is their flexibility, as they allow to easily combine syntactic, semantic and pragmatic features. Each feature f is a binary and can depend on class $c \in C$ and on any properties of the leaf nodes n in a document. In the case of conversion, we add the *content features* that express properties on text in leaves, like $f_1(c, n) = "1 \text{ if } c \text{ is Warning and the content of node } n \text{ has only numeric characters, } 0 \text{ otherwise}"$, as well as the *layout features*, like $f_2(c, n) = "1 \text{ if } c \text{ is Item and } x \text{ position of } n \text{ is } < 250, 0 \text{ otherwise}"$.

With the constraints based on the selected features $f(n, c)$, the maximum entropy method attempts to maximize the conditional likelihood of $p(c|n)$ which is represented as a log-linear model:

$$P(c|n) = \frac{1}{Z_\alpha(n)} \exp \left(\sum_\alpha \lambda_\alpha \cdot f_\alpha(c, n) \right), \quad (1)$$

where $Z_{\alpha}(n)$ is a normalizing factor to ensure that all the probabilities sum to 1.

In the experiments with the example collection, we include two groups of features capturing different annotation types that refer to content fragments in raw or intermediate XML. First, we extract *content features* that concern the content fragments themselves, like the number of words in the fragment, its length, presence of different textual separators, etc. The second group is *layout features*, that on one side, include all layout features of raw XML, like father's tag names, values of attributes for x and y positions, font names and size. On the other hand, this group includes all attributes detected by the logical analyzer, like headings, paragraphs, etc.

6 Visualization, Annotation and Conversion Support

The conversion process is designed as a sequence of transformations where any step depends on the output of previous steps. To ensure the correct final results of the transformation chain, it is desirable for the conversion designer to routinely verify the results at any step of the process.

The visualization component in the *LegDoC* helps achieve several goals. The prime goal is to verify and to visually validate the intermediate and final XML results. Second, the visualization helps debug the transformation chain. In particular, it simplifies the error detection on steps that require an adaptation or a parameter tuning for ensuring a correct output. Finally, the visualization simplifies the preliminary analysis of documents and the generation of different guesses for the conversion design.

We widely deploy generic visualization tools, such as the Adobe Acrobat Reader for PDF files, the XmlSpy editor for XML sources, image viewers and converters for image files, etc. Additionally, we develop specialized solutions for visualizing raw XML, as well as the logical and semantical annotations on the intermediate steps. The visualization component in the *LegDoC* project offers three main functionalities. The first one is the *annotation highlighting* that uses bounding boxes of different colors to present logical and semantic annotations stored in node attributes. The second functionality is *the reading order rendering*; this is implemented as a sequence of highlights and arrows linking in one polygon (see Figure 4). The polygon-based rendering is easy to interpret and to validate the correctness of the reading order. Finally, the visualization provides an extra index for *easy navigation* through pages and *zooming* of page components.

Annotation. Beyond the visualization, we have developed a special annotation component, whose role is to provide a set of manual annotations of the source documents. These annotations can serve, on one side, to automatically test any unsupervised method, like the reading order detection; on the other side, the manual annotations can form a training set for supervised learning methods. During the annotation process, the XML document is updated by adding target tags or attributes where appropriate.

The annotation component is organized around a customized version of the Amaya editor. The major advantage of Amaya consists in providing a triple view on XML documents. These views include a *structural view* of the XML file, a *source view* (the serialized XML), and a *WYSIWYG view* which is an application of the style-sheet to the file. All views are synchronized; a selection in one view is synchronously reflected in two other views. Moreover, Amaya allows one to make annotations on document

fragments. All annotations in Amaya are referred to by XPointer expressions which can be precise up to the character level. Each annotation is a RDF-based description that is stored either locally on the hard disk or remotely on an annotation server.

Manual annotations created with the help of Amaya are managed in the form of the document enrichment; they do not change the hierarchical structure of the document, but injected the annotations as *attribute=value* pairs in the corresponding nodes.

A screen-shot in Figure 5 shows up a synchronized selection in both WYSIWYG (left window) and structural (right window) views of the current document, together with the associated RDF description (window in the left bottom corner).

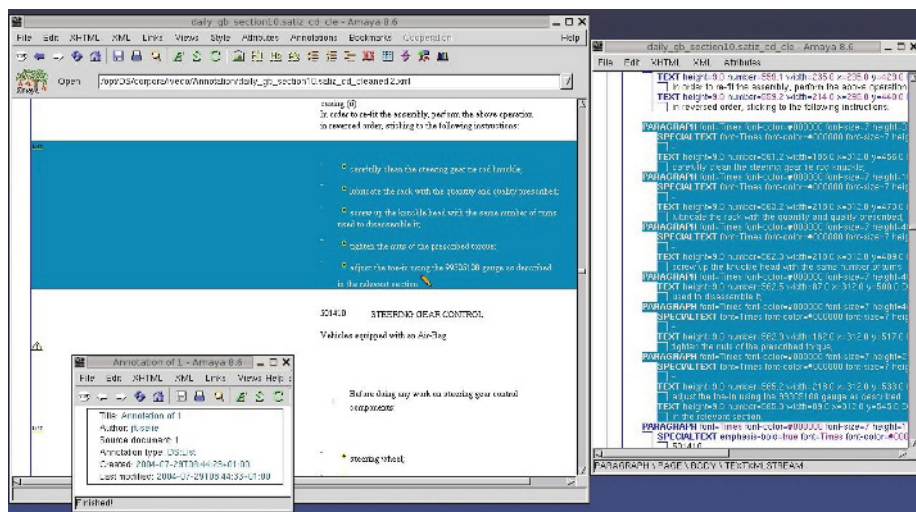


Fig. 5. The document annotation with the customized Amaya editor

Conversion support. Any conversion process is a sequence of steps, where each step is either a document enrichment or document transformation. The *enrichment* does not change the document structure, but enriches it by adding *attribute=value* pairs to some XML nodes, where *attribute* is an index entry, logical or semantic annotation type. The raw XML indexing, the manual and semantic annotations are examples of the document enrichment. Unlike the enrichment, any *document transformation* does change the structure of the document. Cleaning of raw XML, the reading order inference, the ToC-based document structuring do transform the document in question. The most important transformation takes place on the last step of the conversion process. The final XML format is obtained by flopping its layout structure inherited from the original raw XML and transforming the logical and semantic annotations kept in the *attribute=value* pairs as the full-fledged target XML.

Evaluation. The successful application of the *LegDoC* prototype largely depends on the performance of its components, in particular, that of the logical analysis and semantic annotation. All methods for the reading order, the ToC detection and the semantic

annotation have been tested on an important set of different collections and evaluated with the classical measures of the precision and recall.

For the example collection (see Figure 2) which includes 11 documents with 1500 pages in total, all components successfully achieved the required level of 98% in both precision and recall. For the ToC detection, the target has been achieved with the textual similarity threshold set to 50% and a tolerance for 3 optional elements. Moreover, these two values appear valid for most document collections, including in different languages. The semantic annotation of the collection addresses the recognition of elements of eight classes; it required the annotation of around 4% of samples in order to train the maximum entropy model capable to achieve the performance target.

7 Prior Art

Since the importance of semantic annotation of data has been widely recognized and the W3 Consortium published the first XML recommendation in 1998, the migration of documents from legacy and rendering-oriented formats toward XML has become an important research issue [5,6,7,11,15,16]. One important part of research concerns the XML annotation of Web HTML pages. A majority of approaches makes different kinds of assumptions about the structure of source HTML and target XML documents [5,6,16]. In [15], source documents are dynamically generated through the form filling procedure, like on the Web news portals. The domain ontology available on the portal allows to semantically annotate or classify the dynamically generated documents.

The document transformation to and from XML format can be archived by using various tools developed around XML standards and recommendations issues by W3C, in particular the family of XSLT and XQuery languages [17,18]. All these methods cope with a particular approach to the creation of transformation rules, either manually [7] or automatically using a selected heuristics [11].

The research on document analysis also addressed producing XML documents from document images. [1] covers multiple extensions of the OCR (optical character recognition) technology to the recognition of logical structure of documents. It addressed the hierarchical decomposition of a page image and recognition logical elements, like paragraphs, images, titles, etc., by developing a number of extensions of the basic XY-cut algorithm. Ishitani [10] went further in the conversion process; the hierarchical output of the logical analysis is used to generate a pilot XML format, which is manually mapped on the next step to the user-specific XML schema.

8 Conclusion and Future Work

We have presented the Legacy Document Conversion project for automating the conversion of legacy documents to XML according to the generic or domain-specific schema descriptions. Our approach is based on a chain of document transformation and enrichment steps, that allow to gradually migrate the documents from layout to logical and semantic annotations.

In the future work, we plan to extend the current prototype along several dimensions. We intend to improve the existing components and to add new methods to cope

with special structural components, like tables and diagrams. We plan also to further reduce the cost of semantic annotations by implementing the principle of active learning. Finally, we intend to propose a high-level description language allowing to define and validate the conversion chain for any individual conversion task.

References

1. O. Altamura, F. Esposito, and D. Malerba. Transforming paper documents into XML format with WISDOM++. *IJDAR*, 4(1):2–17, 2001.
2. A. L. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.
3. F. Le Bourgeois, H. Emptoz, and S. Bensafi. Document understanding using probabilistic relaxation: Application on tables of contents of periodicals. In *ICDAR*, 2001.
4. R. Cattoni, T. Coianiz, S. Messelodi, C.M. Modena. Geometric layout analysis techniques for document image understanding: a review. Technical Report #9703-09, ITC-IRST, 1997.
5. N. Sundaresan C. Y. Chung, M. Gertz. Reverse engineering for web data: From visual to semantic structures. In *18th Intern. Conf Data Eng. (ICDE)*, 2002.
6. J.R. Curran and R.K. Wong. Transformation-based learning for automatic translation from HTML to XML. In *Proc. 4th Austral. Doc. Comp. Symp. (ADCS)*, 1999.
7. M. Penttonen E. Kuikka, P. Leinonen. Towards automating of document structure transformations. In *Proc. ACM Sym. on Doc. Eng.*, pages 103–110, 2002.
8. J. Ha, R.M. Haralick, and I.T. Phillips. Recursive X-Y cut using bounding boxes of connected components. In *ICDAR*, 1995.
9. F. He, X. Ding, and L. Peng. Hierarchical logical structure extraction of book documents by analyzing tables of contents. In *Proc. of SPIE-IS&T Elect. Imaging, SPIE Vol. 5296*, 1995.
10. Y. Ishitani. Document transformation system from papers to xml data based on pivot xml document method. In *ICDAR*, 2003.
11. L. Kurgan, W. Swiercz, and K.J. Cios. Semantic mapping of XML tags using inductive machine learning. In *Proc. Intern. Conf. Machine Learn. and Applic.*, pages 99–109, 2002.
12. X .Lin. Text-mining based journal splitting. In *ICDAR*, 2003.
13. X .Lin. Automatic document navigation for digital content re-mastering. Master's thesis, HP, 2003 Technical report.
14. G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In *Intern. Conf. Pattern Recogn.*, 1984.
15. I.V. Ramakrishnan S. Mukherjee, G. Yang. Automatic annotation of content-rich web documents: Structural and semantic analysis. In *Intern. Sem. Web Conf.*, 2003.
16. Y. Wang, I. T. Phillips, and R. Haralick. From image to SGML/XML representation: One method. In *Intern. Workshop Doc. Layout Interpr. and Its Applic. (DLIAP)*, 1999.
17. XQuery 1.0: An XML query language, <http://www.w3c.org/TR/xquery/>.
18. XSL Transformations (XSLT) version 1.0, <http://www.w3c.org/TR/xslt/>.

SCOPE – A Generic Framework for XML Based Publishing Processes

Uwe Müller and Manuel Klatt

Humboldt-Universität zu Berlin,
Computer- und Medienservice,
Unter den Linden 6,
10099 Berlin
{U.Mueller, Manuel.Klatt}@cms.hu-berlin.de

Abstract. One of the objectives of the Open Access movement is to establish institutional repositories at universities and other research institutions in order to support self-archiving. Although a lot of software solutions have already been presented in recent years they lack a seamless integration of authoring tools, support for authors, and other technical publication tools. This paper presents a formal approach to describe software components applied in publishing processes. Additionally it is depicted how this formal description leads to the technological basis for SCOPE (Service Core for Open Publishing Environments) – a publishing platform for XML based publishing models. SCOPE is a framework intended for the integration of different publication components into a single platform.

1 Introduction

Presently scholarly communication including scholarly publishing faces one of the most extensive revolutionary processes of its history so far [1]. Basically, this is determined by the fact that publication processes rest upon and are realized as well by electronic technologies and infrastructures to a growing extent. As a main condition, the spreading of the internet has facilitated a much faster dissemination and a wider reach of information and thus also serves as the basis for a new method of scholarly communication.

Obviously, the technological changing does not only concern the creation process of a publication. As a matter of fact this first phase of the publication chain has been nearly completely aided by computers and word processing systems for decades now. But electronic publishing involves more aspects than that such as the submission and review processes, metadata handling and retrieval mechanisms, conversion and transformation tools for different presentation and archival formats (including paper printing), and long term preservation. Nevertheless, the creation phase of a publication is crucial for the following steps and the technological quality of the overall electronic publication. For these reasons the technical qualification of authors as well as the development of appropriate authoring tools to support them became very important issues.

Coinstantaneously to the technological variation of publication processes, the organizational and economic perspective of publication models is changing as well. Traditionally it has been the duty of commercial publishing houses to organize and manage the publication process. Nowadays this task falls – at least partially – to the research institutions. Mainly driven by the Open Access movement, universities and other scientific institutes newly play the role of publishers and therefore have to establish appropriate infrastructures. Among others [2, 3, 4] the adoption of the *Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities* [5] marks a significant milestone on the way to the introduction of institutional Open Access servers.

As the establishment of institutional repositories [6] is one of the main objectives of the Open Access movement several according software solutions have emerged during the last couple of years. The most important ones are *eprints.org*, developed at the University of Southampton and applied for more than one hundred repositories [7], and *DSpace*, originally designed to manage the digital collections at MIT [8]. While they implement upload and management functions as well as parts of a publishing workflow both systems lack the possibility to integrate technical workflow components, such as authoring and conversion tools, with the aim to model the technological part of the publication process. E.g., none of the established institutional repository software solutions is capable of handling XML documents in an appropriate way (management of document models, conversion from current word processing systems, and transformation to presentation formats).

2 Motivation

Since 1997, the Electronic Publishing Group, based at the Computer and Media Service and the University Library of Humboldt University, deals with problems of scholarly electronic publishing. Among other things, the group has established a certified document and publication server (*edoc server*)¹ for Humboldt University. Starting with electronic theses and dissertations, both, technical guidelines and a policy for the *edoc server* [9] have been developed, defining quality standards for scholarly electronic publishing.

One of the most distinctive insights of the first projects launched in conjunction with this subject² is the strong recommendation to use a structured and open document standard as the base file format for electronic publications. The main indicators for this decision are the desire to be company independent, to allow different presentation formats, to enable high quality retrieval and document browsing, and to accomplish the basis for long term preservation activities. Primarily, these considerations led to SGML as the central data format. Later, XML has emerged as the standard for structured document formats.

According to this understanding, various document models, template files, conversion programs, presentation styles and other tools have been developed for numerous different publication types and individual requirements. Different publishing workflows have been implemented centering on SGML / XML.

¹ Edoc server of Humboldt University Berlin: <http://edoc.hu-berlin.de/>

² Projects *Digitale Dissertationen* (1997-2000) and *Dissertationen Online* (1998-2000)

While the overall amount of documents published at the edoc server has been manageable until a short time ago the advancing debate on Open Access causes sensibly increased demands for electronic publishing of scholarly documents on the institutional repository. Likewise, apart from university affiliations the established publishing services are now offered to external research institutions that are lacking the organizational and technological infrastructure to establish their own robust publication server.

Among other things, this results in a more complex variety of related and dependent tools and different workflow models. Moreover, for reasons of costs and complexity, there exists a need to assign some of the operations originally accomplished by the working group's staff back to the institutes, chairs and editors who are responsible for the publications. E.g., it is necessary to enable editors to autonomously manage their publication series. Certainly, this has to happen in a convenient and user friendly way.

Thus, the requirement to more formally manage and describe the different tools, their versions, properties and relations to each other has emerged. Furthermore, the workflows modeling the publishing processes have to be configurable more extensively and easily. The subsequently presented SCOPE architecture aims at satisfying these needs.

The motivating vision of SCOPE is to provide a service platform that meets user defined requirements and preconditions and is able to deliver all necessary tools and a workflow definition enabling a scientific editor to conveniently realize and manage his or her own XML based publication series.

3 Publication Components

In order to formally describe the technological parts of publishing processes SCOPE has defined *publication components*. A publication component (PC) is a distinguished software tool responsible for an atomic step within a publication chain. In most cases it is a publication component's abstract function to transform a document from one defined state to another. Besides, there are tools intended to validate certain document properties and components to generally support particular publication steps – especially the creation phase of publications.

Describing a publication process on the basis of PCs turns out to be an especially sensible part if the workflow is very technology centered. Typically, this is appropriate in case of XML based publication processes, where a multitude of validation, conversion, and transformation steps is necessary for each single document. As indicated in the preceding sections the application of XML as the core document format is considered to implicate many benefits in terms of high quality publishing.

Below, we will enumerate typical classes of publication components, each with some examples already developed and applied within SCOPE:

- *Document models*. They take the central position within the SCOPE framework. Up to now, a couple of different Document Type Definitions (DTD) is available – among others the Dissertation Markup Language (DiML). Moreover, a DTD generation system has been developed, allowing individual compilation of DTDs and

according example documents as well as reference manuals (see section *Workflow System*).

- *Authoring tools*. The best known representatives of this class are document styles and templates for common text processing systems. They enable authors to write their publications in a structured way and additionally keep ready some basic check facilities. The proper usage of template files allows an accurate conversion of the documents to XML. Appropriate document styles and templates for the existing document models are available within SCOPE for Microsoft Word and StarOffice / OpenOffice. Although the templates contain an own bibliographic management system SCOPE also supports current commercial products such as Endnote and Reference Manager. In order to transform the bibliographic information into valid XML segments, a variety of particular styles has been developed for these systems, integrating both, the layout and the structure of the references.
- *Validation and correction tools*. As a counterpart for the offered authoring tools, we have developed a variety of validation and correction tools. They are responsible to verify submitted documents in respect to formal correctness, and – automatically or user supported – to correct problems and formal errors. For example, there are macros to validate the proper usage of templates, the correct handling of images, etc. There also exists a script inspecting submitted PDF files, e.g. the proper embedding of used fonts. Partially, these tools have been integrated into the authoring tools to allow creators to independently validate their files. Other tools are integrated into the submission site of the document server.
- *Conversion tools*. In order to convert documents from one file format to another, a variety of conversion tools has been developed. Among others, scripts have been implemented to convert Microsoft Word or OpenOffice / StarOffice files into valid XML files according to the respective document models. On the other hand, presentation tools, such as XSLT scripts and XSL-FO styles have been developed to allow dynamic or static generation of HTML and PDF output based on the underlying XML sources.
- *Metadata tools*. Metadata plays a vital role in terms of indexing, detection and retrieval of electronic documents. Within the SCOPE framework metadata is managed with the aid of a metadata database and widely configurable input and search / browsing interfaces. These tools can be used by authors and staff as well as by researchers, using the publication server as an information source. Some of the metadata – especially technical metadata – can be extracted automatically from the received documents by according extraction tools.
- *Long term preservation tools*. To transparently ensure integrity and authenticity of the uploaded documents SCOPE uses electronic signatures. For this purpose, partially, tools by a commercial contractor are used. Other components have been self-developed. Development of further long term preservation components will be one of the upcoming businesses of SCOPE. Particularly, we intend to implement tools conforming to the OAIS reference model approach [10].

Most of the PCs available in the existing system have been developed or deployed within SCOPE. But basically, the introduced framework is entirely open to existing or externally developed tools and components.

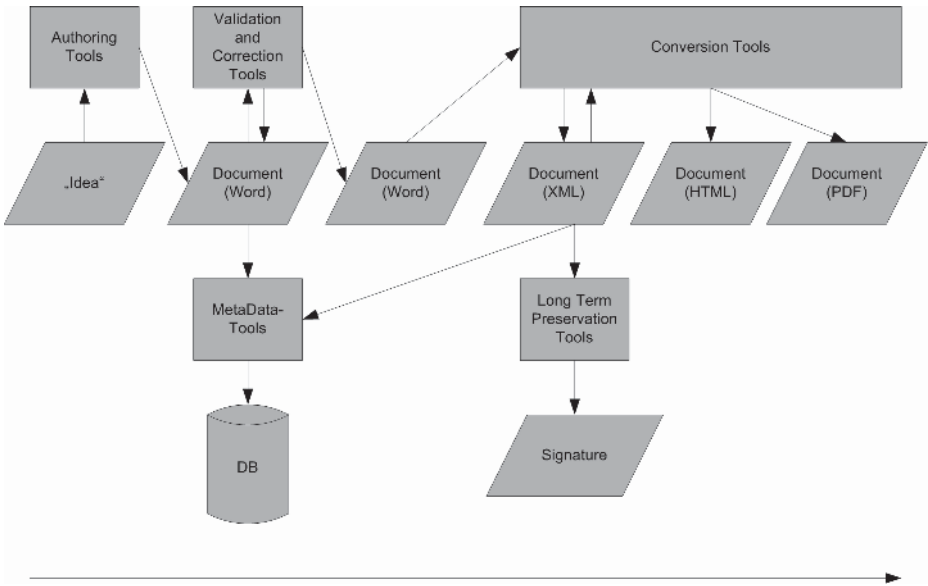


Fig. 1. Example document workflow using publication components

Fig. 1 provides a simple example process showing the way of a scholarly publication from the idea to the final presentation version. In this case the Word document comes into existence with the aid of an *authoring tool*, e.g. a template. Subsequently, it is transformed by a variety of validation and correction tools before it is converted to XML and afterwards to HTML. The PDF version could be generated directly from the Word file or from the XML source using XML-FO. Moreover, metadata and digital signatures are created by the respective tools.

As will be shown in the remaining sections the PCs’ approach is suitable for abstract specification of publishing workflows. It represents the basis for both, the component management system (see next section) and the overall workflow system (see section on *Workflow System*).

4 PC Management System

The publication components necessary to convert documents from one format to another do not exist as context free modules or tools. They depend on each other and may be related to each other as the following examples show: A certain style file is designed for a peculiar DTD A; it could also be used in conjunction with DTD B and Schema A’ but is not applicable for documents structured with DTD C. Since they may have been derived from another component and inherited some properties, they may exist in different versions applicable under different conditions and the like. In order to provide a publication service to editors and authors on the basis of the aforementioned PCs a management system has been developed.

As indicated in the preceding section, a PC is basically characterized by the sort of files it is able to process – more precisely – by the properties of its source and target files.

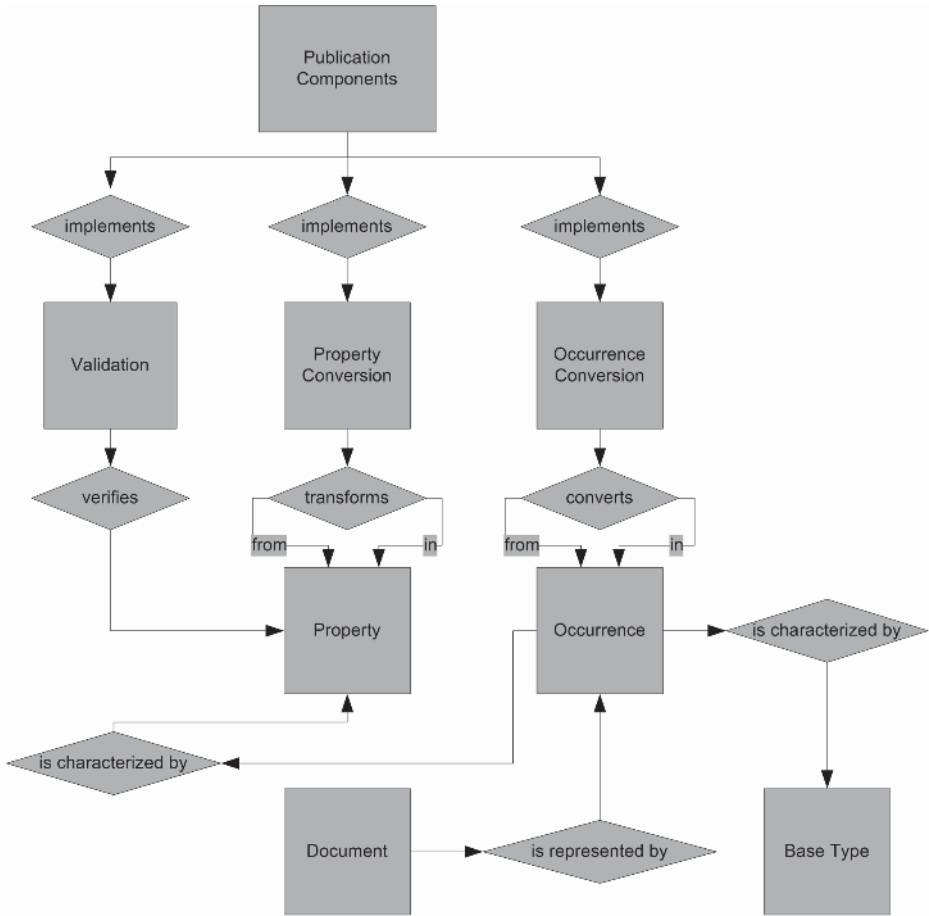


Fig. 2. Correlation between PCs and documents, illustrated in an entity relationship diagram

Fig. 2 depicts the simplified formal relationship between PCs and documents using an entity relationship (ER) diagram. While a *document* is interpreted as a differentiated entity of intellectual output only defined by its content and logical structure, it can be represented by different *occurrences*, e.g. a Word or XML file with certain properties or a paper copy. An occurrence of a document is characterized by a *base type*, i.e. the file format (e.g., MS Word, PDF), and a set of attributes or *properties*, e.g. the page setup, the used fonts and templates, and the type of the embedded pictures. A *publication component* (see previous section) is able to transform certain properties into each other or to convert one occurrence into another one. There are also PCs, e.g. validation tools, which only verify properties and do not change them.

The ER diagram shown in Fig. 2 depicts this correlation by the entities *property transformation*, *occurrence conversion*, and *validation*.

Based on the formal description approach for PCs indicated in Fig. 2 a database model and an according *PC management database* (PC-DB) has been developed mapping the aforementioned interrelations between publication components and documents. Basically, the database serves to contain the formal characteristics of PCs, which are substantially more complex than explained in the preceding paragraphs. I.e., each PC developed or used within SCOPE has been described on the basis of the presented formal approach and stored in the PC-DB.

Among other things, this step provides a big advantage for both, the developers of PCs and the staff being involved in advisory services for publishers: Using simple database requests up-to-date PCs can be very easily and conveniently retrieved and detected according to the respective requirements in case they already exist. Otherwise, the unsuccessful request to the PC-DB simultaneously provides a formal basic description of the PC to be developed.

Thus, the most evident purpose of the PC-DB and the underlying description approach is to efficiently manage a growing and increasingly complex variety of publication components.

Clearly, there are a lot of interrelations and dependencies between the respective considered PCs. Apart from the different versions representing the varying status of development, PCs are primarily tied together by the specific properties and occurrences they are able to process. E.g., a PC implementing a property conversion with certain source and target properties can be substituted by another PC with equal characteristics.

However, the most interesting application of the presented system and the dependencies of PCs is the implicit modeling of whole technological publishing processes. Interpreting the target format of a PC as the source format of another, it is eventually possible for the management system to adequately respond to requests such as “*Give me all PCs necessary to electronically publish MS Word files in a certain HTML layout.*” Basically, this request is translated into recursively arranged simple database requests as mentioned before and leads, if available, to a chain of PCs forming the publishing process starting from the originating MS Word file and ending up with the specified HTML layout.

In general, using this system it is possible to determine, which tools are necessary to convert a document with certain properties to a document with other properties, occurrence or presentation formats. The management system has been implemented prototypically to study its functionality. The formal specification of PCs and the PC-DB is also used for the SCOPE workflow system presented in the next-but-one section.

5 DTD Management System

The aforementioned PC management system’s primary focus is on tools and word-processing properties. As discussed in the first two sections, documents are converted

into XML by default, for reasons of high quality retrieval and long-term preservation. As there can quite easily be understood, different XML tags may be necessary to fully describe different documents.

A simple example is a document with mathematical formulas. To express and tag formulas correctly the underlying DTD must be adapted. Other documents may not consist of formulas at all. For these documents a simpler DTD without formula support is sufficient. One solution to accomplish the various requirements is to provide a universal DTD to accommodate all different documents. However, this would lead to a high complexity and difficult maintenance.

For this reason we have chosen a different approach and have developed a DTD generation system, *DTDSys*, allowing for the compilation of individually assembled DTDs. Therefore, the elements of the virtual “*universal*” DTD have been grouped into modular units. These modules which are XML files themselves are stored in the *DTDBase*.

Using *DTDSys* – a transformation system on the basis of XSLT and Java – the modules can be combined to an individual DTD – e.g. the xDiML DTD used for Theses and Dissertations. Due to its modularity the system can easily be used to supply new publication series with appropriate DTDs, which can contain special elements and which are as slim as possible and therefore more easily applicable than a universal DTD. On the other hand, due to the re-usable modules, the generated individual DTDs are still interoperable, to a certain extend.

Fig. 3 presents the basic structure of *DTDSys*. *DTDBase*, the underlying database of the system contains the different module files including information and description on the single XML elements and attributes. *DTDBase* delivers an HTML file (*dependencies.html*) enclosing the necessary information of all available modules including dependency information. Using a current web browser, a user can compose an individual DTD selecting the required modules which subsequently are stored in the *selection.xml* file. On the basis of this XML file and the original DTD modules the *full-dtd.xml* is generated, containing complete information on all selected modules. Out of this file both, the actual DTD – in this case *xdiml.dtd* – and the element reference websites are generated which consist of one PHP file per XML element (e.g. *chapter.php* for the chapter element). Thus, at the push of a button, an individually adapted DTD file and its corresponding web based reference can be created.

The *DTDSys* also facilitates the integration of externally managed (standard) DTDs such as SVG, SMIL, MathML, or MusicML and thereby allows the generation of DTDs with multimedia extensions. The use of a controlled and centrally managed set of modules provides the advantages of shared semantics beyond the borderline of different DTDs – a feature which is used e.g. for qualified full text retrieval. The XML based publishing approach is currently applied for dissertations and master theses, university series publications, as well as a few electronic journals and conference proceedings. Using *DTDSys* various other document models can be realized rather easily.

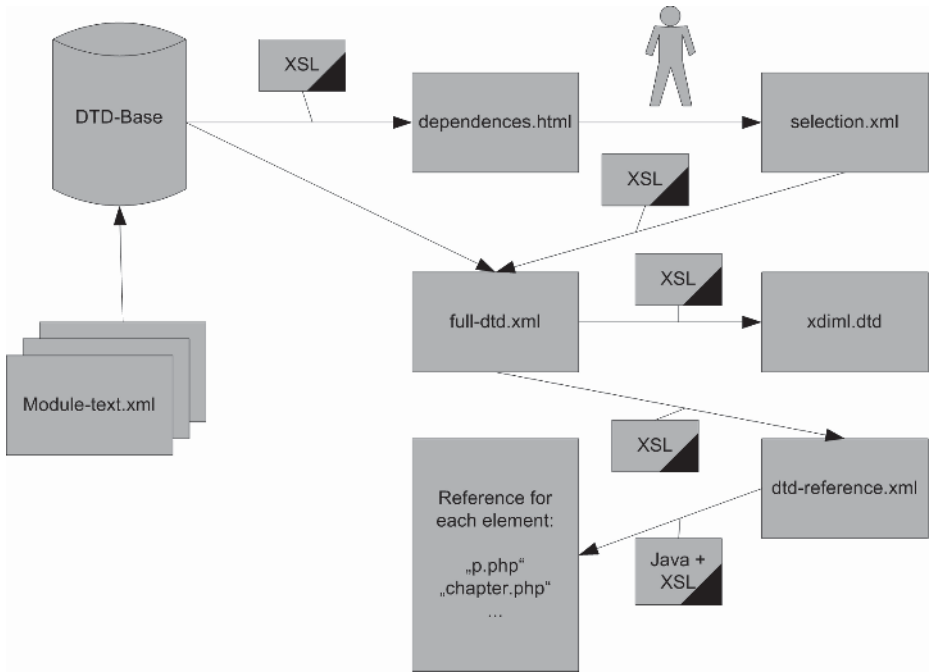


Fig. 3. DTD generation process using DTDSys

6 Workflow System

One of the main developments of SCOPE is the workflow system. It is – in conjunction with the PC management system mentioned two sections before – the centerpiece of the conversion and long term preservation process. Fig. 4 shows the principal functionality of the workflow system. This graphic is based on standard workflow ideas published by the Workflow Management Coalition [11].

The workflow system itself is composed of a *state machine*, a *role model*, and a *work list*. In addition it uses the *file system* and a relational database for storing its *workflow data*. Thereby, the workflow states can be permanently saved. Storing the workflow information other than in the main memory is especially essential to handle long running processes as they are common in publishing environments.

External applications, the *publication components*, that are controlled and whose actions are initiated by the workflow system are found on the right side of Fig. 4. There are PCs which are automatically activated by the workflow engine, and there are PCs partly or completely controlled by human users of the system. The publication components are integrated into the workflow system according to their formal specification. These applications may have access to the *metadata database* and the *edoc server* or can handle user interactions.

The workflow engine is realized by an external state machine which is based on an open source project called Lucille, a project by the Austrian company XiCrypt. It is completely written in Java and allows full state and transition evaluation and control.

As it is necessary in heterogeneous publishing environments the workflows can be easily modeled and altered using an XML based *workflow definition* language. This feature allows quick re-configuration of the workflow using standard XML editors. Additionally there exists a graphical editor for basic configuration. Using the workflow definition language, it is also possible to quickly incorporate new versions of PCs which is necessary for the continuous development of these tools.

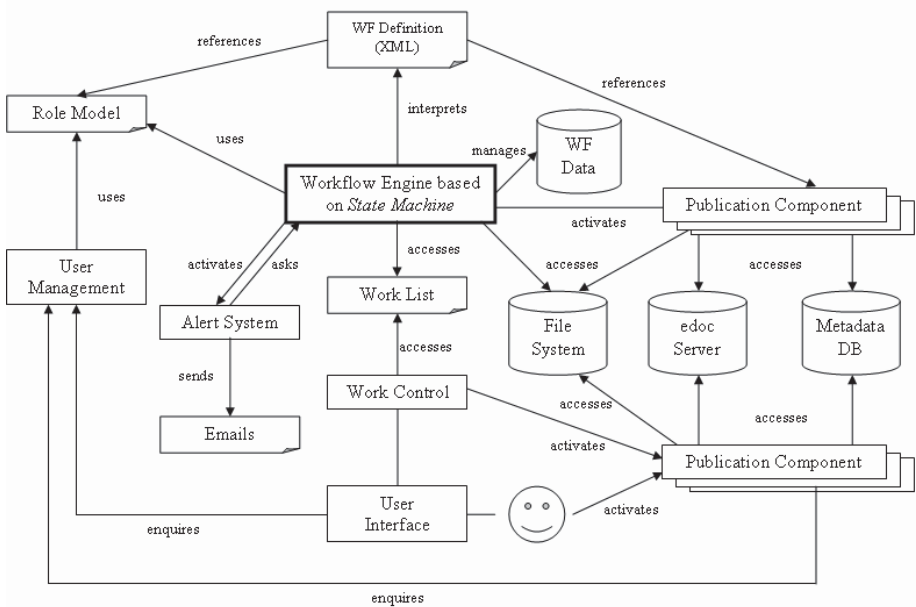


Fig. 4. Overview of the Workflow System

The state machine controls the flow and execution of the workflow. It supports the full spectrum of possible branching and parallelism. Every step in a workflow needs to be implemented using a specific java class. If the performed actions of several steps are alike, they can be implemented by the same java class. Each step is afterwards handled by its own class instance. These classes can implement automatic action or a user interface necessary for human interaction. As mentioned above, many steps in the process of converting somehow need human interaction, due to the fact of the manifold possible errors. We are continuously improving the tools, and some automatic error handling is already possible, but certain aspects as the correct layout of pictures and graphics cannot easily be decided by computers.

As SCOPE is mainly intended for publishers the workflow system needs to incorporate a review process. We have decided to use *GAPWorks*, the workflow system developed in the GAP project [12]. Reusing *GAPWorks* enables us to use a proven technology. To integrate this external workflow system it is completely encapsulated into a single step of the state machine. This step then initializes the *GAPWorks* workflow engine and only communicates using specific file handles with the storage sys-

tem underlying the state machine. The publisher can then use GAPWorks for his review process. If the review process finishes, the GAPWorks workflow ends in a final state, terminates and writes out the result of its reviewing to the file system. Then the state machine step is reactivated and can handle the returned data.

7 Conclusion

With this paper we have presented a generic framework for XML based publishing processes. It is mainly based on a formal approach to describe publication components as technological parts of electronic publishing processes.

The SCOPE architecture consisting of a management system for publication components and a configurable workflow system rests upon this abstract data model. Moreover, a DTD management system has been developed to generate XML DTDs out of individually selected modules.

The seamless integration of authoring tools, transformation and conversion tools, and other technical publication components such as long term preservation and digital signature tools distinguishes the SCOPE architecture from other systems in the market, namely eprints and DSpace. The main difference to these systems is on the one hand formed by the focus on supporting the authors in the publishing and writing process through templates and other support tools. On the other hand the long-term preservation of the documents distinguishes this system from other systems. The use of XML as data format and the publication component management system support the long term preservation efforts. These technologies enable the controlled conversion to new presentation formats and styles. Additionally the usage of the GAPWorks review component with its flexibility and easy configuration allows peer-reviewing and helps to guarantee the publication of high quality works.

While the management system for publication components has been realized prototypically the workflow system is just being implemented.

References

- [1] Shi, Y; Sosteric, M.; Wenker, O. (2001): The Upcoming Revolution in the Scholarly Communication System. *The Journal of Electronic Publishing*. 7 (2), December 2001. Accessed from <http://www.press.umich.edu/jep/07-02/sosteric.html> on February 24, 2005
- [2] Budapest Open Access Initiative (BOAI): <http://www.soros.org/openaccess/>
- [3] Crow, R. (2002): The case for institutional repositories: A SPARC position paper. Accessed from <http://www.arl.org/sparc/IR/ir.html> on February 24, 2005
- [4] Harnad, S. (2001): The self-archiving initiative. *Nature*, 410, 1024-1025. Accessed from <http://www.nature.com/nature/debates/e-access/Articles/harnad.html> on February 21, 2005
- [5] Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities, October 2003. Accessed from <http://www.zim.mpg.de/openaccess-berlin/berlindeclaration.html> on February 24, 2005
- [6] Chan, L. (2004): Supporting and Enhancing Scholarship in the Digital Age: The Role of Open Access Institutional Repositories. *Canadian Journal of Communication*, 29:pp. 277-300. Accessed from <http://eprints.rclis.org/archive/00002590/> on February 22, 2005

- [7] eprints.org, developed at the University of Southampton. Accessed from <http://www.eprints.org/> on February 28, 2005
- [8] DSpace. Accessed from <http://dspace.org/index.html> on February 28, 2005
- [9] Document and Publication Server of Humboldt University Berlin – policy. Accessed from http://edoc.hu-berlin.de/e_info_en/policy.php on February 28, 2005
- [10] Reference Model for an Open Archival Information System (OAIS). Blue Book. Issue 1 (January 1st, 2002). Accessed from <http://ssdoo.gsfc.nasa.gov/nost/wwwclassic/documents/pdf/CCSDS-650.0-B-1.pdf> on February 28, 2005
- [11] Workflow Management Coalition (1995): The Workflow Reference Model. Accessed from <http://www.wfmc.org/standards/docs/tc003v11.pdf> on February 28, 2005
- [12] GAPWorks, developed by ISN Oldenburg. Accessed from <http://www.gapworks.de> on February 28, 2005

DAR: A Digital Assets Repository for Library Collections

Iman Saleh¹, Noha Adly^{1,2}, and Magdy Nagi^{1,2}

¹ Bibliotheca Alexandrina, El Shatby 21526,
Alexandria, Egypt

{[iman.saleh](mailto:iman.saleh@bibalex.org), [noha.adly](mailto:noha.adly@bibalex.org), [magdy.nagi](mailto:magdy.nagi@bibalex.org)}@bibalex.org

² Computer and Systems Engineering Department, Alexandria University,
Alexandria, Egypt

Abstract. The Digital Assets Repository (DAR) is a system developed at the Bibliotheca Alexandrina, the Library of Alexandria, to create and maintain the digital library collections. The system introduces a data model capable of associating the metadata of different types of resources with the content such that searching and retrieval can be done efficiently. The system automates the digitization process of library collections as well as the preservation and archiving of the digitized output and provides public access to the collection through browsing and searching capabilities. The goal of this project is building a digital resources repository by supporting the creation, use, and preservation of varieties of digital resources as well as the development of management tools. These tools help the library to preserve, manage and share digital assets. The system is based on evolving standards for easy integration with web-based interoperable digital libraries.

1 Introduction

The advent of digital technology and high speed networks are leading to widespread changes in services offered by libraries. The heightened user expectations combined with the growth of collections based on digital content makes it increasingly important for all libraries to find efficient tools to manage their digital contents and enable instant access to their digital assets. The Digital Assets Repository (DAR) of Bibliotheca Alexandrina (BA) acts as a repository for all types of digital material and provides public access to the digitized collections through web-based search and browsing facilities. DAR is also concerned with the digitization of material already available in the library or acquired from other research-related institutions. A digitization laboratory was built for this purpose at the Bibliotheca Alexandrina. DAR is built for a library institution; therefore the system adopted a data model able to describe digital objects that include books as well as images and multimedia. Another major objective of DAR is the automation of the digitization workflow and its integration with the repository.

The following goals were driving us while designing and implementing DAR:

- Integrating the actual content and metadata of varieties of objects types included in different library catalogs into one homogeneous repository.

- The automation of the digitization process such that human intervention is minimized and the outputs are integrated within the repository system.
- The preservation and archiving of digital media produced by the digital lab or acquired by the library in digital format.
- Enhancing the interoperability and seamless access to the library digital assets.

The rest of this paper is organized as follows. Section 2 presents some of the related work. Section 3 gives an overview of the system architecture. Sections 4 and 5 present the two main modules; the Digital Assets Keeper and the Digital Assets Factory, respectively. Section 6 presents the tools provided by the system. Section 7 concludes the paper and presents proposed directions for future work.

2 Related Work

There is an increasing number of digital solutions motivated by the increase in the need of preserving and maintaining digital assets.

EPrints [1,2] is a digital repository for educational material that allows authors self archiving their work. A registered user can submit a document to the EPrint archive, the document is described using a super-set of the BibTeX fields. A submitted document is indexed for searching and positioned within a subject hierarchy defined in the system. Dspace [3] is another repository system for handling educational material and depends mainly on Dublin Core records to describe an item. The system defines a workflow for the submission and supports searching, browsing and retrieval. Both EPrints and Dspace implement the OAI-PMH protocol [4]. Greenstone [5,6] is an open source software that provides out-of-the-box solution for the creation and publishing digital material. The system provides easy-to-use interface to define collections of digital objects, the metadata used to describe items within the collection and how items are displayed. According to these configurations, new collections are created and indexes are built for browsing and searching. Greenstone supports different document formats such as HTML, PDF, DJVU and Microsoft Word files. OpenDlib [7] proposes a similar system that aims at providing expandable and searchable system through customizable services.

Commercial library solutions and document management software are used by some libraries and institutions to manage their digital assets. However, most of these systems fail to address interoperability, extendibility and integration with other tools and services in the library due to their proprietary nature.

Contrary to other systems that only manage digital objects or are dedicated to educational material, DAR incorporates in one repository all types of material and formats belonging to the library collections, either born digital or digitized through the system. The DAR data model is capable of describing different metadata sets required by the heterogeneous nature of the collection while still complying with existing and evolving standards. Also, DAR integrates the digitization and OCR process with the digital repository and introduces as much automation as possible to minimize the human intervention in the process. As far as we know, this is an exclusive feature of DAR.

3 System Architecture

The architecture of DAR is depicted in Figure 1. The system core consists of two fundamental modules:

- The Digital Assets Factory (DAF) which is responsible for the automation of the digitization workflow, and
- The Digital Assets Keeper (DAK) which acts as a repository for digital assets.

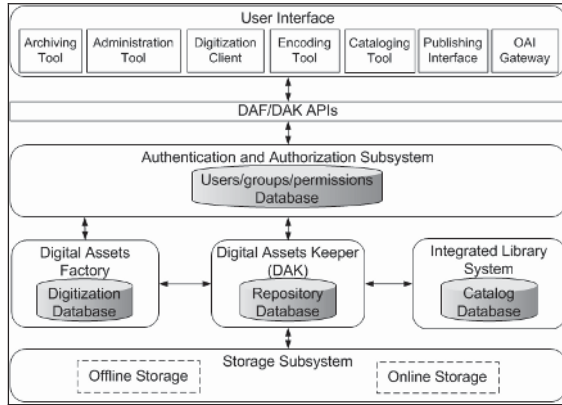


Fig. 1. Architecture of DAR

Both systems interact with the digital objects storage system. The storage system is used to store digital files either for online access and publishing purposes, or offline for long-term preservation. The system contains a set of user interfaces that interact with the system components through APIs. The user interfaces provide tools for the automation of the digitization process, the system parameterization, metadata entry, searching and browsing the repository content, and tools for the interoperability with other repositories. An authentication and authorization system controls the access to the repository contents and functionalities based on the user identity. The repository is integrated with the Integrated Library System (ILS). Plug-in modules control the metadata exchange between the repository database and the ILS database.

The system is implemented in C# using the Microsoft .Net technology. The web-based components are implemented as ASPX pages running on Microsoft IIS web server. The repository APIs are implemented as Web services. SQL sever database is used as the main repository database. The repository is integrated with the Virtua ILS [8] which uses Oracle database on UNIX platform.

4 Digital Assets Keeper - DAK

The DAK acts as a repository for digital material either produced by the digital lab or introduced directly in a digital format. All metadata related to a digital object is stored in the DAK repository database.

4.1 Data Model

One of the challenges faced by DAR is to derive a data model capable of describing all types of library assets including books, maps, slides, posters, videos and sound recordings. For this purpose, two existing standard for data representation have been studied, namely MARC 21 [9] and VRA Core Categories [10]. While the MARC standard is widely used as a data interchange standard for bibliographic data, it is designed mainly for textual materials. Therefore, MARC is seen by the visual resources community as overly elaborate and complex in ways that provide no benefit to visual resources collections, while at the same time lacking or obscuring some concepts which are important to them. On the other hand, VRA core is designed specifically for works of art and architecture that the library is likely to include in its multimedia collections. One of the imaging systems based on the VRA is the Luna Insight [11] which is a commercial imaging software that is widely used by many libraries, universities and museums as a repository for visual assets. The advantages and usefulness of the VRA are discussed in [12]. The data model used by DAR is inspired by the one proposed by the VRA. However, the VRA categories have been extended to accommodate for bibliographic data supported by the MARC standard. This resulted in a data model capable of describing both visual and textual materials in one homogeneous model that is, at the same time, compliant with both standards. The data model is depicted in Figure 2.

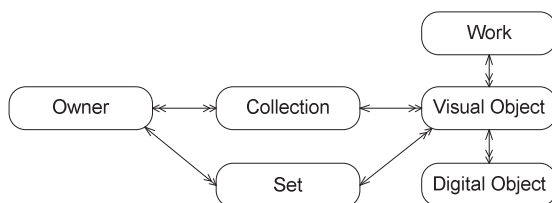


Fig. 2. DAR Basic Data Model

DAR represents a digital object by a *Work* entity related to one or more *Visual Object* entities. This is inspired by the VRA *Work* and *Image* entities. The *Work* refers to a physical entity; it might be a performance, composition, literary work, sculpture, event, or even a building, while the *Visual Object* refers to a visual representation of a *Work*. It can exist in photomechanical, photographic and digital formats. In a typical visual resources collection, a *Visual Object* is a reproduction of the *Work* that is owned by the cataloging institution and is typically a slide, photograph, or digital file. A *Visual Object* exists in one or more digital forms denoted as *Digital Objects*. A *Digital Object* might be a JPG file presenting a scanned slide, an Image-On-Text PDF for an OCR-ed book or an audio or video file.

A *Visual Object* has one *Owner*. The *Owner* is typically an institution, a department or a person. *Visual Objects* related to an *Owner* are grouped into Sets. The *Set* represents a physical grouping of *Visual Objects*; this grouping is established at the digitization phase. On the other hand, the *Collection* represents a descriptive grouping of *Visual Objects* based on a common criteria. Table 1 shows examples of values for each of the described objects.

Table 1. Example of Digital Objects

Object	Value
<i>Collection</i>	Million Book Project, OACIS Collection
<i>Set</i>	Box of 100 slides donated to the library
<i>Owner</i>	Bibliotheca Alexandrina, Yale University
<i>Work</i>	The building of Bibliotheca Alexandrina, The new year concert
<i>Visual Object</i>	A slide of the library building, a video taken in a concert
<i>Digital Object</i>	A JPG file of a scanned a slide, a PDF file for an OCR-ed book

4.2 Metadata

Within the DAR data model, the system holds six categories of metadata describing assets and its digital reproductions, a demonstrative example can be found in [13]:

4.2.1 Descriptive Metadata

This includes metadata common to all types of *Works* and *Visual Objects*, such as:

- Type, for a *Work* object, the type could be a painting, map, event or building. For a *Visual Object*, the type could be a slide, photograph, video, audio or book
- Title
- Creator(s), a creator could be the author, publisher, architect or artist
- Date(s), a date could represent the date of the creation, alteration or restoration

This is as well as keywords, description, dimensions, location, etc. Other metadata that is specific to a *Work* type include fields like the ISBN, language and publisher in the case of books, the technique and material in the case of a work of art.

4.2.2 Digital Content Metadata

This includes metadata describing a Digital Object. DAR supports a variety of digital objects' formats including JPG, TIFF, JPG 2000, PDF, DJVU, OCR Text and others. Metadata such as image resolution, dimensions, profile, or a video duration are extracted from digital files automatically and stored in DAK. New formats can be introduced into the system and appropriate tools can be integrated to deal with the new file formats.

4.2.3 Archiving Metadata

This includes metadata about the archiving location of a Digital Object file. The archiving metadata consists of the archiving media unique identifier. The archiving metadata can also be attached to the Visual Object, denoting the physical location where the object can be found in the owning institution.

4.2.4 Publishing Metadata

Encoded objects for publishing are stored on online storage. The publishing metadata includes the path of the published Digital Object on the server, the date of publishing, duration of publishing as well as the category of targeted users e.g. students, researchers, etc.

4.2.5 Access Right Metadata

Copyright restrictions on the repository contents are forced by defining access rights attached to each object. This consists of a copyright statement linked to the Visual Object. Also, an access right level is used by the system to indicate whether a Visual Object and its related Digital Objects are free of copyright restrictions or not. This level is used by the publishing interface to determine the display of objects; whether to display metadata only, the full digital objects or only part of it.

4.2.6 Authentication and Authorization Metadata

DAR users are identified by a username and a password. Further, user groups are defined where a user can belong to one or more groups. Permissions are given to each user or group, which are checked before accessing an application and/or digital object. User and group rights can be specified on the Visual Object level or, more practically, on the Collection level.

5 Digital Assets Factory – DAF

The DAF governs the digitization process of the library collection at the digital lab. DAF realizes one of the main goals of DAR which is the automation of the digitization process. This supports the digitization of library assets including textual material, slides, maps and others. It provides the digital lab operators with tools for entering a digitization job metadata, keeping track of digitization status, applying validation tests on digitized material, recording productions, archiving the digitized material for long term preservation and retrieving the archived material when needed. The system supports different workflows for different types of material.

After initiating a new job, the asset passes by the general phases depicted in Figure 3:

- Scanning the material.
- Processing the scanned files to enhance the quality.
- Perform Optical Character Recognition (OCR) on the textual material.
- Encoding the digitized material by generating a version suitable for publishing.
- Archiving the output of each step of the digitization. Two offline backups are taken for a file, one on CD and the other on tape. Encoded versions are moved to online storage for publishing.

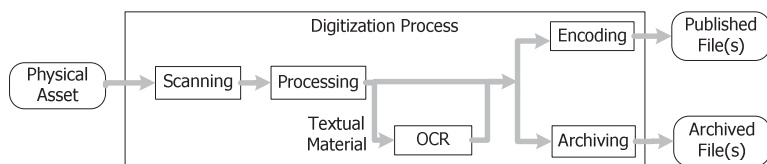


Fig. 3. Digitization Phases

The files and folders produced by each phase are stored in separate queues on a central storage server. A job folder resides in one of the four main queues: *scanned*,

processed, *OCRed* and *ready for archiving* queue. The digital lab operator withdraws jobs from the queues, performs the job and places the output in the next queue in the process. Alternatively, files can be introduced directly into any of the queues, for example an image that is already provided in digital form is placed directly into the *processing queue*. Table 2 shows a one year digitization statistics at the BA digital lab since the deployment of DAF in March 2004.

Table 2. Digital Lab Production Statistics

	Arabic	Latin
Scanned pages	4,591,463	730,141
Processed pages	4,585,833	730,141
OCRed pages	1,148,465	693,978
Scanned Slides	12,013	
Archived Data on CDs/Tapes	480 GB	

The main goals of DAF are:

- To provide a database system to keep track of the digitization process through the scanning, processing, OCR-ing, archiving and publishing.
- To keep track of digitized materials; unifying the naming conventions and exhaustively checking the produced folders and files for consistency.
- To provide timely reports to various levels of management describing the workflow on a daily, weekly or longer basis and to allow online queries about the current status of a certain asset at the digital lab.
- To apply necessary encodings on the scanned materials to be suitable for electronic publishing.
- To manage the archiving and retrieval of the digitized material.

Digitization Metadata

For objects that are digitized using the DAF applications, the digitization metadata is gathered during the different digitization stages, such as the scanning date(s) and scanning operator(s), the processing date(s) and processing operator(s), the OCR font data, the accuracy achieved by the OCR before and after learning, etc...

6 Tools

The DAR system deals with three types of users; digitization operators, librarians - which are divided into catalogers and reviewers - and the end users. Each type of users is provided with tools to make use of the system functionalities.

6.1 Administration Tool

The *Administration Tool* is one of the DAF Web-based tools used by the operator in the digital lab. The tool is used to initiate a new job by entering minimal descriptive metadata for the material to be digitized. If the material is cataloged in the library catalog, the ILS id - a book barcode, for example - is used to retrieve the metadata

from the library catalog. This id is also used to link the record in DAR to the one in the library catalog for future synchronization. If the material is not previously cataloged, the operator enters the minimal metadata that can be deduced from the physical item in hand. The tool uses this metadata to derive a unique folder name for the scanned files. The tool is also used for the system parameterization and to generate reports on production rates and jobs in different digitization queues in the lab.

6.2 Digitization Client – DLClient

The *DLClient* is a DAF application used by the operator in the digital lab. The tool creates structured folders for new digitization jobs and after the completion of each digitization phase, the *DLClient* tool is used to perform the following:

- Validate the files; generate warnings if any inconsistencies are detected.
- Update the job status in the database by setting the username for the operator who performed the job, the job completion date and the count of produced files.
- Move the folders and files to the queue of the next digitization phase on a storage server. Before moving any folder, a lock is acquired on the folder and sub files to avoid concurrent access to the folder while moving.

The *DLClient* is used by the operator through the three main digitization phases; scanning, processing and OCR.

Scanning

Physical assets submitted to the lab for digitization are placed in a *scanning queue*. The operator retrieves a job from the queue and uses the *DLClient* to create the folder structure where scanned files are to be stored. Mainly, a digitization folder contains three subfolders for three types of files: the original scanned files, the processed files and the encoded output. The encoded output, the folder structure and the scanning resolution varies according to the material type; text, image, audio or video. When the scanning is done, the *DLClient* places the produced files in the *processing queue*.

Processing

The operators use the *DLClient* to retrieve a job from the *processing queue*. A combination of manual and automated image processing tools is used to enhance the quality of the scanned images. After the job is done, the *DLClient* places the job at the *OCR queue* for textual material and directly to the *archiving queue* for other types of material.

Optical Character Recognition

Using the *DLClient*, a processed textual material is retrieved from the *processing queue* to be OCR-ed extracting text from the scanned images. OCR is used to enable full text searching. Currently, the system supports Latin OCR using Fine Reader 6.0 from ABBYY [14] and Arabic OCR using Sakhr Automatic Reader [15]. To enhance the recognition quality of Arabic text, BA has built a library of fonts using learning samples taken from different books. Before starting the recognition, the OCR operator matches the book font with the nearest font library.

Reprocessing

The system supports a special workflow for reprocessing a digitized material. Reprocessing may be needed to enhance the OCR quality, to apply new image process-

ing procedure or simply to generate new publishing format of the digitized material. Reprocessing begins by searching and retrieving the files to be reprocessed from the archive. The files are then placed in the appropriate digitization queue. The reprocessed files go through the normal digitization steps described before until they reach the archiving phase. Only altered files are re-archived, changes in files are detected using checksums that are calculated before and after the reprocessing. The archiving information of a new file version is recorded in the repository database and a link is made to the parent version archiving location so that file versions may be tracked in the database from the most recent to the base version.

6.3 Archiving Tool

In the current version, a digital object is represented by one or more files with different formats and/or resolutions, these files are stored for online access on RAID storage system or on offline storage for long-term preservation. Typically, the preserved material is the scanned originals and the processed version with high resolution. Lower versions derived for publishing purposes are saved on online storage for ease of access; this includes low resolution JPG, PDF, and DJVU. Files stored offline are archived on two medias; CDs and tapes. Unique labels are generated, printed and attached to the media for future retrieval. The system keeps track of different versions of a file by linking a newer version to its older one. More sophisticated content versioning and object representation is to be applied in future versions of DAR, this could build on the architecture proposed by the Fedora repository [16].

The *Archiving Tool* is one of the DAF Windows-based applications used by the lab operators and offers the following functionalities:

- Checking files and folders consistency.
- Preparing the folders for archiving by compressing the subfolders and files, grouping them into bundles that fit into the media capacity (CD or tape), generating the media label, printing the label.
- The tool generates checksums for the archived files to detect changes in case of downloading and reprocessing a file.
- A search facility enables the user to retrieve an archived folder by locating the folder, uncompressing the subfolders and files and copying the uncompressed files and folders to a destination specified by the user.
- Managing the space on the storage server hard drives, the tool generates warnings when storage level exceeds a predefined value for each drive.
- The tool updates the DAK database by recording the archiving information related to a digital file.

6.4 Encoding Tool

In the encoding step, a final product is generated for publishing. For images, slides and maps, different JPG resolutions are generated. For audio and video, different qualities are generated to accommodate for different network connections' speed. For textual material like books, special developed tools are used to generate the image-on-text equivalent of the text; this is done on an encoding server built on Linux platform. The Encoding Server encodes digital books into light-weight image-on-text

documents in DjVu and PDF. Support for DjVu is built around DjVu Libre, an open source implementation of a DjVu environment, or, alternatively, Document Express, LizardTech's commercial DjVu product. Support for PDF is implemented based on iText, an open source API for composing and manipulating PDF documents. The Encoding Server supports multilingual content through integration with Sakhr Automatic Reader [13]. The Encoding Server allows for the integration of any OCR engine through writing OCR converters, which transforms the native OCR format into a common OCR format that the Encoding Server is capable of processing along with page images in TIFF or JFIF format to compose image-on-text documents. A generated file is copied to a publishing server, the encoding tool updates the DAK database by inserting the corresponding *Digital Object* record. The record is populated with metadata extracted from the digital files and with the publishing information; publishing server and URL.

6.5 Cataloging Tool

The *Cataloging Tool* is a Web-based application used by the librarian to add and edit metadata in the DAK subsystem. Using the Cataloging Tool, the librarian enriches the digital repository records – created in the digitization phase - by adding metadata. The librarian can also create new records for digital objects and upload their corresponding files. The repository is preloaded with controlled vocabularies lists. The tool allows defining configurable templates, importing metadata from external sources and automatic extraction of digital content metadata.

6.6 Publishing Interface

The *Publishing Interface* is a Web-based interface related to the DAK that provides access to the repository of digital objects through search and browsing facilities.

The repository Publishing Interface offers the following functions:

- Browse the repository contents by *Collection*, *Work Type*, *Visual Object Type*, *Subject*, *Creator* and *Title*.
- Search the content by an indexed metadata field; *Creator*, *Title*, *Subject*, ...
- For textual material, a search in the full text can be conducted. The user can choose whether exact or morphological matching is applied.
- For images, different levels of zooming are available.
- Display brief and full record information with links to the digital objects.
- Display the records in MARC or DC in XML formats.
- Hyperlinked data fields that can invoke searches e.g. by *Keywords*, *Subjects* and *Creator*.

6.7 Integration with the ILS

DAR can be easily integrated and synchronized with external sources – e.g. bibliographic catalog, external repository, imaging systems - by implementing appropriate integration modules. An integration module is a plug-in component designed to export records from DAR to an external repository, or to import records from an external repository into DAR, or both.

The integration module is fully configured based on the following:

1. A record unique identification: This identifier is used as a link between the record in DAR and the one in the external repository.
2. Metadata mapping table: The mapping table defines how data fields are mapped from DAR to the external repository and vice versa including the concept of *Work* and *Visual Object*.
3. Synchronization schedule: This schedule defines how often the two repositories are synchronized. The synchronization process considers only the newly created and modified records.

In the current version, a module is implemented for integration with the Virtua ILS [8] which is deployed at BA.

6.8 Authentication and Authorization

In DAR, a *User* is a member of one or more *Group*. Each *Group* is assigned access permissions on the repository contents and functionalities. A basic username and password scheme is used to identify the user. Anonymous access to the repository is also allowed, the access right of an anonymous user are defined by the permissions assigned to a special group *Guest*. This simple authorization scheme will be augmented in future versions to accommodate for special groups' permissions.

6.9 OAI Gateway

DAR OAI Gateway implements the Open Archive Protocol for Metadata Harvesting developed by the Open Archives Initiative [4] to provide access to the repository contents across an organization's architecture. The Gateway receives XML requests and translates them to the equivalent database queries. When the request result sets are retrieved, the gateway translates them into XML and responds to the requesting application.

The gateway implements the six types of requests required for OAI-PMH compliance: Identify - ListMetadataFormats - ListSets - ListIdentifiers - ListRecords - GetRecord

7 Conclusions and Future Work

We have presented in this paper the DAR system implemented at the Bibliotheca Alexandrina. The system acts as a repository for digital assets owned by the library and associates the metadata with the content to provide efficient search and retrieval. DAR addresses the main challenges faced by digital repositories including supporting different digital formats, digitization workflows, preservation of digital material and content dissemination. The DAF subsystem has been fully implemented and deployed since March 2004. The DAK cataloging and publishing modules are being implemented. It is foreseen that these modules will need several iterations in the implementation based on user feedback and requirements. The beta version will include features presented in this paper. Future enhancements include:

- Building a more sophisticated security system based as on existing and emerging standards that are appropriate for the web services environment.
- Implementing a generic digital assets viewer. The viewer should support different file formats (PDF, DJVU, Images, Video and Audio).
- Joining the Open Source community by making the system source code publicly available and using free-of-charge development tools and database engine.
- Providing query translation tools to enable cross-language information retrieval.
- Using XML format for encoding objects metadata. This will facilitate exchange of objects among repositories.

References

1. GNU EPrints. <http://software.eprints.org/>
2. L. Carr, G. Wills, G. Power, C. Bailey, W. Hall and S. Grange: Extending the Role of the Digital Library: Computer Support for Creating Articles. Proceedings of Hypertext 2004 (Santa Cruz, California, August 2004).
3. R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan and M. Smith: The DSpace Institutional Digital Repository System: Current Functionality. Proceedings of JCDL '03 (Houston, Texas, May 2003).
4. The Open Archives Initiatives. <http://www.openarchives.org/>
5. D. Bainbridge, J. Thompson and I. H. Witten: Assembling and Enriching Digital Library Collections. Proceedings of JCDL '03 (Houston, Texas, May 2003).
6. I. H. Witten, S. J. Boddie, D. Bainbridge and R. J. McNab: Greenstone: a comprehensive open-source digital library software system. Proceedings of the fifth ACM conference on Digital libraries (June 2000).
7. D. Castelli and P. Pagano: A System for Building Expandable Digital Libraries. Proceedings of JCDL '03 (Houston, Texas, May 2003).
8. Virtua Integrated Library System. <http://www.vtls.com/>
9. MARC 21 Standard. <http://www.loc.gov/marc/>
10. VRA Core Categories, Version 3.0. <http://www.vraweb.org/vracore3.htm>
11. Luna Imaging Software. <http://www.luna-imaging.com/>
12. P. Caplan: International Metadata Initiatives: Lessons in Bibliographic Control. Available at http://www.loc.gov/catdir/bibcontrol/caplan_paper.html (2001).
13. I. Saleh, N. Adly and M. Nagi: DAR: A Digital Assets Repository for Library Collections – An Extended Overview. Internal Report available at <http://www.bibalex.org/English/researchers/isis/TR-DAR.pdf>
14. ABBYY Fine Reader OCR software. <http://www.abbyy.com/>
15. Sakhr Automatic Reader OCR software. <http://www.sakhr.com/>
16. S. Payette and C. Lagoze: Flexible and Extensible Digital Object and Repository Architecture. Proceedings of ECDL '98 (Greece, September, 1998).

Webservices Infrastructure for the Registration of Scientific Primary Data

Uwe Schindler¹, Jan Brase², and Michael Diepenbroek¹

¹ World Data Center for Marine Environmental Sciences (WDC-MARE), MARUM, University of Bremen, Leobener Str., 28359 Bremen, Germany
uschindler@wdc-mare.org, mdiepenbroek@wdc-mare.org

² Research center L3S, Expo Plaza 1, 30539 Hannover, Germany
brase@l3s.de

Abstract. Registration of scientific primary data, to make these data citable as a unique piece of work and not only a part of a publication, has always been an important issue. In the context of the project "Publication and Citation of Scientific Primary Data" funded by the German Research Foundation (DFG) the German National Library of Science and Technology (TIB) has become the first registration agency worldwide for scientific primary data. Registration has started for the field of earth science, but will be widened for other subjects in the future. This paper shall give an overview about the technical realization of this important usage field for a digital library.

1 Motivation

In its 2004 report "Data and information", the *International Council for Science* (ICSU) [11] strongly recommended a new strategic framework for scientific data and information. On an initiative from a working group from the *Committee on Data for Science and Technology* (coData) [9], the *German Research Foundation* (DFG) [2] has started the project *Publication and Citation of Scientific Primary Data* as part of the program *Information-infrastructure of network-based scientific-cooperation and digital publication* in 2004. Starting with the field of earth science the *German National Library of Science and Technology* (TIB) is now established as a registration agency for scientific primary data as a member of the *International DOI Foundation* (IDF).

2 Registration of Scientific Data

Primary data related to geoscientific, climate and environmental research is stored locally at those institutions which are responsible for its evaluation and maintainance. In addition to the local data provision, the TIB saves the URL where the data can be accessed including all bibliographic metadata. When data are registered, the TIB provides a *Digital Object Identifier* (DOI) as a unique identifier for content objects in the digital environment. DOIs are names assigned

The screenshot shows the TIBORDER library catalogue interface. At the top, there are navigation links for 'Search', 'Results', 'Advanced search', 'Saveset', and 'Help'. The search bar contains the text 'Berger, Wolfgang' and is set to search 'all words' sorted by 'year of publication'. Below the search bar, there are tabs for 'review', 'shortlist', and 'title data'. The main content area displays the search results for 'Berger, Wolfgang', showing a list of results with '2 of 41' items. The first result is expanded, showing the following metadata:

- Title:** [Oxygen isotope stratigraphy Ojplus03 for the last 838,000 years / PANGAEA](#), AWI/MARUM, Germany. **Wolfgang H Berger**
- Collaborator:** [Wolfgang H Berger](#)
- Corporate body:** [PANGAEA](#)
- Published:** 2004-08-23
- Extent:** Online-Ressource (420DataPoints).
- Note:** Mode: Abstract
StructuralType: Digital
- Abstract:** REFERENCE(S): **Berger, Wolfgang H** (2004): Climate future in a warming world: lessons from the ice ages, in: Wernli, RL & Kennel, CF (eds) Oceans 2003 MTS/IEEE Conference Proceedings, Holland Enterprises, Escondido CA, 404-408
PARAMETER(S): AGE [kyr BP] (Age) * Geocodedelta 18O, adjusted/corrected [per mil] (d18O adj) * PI: **Berger, Wolfgang H** (eMail: wberger@ucsd.edu) * METHOD: calculated
- Techn. data:** Format: TEXT
- Full text/Image:** [Display entire document free access!](#)
[URN free access!](#)
- Holding:** [Display free access!](#)
Note: Primaerdaten

Fig. 1. A dataset as a query result in the library catalogue

to any entity for use on digital networks. They are used to provide current information, including where they (or information about them) can be found on the Internet. Information about a digital object may change over time, including where to find it, but its DOI will remain stable. For more information, we refer to [3]. In cooperation with the *German National Library* (DDB) every dataset is furthermore registered at the infrastructure of the project EPICUR [8] with a unique URN. Due to the expected large amount of datasets that need to be registered, we have decided to distinguish between citable datasets on the collection level and core datasets usually are data entities of finer granularity. Core datasets receive their identifiers, but their metadata is not included in the library catalogue whereas citable datasets, usually collections of, or publications from core datasets are included in the catalogue with metadata compatible to ISO 690-2 and Dublin Core (DC, see [10]). The DOI guarantees that these data are generally accessible and are citable inside traditional publications. By this, scientific primary data are not exclusively understood as part of a scientific publication, but have its own identity.

All information about the data is accessible through the online library catalogue of the TIB. The entries are displayed with all relevant metadata and persistent identifiers as links to access the dataset itself (see fig. 1).

3 Infrastructure

A special infrastructure is needed for flexible registration of DOIs for datasets and migration of meta information into related library catalogues. The key ele-

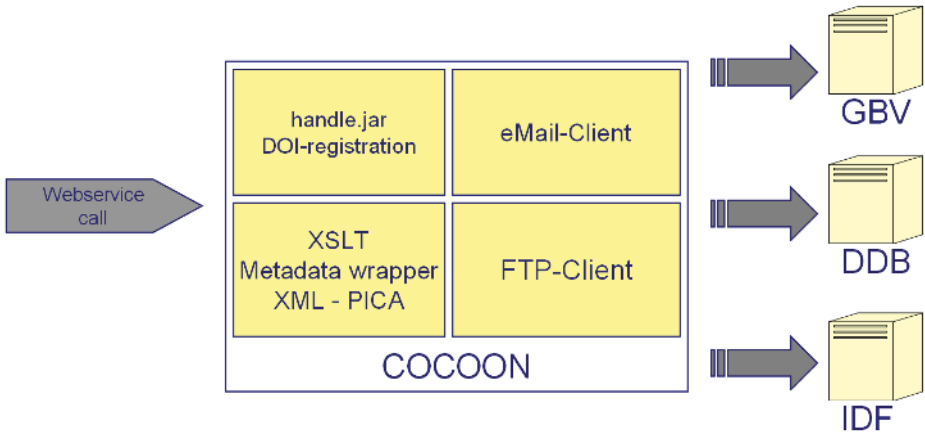


Fig. 2. The architecture of the registration process

ment is a webservice as part of the middleware at the TIB that offers automatic and manual upload of registration information.

Figure 2 gives an overview on the components and the possible workflows.

3.1 Webservice

We have chosen the SOAP (**S**imple **O**bject **A**ccess **P**rotocol) webservice standard for the communication between the data providers and the *TIB* because webservices provide interoperability between various software applications running on disparate platforms and programming languages. This is possible because open standards and protocols are used. Protocols and data formats are XML based, making it easy for developers to comprehend. By utilizing HTTP/HTTPS on the transport layer, web services can work through many common firewall security measures without requiring changes to the firewall filtering rules. This is not the case with RMI or CORBA approaches.

As the STD-DOI webservice is SOAP conformant, data providers can embed the client stub into their middleware by importing the WSDL (**W**eb**S**ervice **D**escription **L**anguage) file into their application server. For the webservice we have identified five different methods:

1. **registerCitationDOI** - For a citable dataset a DOI and an URN are registered
2. **registerDataDOI** - A core dataset only receives a DOI
3. **transformData2CitationDOI** - Upgrade a core dataset to a citable dataset by adding metadata
4. **updateCitationDOI** - If any part of metadata changes for a citable dataset, a new library record has to be created
5. **updateURL** - If the URL of a dataset changes, this information has to be stored at the DDB for the URN and the IDF for the DOI resolution

An excerpt of the WSDL-file describing the Webservice can be seen below:

```
...
<wsdl:portType name="CodataWS">
  <wsdl:operation name="registerCitationDOI" parameterOrder="xml url">
    <wsdl:input message="impl:registerCitationDOIRequest"
      name="registerCitationDOIRequest"/>
    <wsdl:output message="impl:registerCitationDOIResponse"
      name="registerCitationDOIResponse"/>
  </wsdl:operation>
  <wsdl:operation name="registerDataDOI" parameterOrder="doi url">
    <wsdl:input message="impl:registerDataDOIRequest" name="registerDataDOIRequest"/>
    <wsdl:output message="impl:registerDataDOIResponse" name="registerDataDOIResponse"/>
  </wsdl:operation>
  <wsdl:operation name="transformData2CitationDOI" parameterOrder="xml">
    <wsdl:input message="impl:transformData2CitationDOIRequest"
      name="transformData2CitationDOIRequest"/>
    <wsdl:output message="impl:transformData2CitationDOIResponse"
      name="transformData2CitationDOIResponse"/>
  </wsdl:operation>
  <wsdl:operation name="updateCitationDOI" parameterOrder="xml">
    <wsdl:input message="impl:updateCitationDOIRequest"
      name="updateCitationDOIRequest"/>
    <wsdl:output message="impl:updateCitationDOIResponse"
      name="updateCitationDOIResponse"/>
  </wsdl:operation>
  <wsdl:operation name="updateURL" parameterOrder="doi url">
    <wsdl:input message="impl:updateURLRequest" name="updateURLRequest"/>
    <wsdl:output message="impl:updateURLResponse" name="updateURLResponse"/>
  </wsdl:operation>
</wsdl:portType>
...
```

To prevent unauthorized access of the webservice we have chosen HTTPS as transport layer. In the first approach we have simple username/password access restrictions. In the future we want to use client authorization with certificates on the SSL/TLS layer of the HTTPS protocol. Every data provider then gets his own client certificate that can be embedded into the webservices client key store for authorization.

3.2 Metadata Scheme

We identified a set of metadata elements to describe the bibliographic information of our scientific primary data. Whenever possible, we have tried to use Dublin Core (DC) equivalent metadata elements. The metadata scheme can be found in [6] or at the project's webpage [13], it includes all information obligatory for the citing of electronic media (ISO 690-2). For inclusion into the library catalogue, however, we had to convert the XML-based metadata into PICA-format. PICA, an acronym for *Project for Integrated Catalogue Automation* is a cataloguing format based on MARC21 of the *Library of Congress*. It is used in the *Central Library Database of northern Germany* (GBV) that is responsible for the cataloguing at the *TIB*.

3.3 The Infrastructure at the Data Providers – Example PANGAEA/WDC-MARE

The DOI registration webservice client is embedded into the metadata publishing workflow of PANGAEA (Publishing Network for Geoscientific & Environmental

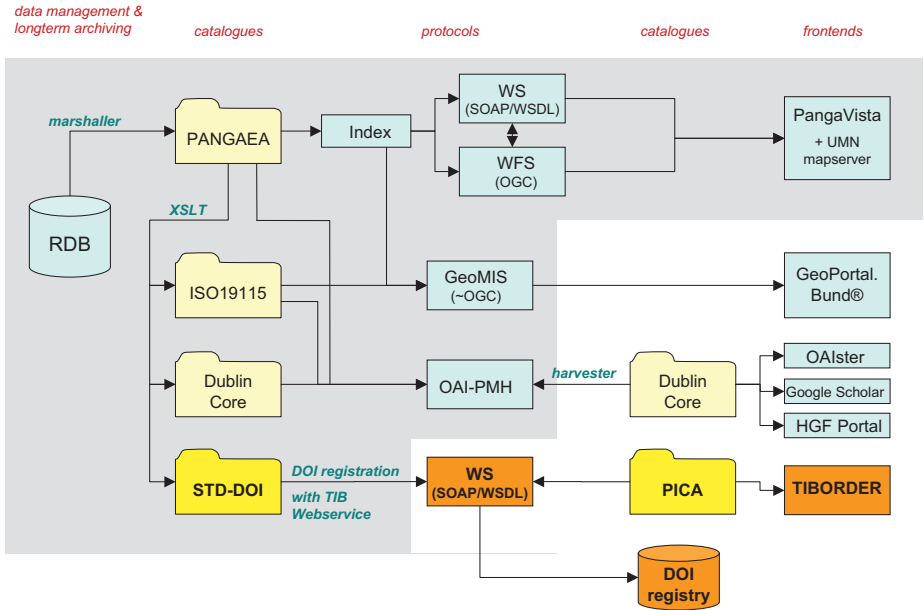


Fig. 3. The STD-DOI webservice embedded into the PANGAEA middleware structure

Data [17]). After inserting or updating a dataset in PANGAEA the import client queues background services which keep the XML metadata repository up to date (see fig. 3).

These background services marshal in a first step the metadata into an internal XML schema. This schema reflects the PANGAEA database structures and is optimized for simple marshalling of database records and transformation into other formats. We have chosen "jAllora" of HiT Software (see [15]) because of its rich marshalling options for this task. With this software the underlying SYBASE [16] database structure can be easily mapped to a given XML schema.

Because of the relational database structure, a change in one relational item can lead to a change in a lot of XML files. Database update triggers fill the background services queue on changes in these related tables. This keeps the "flat" XML table in synchronization with the relational data.

The internal XML is stored as binary blobs in a database table linked to the datasets. On top of this a full text search engine (SYBASE EFTS) provides fast search access to the metadata. These XML blobs can be transformed into various other schemas with XSLT [14] on the fly:

- ISO 19115
- OGC WebFeatures (for WFS)
- Dublin Core (for a OAI-PMH repository)

Welcome to the DOI-Registrationsservice

1. Please choose your task

1. Citation-DOI
 2. Metadata-Update
 3. Data-DOI
 4. URL-Update

2. Provide metadata of your primary dataset:

If your Task is: '1. Citation-DOI' or '2. Metadata-Update' choose the metadata-containing XML-file:	If your Task is: '3. Data-DOI' or '4. URL-Update' just enter your DOI
<input style="width: 90%;" type="text"/> <input style="width: 10%; text-align: center;" type="button" value="Durchsuchen..."/>	<input style="width: 90%;" type="text"/>

3. Enter the URL where the primary data is available

4. Submit your request

Fig. 4. Screenshot of the web interface to the webservice for registration of datasets

- another internal thumbnail format, which is also stored as binary blob for fast access by the PANGAEA search engine "PangaVista" [18]
- STD-DOI for the DOI registration of citable datasets

For DOI registration another background service registers all new/updated datasets with the status "published" after a lead time of 30 days at the TIB by the webservice described before as *core datasets*. The lead time helps preventing inadvertent registration of datasets. During this time other data curators can look after the data and metadata and make changes which resets the lead time to 30 days again.

In PANGAEA all datasets have an unique integer ID from what the DOI is created by prefixing with a static string. The URL of the data resource is made available through the PANGAEA webserver also by the unique ID embedded into the DOI:

dataset id 80967
 ↓
 doi:10.1594/PANGAEA.80967
 ↓
 http://doi.pangaea.de/10.1594/PANGAEA.80967

After registering the core datasets the data curator can group them into a *collection dataset* (e.g. all data of a project or all data linked to a single publication) and give them a separate *citable DOI*. For that the assigned metadata gets transformed by XSLT to the STD-DOI schema from the internal XML file. Nevertheless, it is also possible to choose a single core dataset and make a citable. Due to this workflow the registering of citable PANGAEA datasets is always an upgrade of a previous core dataset (single datafile or collection) to a citable one by adding metadata at the TIB. This is done after a 30 days lead time, too.

3.4 The Infrastructure at the TIB

The webservice at the *TIB* receives XML files from the data providers, and starts the registration process:

- The DOI is registered via a java based transmission to the DOI foundation.
- For the URN registration a XML file has to be send by e-mail to the DDB.
- The metadata has to be transformed to PICA format and uploaded on a FTP server at the central library database (GBV).

To execute these different tasks, based on a single XML file, we have based the system on Apache Cocoon (see [1])

3.5 Cocoon

Cocoon is an XML publishing framework, it was founded in 1999 as an open source project under Apache Software Foundation. Cocoon offers the separation of content, style, logic and management functions in an XML content based web site (see fig. 5).

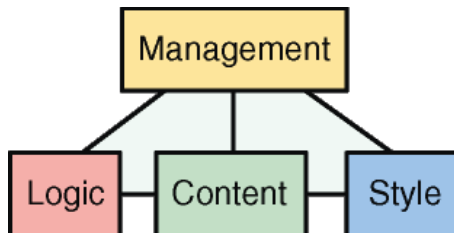


Fig. 5. Cocoon: separation of content, style, logic and management functions

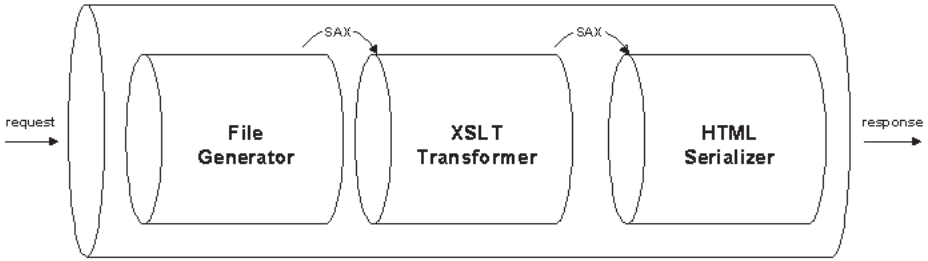


Fig. 6. Cocoon: pipeline processing

This separation allows us to easily change the parts of the architecture or the appearance of the web interface. Since it is initialised by the retrieval of a XML-file, sent to the system by the research institutes, every registration starts a XML based pipeline process (see fig. 6).

All transactions are based on XML and XSLT files.

The *eXtensible Stylesheet Language Transformation* (XSLT) is a language for transforming XML documents into other XML documents. The origins of XSL are in *Cascading Style Sheets* (CSS), where a "stylesheet" is used to add formatting to an HTML file. The syntax to use a stylesheet in XSLT is similar to the syntax in CSS.

XSLT stylesheets have a very different function than CSS stylesheets, however. CSS allows you to define the colours, backgrounds, and font-types for an HTML web page. XSLT allows you to transform an XML file into an HTML file or another text-based format.

For a complete description of XSLT we refer to [14].

3.6 Converting XML to PICA

As you can see from fig. 2 our system also includes a translation from XML-files to the PICA format. Some example XSLT commands are:

Simple tag values. Some PICA entries can easily be derived from XML entries, or combination of XML-entries: The title in PICA (4000) is a combination of the metadata attributes title, publisher, publicationPlace and creator.

```
4000 <xsl:value-of select="/resource/title"/>
    <xsl:value-of select="/resource/publisher"/>,
    <xsl:value-of select="/resource/publicationPlace"/>.
    <xsl:value-of select="/resource/creator"/>
```

Loops. For every author (instances of the attribute "creator") there is a new ordered PICA entry:

```
<xsl:for-each select="/resource/creator">
30<xsl:value-of select="position()+10"/> <xsl:value-of select="."/>
</xsl:for-each>
```

If-then structures. If the attribute `relationType` has the value "isCompiledBy", then the related DOI has to appear in the PICA category *4227 Compilation*, otherwise it appears in *4201 Footnote*.

```
<xsl:for-each select="/resource/relatedDOIs">
  <xsl:choose>
    <xsl:when test="relationType='isCompiledBy'">
      4227 <xsl:value-of select="relatedDOI"/>
    </xsl:when>
    <xsl:otherwise>
      4201 <xsl:value-of select="relationType"/>:
      <xsl:value-of select="relatedDOI"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each>
```

The XSLT code of the complete transformation can be found in the appendix of [5].

4 Status

In cooperation with

- World Data Center for Climate (WDCC) at the Max Planck Institute for Meteorology (MPIM), Hamburg
- Geoforschungszentrum Potsdam (GFZ)
- World Data Center for Marine Environmental Sciences (WDC-MARE) at the University of Bremen and the Alfred-Wegener-Institute for Polar and Marine Research (AWI), Bremerhaven

the TIB now is the world's first registration agency for primary data in the field of earth sciences.

This development will ameliorate current shortcomings in data provision and interdisciplinary use, where data sources may not be widely known and data are archived without context. It will enable citations of data in a standard manner, and also facilitate links to more specialised data schemes. The DOI system offers a proven well-developed system which is already widely deployed and enables to focus the efforts on the scientific data aspects of the project.

Authors of articles started to cite datasets using the DOI in the bibliography. One example is the following:

Lorenz, S.J., Kasang, D., Lohmann, G. (2005):
Globaler Wasserkreislauf und Klimaänderungen – eine Wechselbeziehung, In: *Warnsignal Klima: Genug Wasser für alle?* Lozán, Graßl, Hupfer, Menzel, Schönwiese (Eds.), pp. 153-158. Wissenschaftliche Auswertungen, Hamburg, Germany.

This article uses and cites:

Stendel, M., T. Smith, E. Roeckner, U. Cubasch (2004):
ECHAM4-OPYC_SRES_A2: 110 years coupled A2 run 6H values, WDCC,
 doi:10.1594/WDCC/EH4_OPYC_SRES_A2.

The web service installed at the *TIB* is fully functional and running. We have registered 50 citable and 200,000 core datasets so far (June 2005). All registration agents have used successfully the web interface to register datasets. WDC-MARE and WDC are using the webservice successfully in their production environments to automatically register DOIs without any manual user interaction. We expect an amount of approximately 1,000,000 datasets to be registered by the *TIB* until the end of 2005.

We are currently discussing cooperations to extend the registration to other disciplines like medicine and chemistry. The metadata schema is flexible enough to hold entries of these disciplines. First expressions of interest came from e.g. the *European Academy of Allergology and Clinical Immunology* (EAACI, see [19]), that wishes to register the data of its content.

The possibility of citing primary data as a unique piece of work and not only a part of a publication opens new frontiers to the publication of scientific work itself and to the work of the *TIB*. The longterm availability and accessibility of high-class data respectively content can be assured and may therefore significantly contribute to the success of "eScience".

References

1. The Apache Cocoon project <http://cocoon.apache.org/>
2. Deutsche Forschungsgesellschaft (German research foundation) homepage, <http://www.dfg.de/>
3. *International DOI foundation*, doi:10.1000/1, <http://www.doi.org/>
4. *The Handle System homepage*, <http://www.handle.net/>
5. J. Brase *Usage of metadata*, Ph.D. thesis, university of Hannover 2005
6. J. Brase *Using digital library techniques - Registration of scientific primary data*, In: "Research and advanced technology for digital libraries - LNCS 3232", Springer Verlag 2004, ISBN 3-540-23013-0
7. C. Plott, R. Ball *Mit Sicherheit zum Dokument - Die Identifizierung von Online-Publikationen*, In: B.I.T. journal **1** (2004) 11-20
8. *Project "Enhancement of Persistent Identifier Services - Comprehensive Method for unequivocal Resource Identification" homepage*, <http://www.persistent-identifizier.de/>
9. Committee on Data for Science and Technology (coData), <http://www.codata.org/>
10. The Dublin Core Metadata Initiative (DCMI), <http://dublincore.org/>
11. International Council for Science, <http://www.icsu.org/>
12. Learning Technology Standards Committee of the IEEE: *Draft Standard for Learning Objects Metadata IEEE P1484.12.1/D6.4* (12. June 2002), http://ltsc.ieee.org/doc/wg12/LOM_1484.12.1_v1_Final_Draft.pdf
13. Project webpage, <http://www.std-doi.de/>
14. W3C *XSL Transformations Version 1.0, W3C Recommendation*, <http://www.w3.org/TR/xslt>

15. HiT Software: jAllora, <http://www.hitsw.com/>
16. SYBASE Inc., <http://www.sybase.com/>
17. Diepenbroek, M; Grobe, H; Reinke, M; Schindler, U; Schlitzer, R; Sieger, R; Wefer, G (2002) *PANGAEA - an Information System for Environmental Sciences*. *Computer and Geosciences*, 28, 1201-1210, doi:10.1016/S0098-3004(02)00039-0
18. PangaVista search engine, <http://www.pangaea.de/PangaVista>
19. EAACI website, <http://www.eaaci.net/>

Incremental, Semi-automatic, Mapping-Based Integration of Heterogeneous Collections into Archaeological Digital Libraries: Megiddo Case Study

Ananth Raghavan¹, Naga Srinivas Vemuri¹, Rao Shen¹, Marcos A. Goncalves²,
Weiguo Fan¹, and Edward A. Fox¹

¹ Digital Library Research Laboratory, Virginia Tech,
Blacksburg, VA 24061
{ananthr, nvemuri, rshen, wfan, fox}@vt.edu

² Department of Computer Science,
Federal University of Minas Gerais,
Belo-Horizonte-MB Brazil 31270-901
{mgoncalv@vt.edu}

Abstract. Automation is an important issue when integrating heterogeneous collections into archaeological digital libraries. We propose an incremental approach through intermediary- and mapping-based techniques. A visual schema mapping tool within the 5S [1] framework allows semi-automatic mapping and incremental global schema enrichment. 5S also helped speed up development of a new multi-dimension browsing service. Our approach helps integrate the Megiddo [2] excavation data into a growing union archaeological DL, ETANA-DL [3].

1 Introduction

During the past several decades, Archaeology as a discipline and practice has increasingly embraced digital technologies and electronic resources. Vast quantities of heterogeneous data are generated, stored, and processed by customized monolithic information systems. But migration or export of archaeological data from one system to another is a monumental task that is aggravated by peculiar data formats and database schemas. This problem hampers interoperability, long-term preservation, and reuse. The intermediary-based approach is one way to address the interoperability problem [4]. It uses mechanisms like mediators, wrappers, agents, and ontologies. Yet, while many research projects developed semantic mediators and wrappers to address the interoperability issue, few tackled the problem of (partial) automatic production of these mediators and wrappers (through a mapping-based approach).

The mapping-based approach attempts to construct mappings between semantically related information sources. It is usually accomplished by constructing a global schema and by establishing mappings between local and global schemas. However, in archaeological digital libraries it is extremely difficult to construct a global schema that may be applied to every single excavation. Archaeological data classification depends on a number of vaguely defined qualitative characteristics, which are open to

personal interpretation. Different branches of Archaeology have special methods of classification; progress in digs and new types of excavated finds makes it impossible to foresee an ultimate global schema for the description of all excavation data [5]. Accordingly, an “incremental” approach is desired for global schema enrichment.

We explain how all these DL integration requirements can be satisfied, through automatic wrapper generation based on a visual schema mapping tool that simultaneously can improve the global schema. Further, in addition to integrating new collections, we extend access to this newly integrated data. We also enhance browsing through our multi-dimension browsing component, based on the 5S framework.

We demonstrate the integration process through a case study: integrating Megiddo [2] excavation data into a union archaeological DL, ETANA-DL [3] (see Fig. 1). Thin black lines show input whereas thick brown lines show output in the figure. First, we analyze the Megiddo excavation data management system based on a formal archaeological DL model [6]. Next we use the visual mapping service to create a wrapper (thick brown lines show wrapper generation and global schema evolution), which converts data conforming to local (Megiddo) schema to the global ETANA-DL schema. Initially, the global schema does not contain specific excavation details, but it is enriched during the mapping process. Finally, the converted data is stored in a union catalog, upon which a multi-dimension browsing service is built (see rightmost arrows). Thus, we describe the entire largely automated workflow (see Fig. 1), from integrating new collections into the union DL, to providing services to access the newly integrated data.

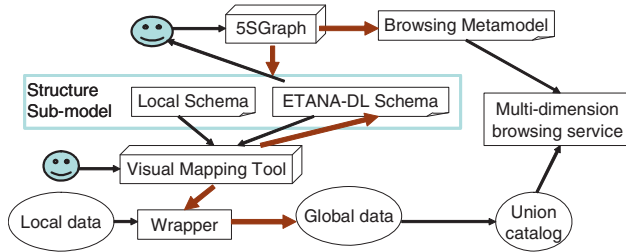


Fig. 1. Process of integrating local archaeological data into ETANA-DL

The rest of the paper is organized as follows. Section 2 gives an overview of the Megiddo collection. Section 3 describes the visual mapping service provided by ETANA-DL. Section 4 presents the componentized multi-dimension browsing service module. Section 5 describes the automation achieved through the mapping and browsing services. Conclusions and future work are summarized in Section 6.

2 Megiddo Overview

Megiddo excavation data integration is considered as a case study to demonstrate our approach to archaeological DL integration. Megiddo is widely regarded as the most

important archaeological site in Israel from Biblical times, and as one of the most significant sites for the study of the ancient Near East. The excavation data collection we received from Megiddo is stored in more than 10 database tables containing over 30000 records with 7 different types, namely wall, locus, pottery bucket, flint tool, vessel, lab item, and miscellaneous artifact. The Megiddo schema is described in a structure sub-model (see Fig. 1) within the 5S framework (Streams, Structures, Space, Scenarios, and Societies) [1]. Structures represent the way archaeological information is organized along several dimensions; it is spatially organized, temporally sequenced, and highly variable [7]. Consider site organization (see Fig. 2). The structures of sites evidence a containment relationship at every level of detail, from the broadest region of interest to the smallest aspect of an individual find, in a simple and consistent manner. Generally, specific regions are subdivided into sites, normally administered and excavated by different groups. Each site is subdivided into partitions, sub-partitions, and loci, the latter being the nucleus of the excavation. Materials or artifacts found in different loci are organized in containers for further reference and analysis. The locus is the elementary volume unit used for establishing archaeological relationships.

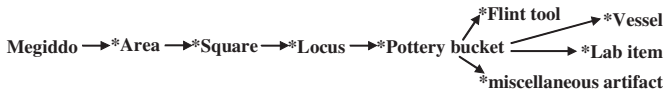


Fig. 2. Megiddo site organization

3 Visual Mapping Service

In this section we describe our visual mapping service for integrating heterogeneous data describing artifacts from various sites, especially from Megiddo into the union DL. First, we present the architecture and features of the visual mapping component. We then give a scenario-based description of mapping the Megiddo local schema into the ETANA global schema, thus integrating the data describing the various artifacts excavated from the Megiddo site into the union DL. We conclude this section by describing the results of a pilot study comparing Schema Mapper with MapForce [8], for mapping the Megiddo local schema into the global ETANA-DL schema.

3.1 Features

Schema mapping is an interesting problem which so far has been addressed from either an algorithmic point of view or from a visualization point of view. In the former case are tools based on a data driven mapping paradigm, like Clio [9] and commercial tools like MapForce [8] and BizTalk Mapper [10]. Based on the latter view, we developed Schema Mapper, the visual mapping component of ETANA-DL, to present local and global schemas using hyperbolic trees [11]. This allows for more nodes to be displayed than with linear representation techniques, and avoids the problem of scrolling. However as full node names cannot be displayed (for conserving

space), these are available as tool-tip information on individual nodes. Different colors are assigned to differentiate between root level, leaf, non-leaf, recommended, and mapped nodes (with a color legend present on the GUI in the lower right – see Fig. 3). A table that contains a list of all the mappings in the current session also is shown. Schema Mapper recommends matches (global schema nodes) to selections (local schema nodes) made by the user. These recommendations are made using name-based matching algorithms at the schema level and using rules specific to the ETANA-DL domain. The user may or may not choose to accept these recommendations. The recommendations appear in the table shown at the bottom left of the screen in Fig. 4.

The other important aspect in integration of data into the union DL is the evolution of the global ETANA-DL schema. Schema Mapper allows global schema editing: deleting nodes, renaming nodes, and adding a local schema sub-tree to the global schema. This has special value for many DLs, e.g., ArchDLs, where it is impossible to predict the final global schema because of its evolutionary nature. Schema Mapper is superior in this respect to commercial mapping tools like MapForce which lack schema editing capabilities. Further, as a global schema evolves, in order to preserve consistency in the naming of semantically similar nodes, Schema Mapper recommends appropriate name changes to global schema nodes, based on the history stored in a mappings database.

Once the local schema has been mapped to the global schema, an XSLT style sheet containing the mappings is produced by Schema Mapper. This style sheet is essentially the wrapper containing the mappings. When applied to a collection of XML files conforming to the local schema, the style sheet transforms those files to a set of global XML files, which can be harvested into the union DL. Schema Mapper also saves any changes made to the global schema, and updates the mappings database.

3.2 Scenario for Mapping Megiddo Local Schema into ETANA Global Schema

As described earlier, the Megiddo local schema consists of seven different types of artifacts. For integrating items into the union DL, we produce one style sheet of mappings per item. For the purpose of integration of the Megiddo collection into the global schema, we first consider mapping of “flint tool” and then use the knowledge of these mappings to map “vessel”.

Figures 3 and 4 show screenshots before and after the mapping of flint tool to the global schema. The left hand side screen shows the Megiddo local schema, while the right hand side shows the ETANA global schema. The ETANA global schema contains the BONE, SEED, FIGURINE, LOCUSSHEET, and POTTERY artifacts already included, apart from the top-level leaf nodes (OWNERID, OBJECTTYPE, COLLECTION, PARTITION, SUBPARTITION, LOCUS, and CONTAINER) which would be present in all artifacts (see Fig. 3).

Based on rules and name based matching strategies, we recommend mappings: OWNERID->OWNERID, OBJECTTYPE->OBJECTTYPE, COLLECTION->COLLECTION, Area->PARTITION, Square1->SUBPARTITION, Locus->LOCUS, and OriginalBucket->CONTAINER (here OWNERID, OBJECTTYPE, and COLLECTION are top-level leaf-nodes whereas Area, Square1, Locus, and OriginalBucket are all elements of the Flint tool collection).

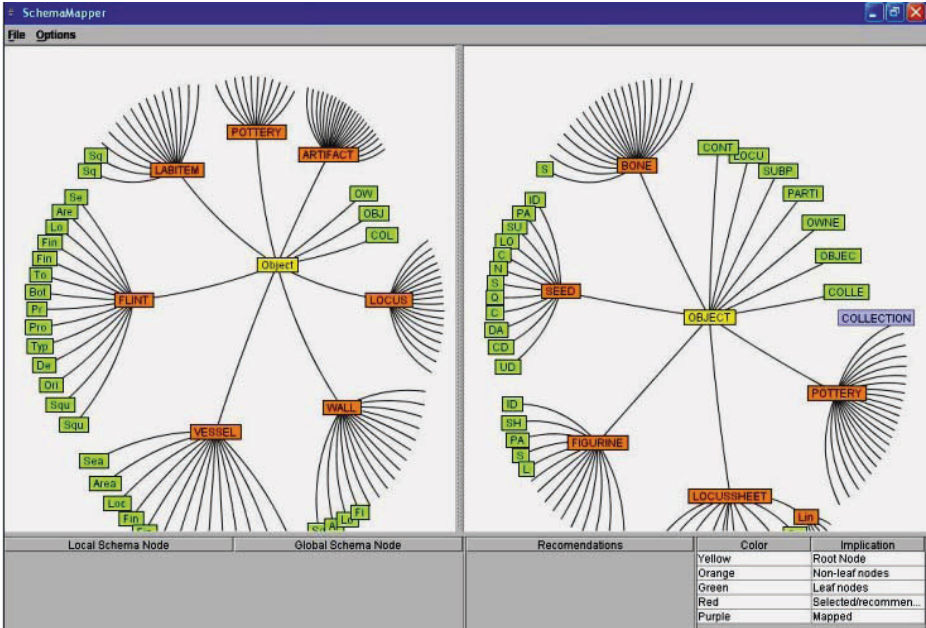


Fig. 3. Before mapping of flint tool in Megiddo to ETANA global schema

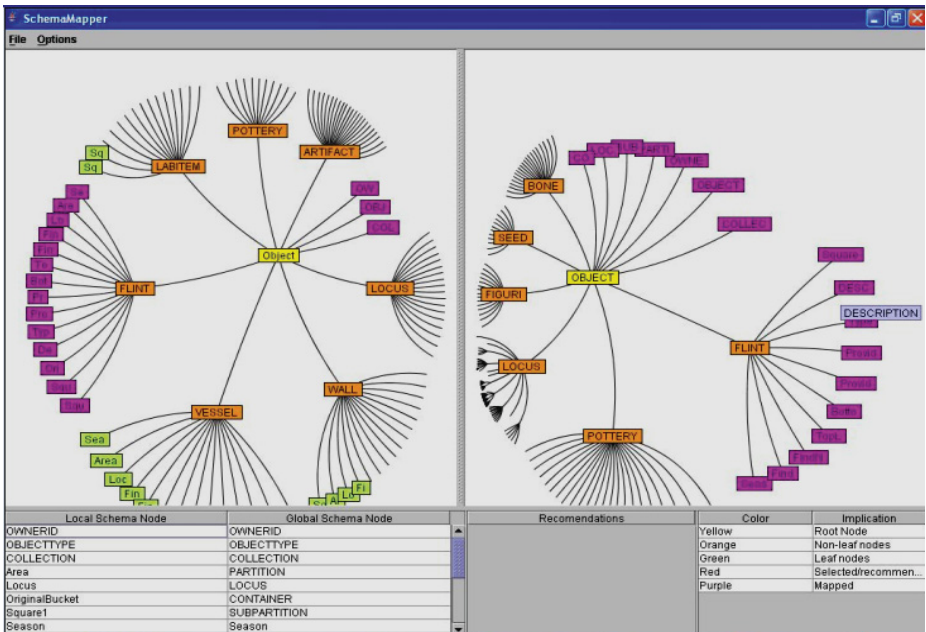


Fig. 4. After mapping of flint tool in Megiddo to ETANA global schema

The above mapping format has the local schema node on the left hand side of the arrow and the recommended global schema node on the right hand side. We map the nodes according to the recommendations, indicated by coloring these nodes purple (see Fig. 4).

As the remaining nodes in the local schema do not have corresponding global schema nodes, we add the flint tool sub-tree as a child of the OBJECT node in the global schema. This ensures that local schema elements and properties are preserved during the mapping transformation. Schema Mapper determines that some of the nodes (Area, Locus, OriginalBucket, and Square1) are already mapped, deletes these nodes from the global schema sub-tree, and automatically maps the rest with the corresponding elements in the local sub-tree (see Fig. 4). The user may decide to rename some nodes in the global schema from within this sub-tree to avoid any local connections with the name. Assume the user renames global schema node Description to DESCRIPTION. With this the mapping process is complete (see Fig. 4). Once the user decides to confirm the mappings, a style sheet is generated, the mappings are stored in the database, and the ETANA global schema is updated with the flint tool schema.

We next integrate the VESSEL artifact of Megiddo into the ETANA global schema. When we open the global schema for mapping, along with the other artifacts the Flint tool collection which was integrated in the previous step also is present. From the mapping of Flint tool we realize that mapping of a completely new artifact requires only the top-level leaf nodes to be displayed in the global schema. This however may not be the case for integration of a new collection of an existing artifact, say a new Flint tool or Pottery collection, wherein just a few additions may be required to the existing global schema sub-tree. For integration of a completely new artifact, the user may choose to view only the top-level leaf nodes in order to avoid erroneous cross mappings from schema nodes of one of the artifacts to similar schema nodes present in other artifacts. This also prevents the user from accidentally modifying a node, from say the flint tool sub-tree in the global schema, and rendering the previously generated XML files inconsistent. Also, this avoids confusing the user by presenting him with only the information he needs to see for mapping. Once again recommendations are made to enable the initial set of seven mappings; after this, the user adds the VESSEL sub-tree to the global schema.

As before, Schema Mapper finds that the Area, Locus, Square1, and Original-Bucket nodes are already mapped – and deletes them in the global sub-tree and maps the remaining nodes to corresponding local schema nodes automatically. Schema Mapper also goes through the mappings history and finds that the Description node in the flint tool sub-tree was mapped to the DESCRIPTION node in the global schema. In order to keep naming consistent, Schema Mapper recommends the user to change the name of the Description node in the VESSEL sub-tree to DESCRIPTION. This is due to the fact that both the DESCRIPTION node in the flint tool sub-branch of the global schema and the Description node in the VESSEL sub-branch of the global schema *describe* the same artifact type, but as DESCRIPTION has been selected as the global name, all Description elements in the global sub-tree should be renamed as DESCRIPTION. The recommendation, as always, is not mandatory, but if followed will help keep names consistent. When the user confirms the mappings, the database is updated, the style sheet generated, and the global schema updated with the

VESSEL schema. It is important to note that the integration of vessel artifacts into the global schema in no way changed the existing flint global entry. This leads us to the observation that, for the Megiddo Collection, modification of the global schema is simply appending a new local artifact into the global schema without changing the existing global artifacts.

The style sheets generated are applied on local XML files corresponding to particular artifacts, like vessel or flint tool; corresponding global XML files are generated. These are ready for harvest into the union DL, and available for access by services like Searching and Browsing. Detailed screenshots of the mapping process are in [12]. We integrate other artifacts in the Megiddo schema into the union DL similarly.

3.3 Comparison of Schema Mapper with MapForce for integrating Flint Tool

We undertook a pilot study comparing Schema Mapper with MapForce, regarding the amount of time required to map the flint tool collection to the ETANA global schema. The pilot tester was an expert in the domain of XML schema mapping and had used MapForce and XML Spy [13] earlier for mapping and editing schemas, respectively. Before performing the actual benchmark task, the pilot tester was given sample tasks to familiarize himself with both Schema Mapper and MapForce.

The actual benchmark task required the user to map the flint tool collection to the ETANA global schema. There were explicit guidelines on how to go about achieving this for both Schema Mapper and MapForce. The metrics for measurement were: time taken to achieve the task, number of errors, and number of times the user scrolled in MapForce vs. number of reorient actions (moving the hyperbolic tree) in Schema Mapper. The pilot tester used Schema Mapper first, and then MapForce, for performing the benchmark task. The results are shown in Table 1.

Table 1. Comparison of Schema Mapper and MapForce [8]

	Schema Mapper	MapForce
Time taken (in minutes)	4:10	9:00
Number of errors in mapping	0	0
Scrolling vs. Re-orient actions	9 (re-orient)	13 (scrolls)

From Table 1 we see that Schema Mapper significantly outperformed MapForce in the amount of time that it took for the user to perform the task. Schema Mapper also required fewer re-orient actions than scroll actions. The user made 0 errors.

Another observation was that MapForce did not help the user to edit the global schema. As a result, whenever the user had to update the global schema he had to open it in XML Spy and edit the schema by hand. For a simple collection like flint tool, the user had to switch between XML Spy and MapForce 4 times, while no switching was required for Schema Mapper (as it supports editing).

Thus the pilot study strongly indicates that for simple one-to-one mappings as are found in the Megiddo collection, Schema Mapper significantly outperforms MapForce. Further usability tests will be conducted to expand the comparison study.

4 Integrated Service in Union DLs: Multi-dimension Browsing

In this section, we consider integrated services for union DLs, specifically the multi-dimension browsing service. Later, we describe a scenario for extending this browsing service to incorporate the Megiddo collection. From the 5S point of view, it should be noted that this scenario is for the *society* of DL administrators.

4.1 Extending Integrated DL Services: Overview

A digital library is not just about the data but also about the services it provides. ETANA-DL supports various integrated DL services such as Annotation, Browsing, DL object comparison, marking DL objects, multi-dimension browsing, Recommendation, and Searching. The expressiveness of Archaeology-specific services is defined relative to the collections currently present in ETANA-DL. As new collections are integrated into the union catalog, the ETANA-DL global schema [14] is extended, and maintains the up-to-date state of the DL. In the current scenario, the ETANA-DL global schema is extended to include flint and vessel DL objects for integrating the Megiddo collection into the union catalog.

An integrated DL service can be provided for newly integrated collections in two ways. One is to re-implement the service built upon the union catalog based on the global schema. The other method is to extend the existing service to incorporate new collections. We adopt the second approach since it is more efficient.

The basic idea is to re-engineer domain specific services such that they are updated based on the global schema. So, whenever the global schema is modified, the update routine associated with each of these services automatically updates its internal data structures, if necessary. Then, newly added data collections are harvested from the union catalog into its index, and the service is made available. This approach leads to domain independent, flexible, and reusable components in union digital libraries.

We demonstrate the feasibility of the above idea by developing a prototype of a multi-dimension browsing component. In contrast to the ODL [15] browsing component, Greenstone [16]'s classification based browsing service, and Sumner et al. [17]'s browsing interface built using dynamically generated components, our focus is to modularize the browsing service and partially automate its development.

4.2 Integrated Multi-dimension Browsing Service

Digital objects in ETANA-DL are various archaeological data, e.g., figurine images, bone records, locus sheets, and site plans. They are organized by different hierarchical structures (e.g., animal bone records are organized based on: sites where they were excavated, temporal sequences, and animal names). By navigational dimension we mean a hierarchical structure used to browse digital objects. This hierarchical structure contains one or more hierarchically arranged categories that are determined by the elements of the global schema. In addition to this, a dimension of ETANA-DL can be refined based on taxonomies existing in botany and zoology, or from classification and description of artifacts by archaeologists. Our multi-dimension browsing component allows the user to browse through multiple dimensions simultaneously. In

ETANA-DL, we can browse for DL objects through the OBJECT, SPACE, and TEMPORAL SEQUENCE dimensions.

The three main sub-components of the multi-dimension browsing component are the *browsing database maintenance module* (for creation and update of the database), the *browsing engine*, and the *browsing interface module*. The last two support browsing interaction. The browsing component maintains a browsing database, i.e., a quick index to browse for DL objects quickly and efficiently.

All three modules work based on input provided by the browsing metamodel – an XML document that encodes the details of all navigational dimensions. This metamodel performs a mapping from elements of the global DL schema to the hierarchical levels in each dimension using XPath expressions. This metamodel, derived from the global schema, is generated by 5SGraph [18]. The browsing metamodel used for ETANA-DL is at <http://feathers.dlib.vt.edu/~etana/browse/etanabrowse.xml>. The *browsing database maintenance module* created the browsing database from the previously described metamodel.

4.3 Scenario: Extending the Browsing Service to Incorporate Megiddo Collection

We integrated the Megiddo collection into the ETANA-DL union catalog with the help of the XSLT style sheets generated by the Schema Mapper. The union catalog contains flint and vessel DL objects from the Megiddo site, along with other DL objects from various sites. Hence, we consider how to extend the browsing component to provide browsing services for the newly integrated Megiddo collection.

The first step is to determine the elements of the flint and vessel DL objects of the Megiddo collection that map to the hierarchical levels in each browsing dimension. The flint and vessel DL objects support browsing through dimensions SPACE and OBJECT. The elements COLLECTION, PARTITION, SUBPARTITION, LOCUS, and CONTAINER of these two DL objects define hierarchical levels in the SPACE dimension. The element OBJECTTYPE defines the OBJECT dimension. Since these elements are shared by all earlier DL objects present in the union catalog and are already used in generating space and object dimensions, the browsing metamodel need not be modified. The mapping component maps similar elements of local schemas to the same element in the global schema, eliminating the need to modify the browsing metamodel. This is true for all the artifacts in the Megiddo collection. However, in the case wherein there exist local schema elements that do not map to the same element in the global schema, the browsing metamodel will be suitably modified to include a new hierarchical level inside the specific browsing dimension.

The next step is to run the *browsing database maintenance* module to harvest the Megiddo collection from the union catalog into the browsing database. We demonstrate the extended browsing service using screen shots. Once the Megiddo collection is harvested into the browsing database, the browsing component automatically enables browsing through the Megiddo collection using SPACE and OBJECT dimensions. From Fig. 5, it can be observed that the SPACE dimension of the browsing component contains the Megiddo site (indicated by a red rectangle around it) along

with all other sites. Also, the OBJECT dimension has two new objects FLINT and VESSEL (indicated by red rectangles around them) for browsing. Fig. 6 is a screenshot of records that are displayed when we select Megiddo as the site and Flint as the Object Type (see Fig. 5) and choose to display results. Thus, we have demonstrated extending the browsing component to incorporate the new DL collection from the Megiddo site. The browsing service that incorporated the Megiddo collection is accessible at <http://feathers.dlib.vt.edu:8080/ETANA/servlet/BrowseInterface>.

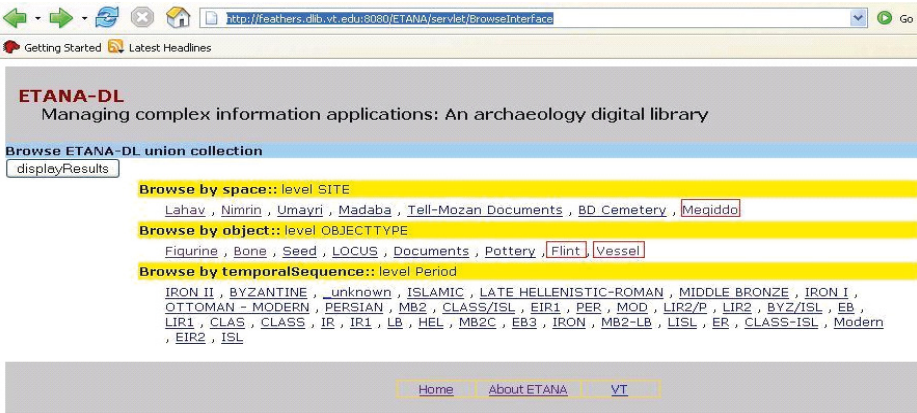


Fig. 6. Browsing component after incorporating the Megiddo Collection

The screenshot shows the ETANA-DL interface displaying a list of records. It indicates "Showing 1-10 out of 4292 records" and has navigation links for 1, 2, 3, 4, 5, 6, >, and >>. The breadcrumb trail is "You are in: Main >> Flint >> Megiddo". The records are displayed in a table format with the following columns: COLLECTION, AREA F, SQUARE, SEASON, OWNERID, LOCUS, PROVIDENCE X, PROVIDENCE Y, FIND, ORIGINAL BUCKET, and FIND NUMBER.

COLLECTION	AREA F	SQUARE	SEASON	OWNERID	LOCUS	PROVIDENCE X	PROVIDENCE Y	FIND	ORIGINAL BUCKET	FIND NUMBER
Megiddo		AY/69	2000	Israel Finkelstein	001	0.00	0.00	fink2@post.tau.ac.il	00/F/001/PT001	001
Megiddo		AY/69	2000	Israel Finkelstein	001	0.00	0.00	fink2@post.tau.ac.il	00/F/001/PT001	002
Megiddo		AY/69	2000	Israel Finkelstein	001	0.00	0.00	fink2@post.tau.ac.il	00/F/001/PT001	003
Megiddo		AY/69	2000	Israel Finkelstein	002	0.00	0.00	fink2@post.tau.ac.il	00/F/002/PT001	001
Megiddo		AY/71	2000	Israel Finkelstein	006	0.00	0.00	fink2@post.tau.ac.il	00/F/006/PT001	001

Fig. 7. Records from the flint tool collection of ETANA-DL

5 Savings from Automation

Automation is achieved in the wrapper code generation (style sheet generation), global schema enrichment, and through the extensible multi-dimension browsing component. We measure this automation in terms of lines of code (LOC) that we saved using Schema Mapper and the Multi-dimension Browsing Component as compared with the former hard coded approach. Table 2 shows the LOC required in each case to add a collection into the Union-DL. With Schema Mapper, there is no need to write new code when we integrate a new collection, such as Nimrin or Madaba.

Table 2. LOC Comparison between Schema Mapper and Hard Coded Approach

	Hard-Coded Wrapper for Nimrin	Hard-Coded Wrapper for Madaba	Schema Mapper (for Madaba and Nimrin)
Additional LOC required	1605	1770	0

Regarding the Browsing Service, comparison is more difficult, since we earlier hard coded a (less flexible) browsing component. Clearly, we avoid between 50-75 lines of additional code that would be required for a change the old way. But the main advantage of the new component is that it can be plugged into any digital library (not necessarily an Archaeological DL) and be driven by a formal (5S) browsing model.

6 Conclusions and Future Work

Through the integration of artifact data from the Megiddo excavation site into the union DL, using the visual mapping component, we have successfully demonstrated a semi-automatic tool which generates a wrapper (XSLT style sheet) using the schema mapping approach. Through the mapping component we also achieve the goal of incrementally enriching the global schema with local schema information from new excavations. Further, through the multi-dimension browsing component we complete the automation of the workflow for adding a site, from integrating data into the union DL, to extending the browsing service to access all integrated data.

Initial pilot studies of the mapping and browsing components have been positive. We plan extensive usability tests. Also, complex (one to many and many to one) mappings will be explored and the mapping component will be enhanced accordingly. Future work will include enriching mapping recommendations through statistical data analysis, and enhancing the functionality and portability of the browsing component.

Acknowledgements. This work is funded in part by the National Science Foundation (ITR-0325579). We also thank Doug Gorton and members of the DLRL for their support. Marcos Goncalves had an AOL fellowship and has support from CNPq.

References

1. Gonçalves, M.A., Fox, E.A., Watson, L.T., and Kipp, N.A. Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. *ACM Transactions on Information Systems (TOIS)*, 22 (2): 270-312, 2004.
2. Megiddo, 2005. <http://www.tau.ac.il/humanities/archaeology/megiddo/index.html>
3. U. Ravindranathan. Prototyping Digital Libraries Handling Heterogeneous Data Sources - An ETANA-DL Case Study. Masters Thesis. Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, April, 2004, <http://scholar.lib.vt.edu/theses/available/etd-04262004-153555/>
4. Park, J. and Ram, S. Information systems interoperability: What lies beneath? *ACM Transactions on Information Systems (TOIS)*, 22 (4): 595-632, 2004.
5. Finkelstein, S., Ussishkin, D. and Halpern, B. Monograph Series of the Institute of Archaeology, Tel Aviv University, 2000.
6. Shen, R. Apply the 5S Framework in Integrating Digital Libraries. Dissertation Proposal, Virginia Tech, 2004
7. Schloen, J.D. Archaeological Data Models and Web Publication Using XML. *Computers and the Humanities*, 35 (2): 123-152, 2001
8. Altova. *Mapforce*, 2005. http://www.altova.com/products_mapforce.html
9. L. L. Yan, R. J. Miller, L. M. Haas, and R. Fagin. Data-driven understanding and refinement of schema mappings. In *Proc. SIGMOD Conference*, 2001
10. Microsoft. *BizTalk Mapper*, 2005 <http://www.samspublishing.com/articles/article.asp?p=26551&seqNum=5>
11. Lamping, J. and Rao, R., Laying Out and Visualizing Large Trees Using a Hyperbolic Space. In *Proc. ACM Symp. on User Interface Software and Technology*, 1994, 13-14
12. Schema Mapper Screenshots, 2005
<http://feathers.dlib.vt.edu/~etana/Papers/Screenshot.doc>
13. Altova, XML Spy, 2005. http://www.altova.com/products_ide.html
14. ETANA-DL Global XSD, 2005. <http://feathers.dlib.vt.edu/~etana/etana1.1.xsd>
15. Suleman, H. Open Digital Libraries, Ph.D. Dissertation, Dept. Comp. Sci., Virginia Tech, 2002, <http://scholar.lib.vt.edu/theses/available/etd-11222002-155624>
16. Bowman, C.M., Danzig, P.B., Hardy, D.R., Manber, U., and Schwartz, M.F. The Harvest information discovery and access system. *Computer Networks and ISDN Systems*, 28 (1-2): 119 - 125.
17. Sumner, T., Bhushan, S., Ahmad, F., and Gu, Q. Designing a language for creating conceptual browsing interfaces for digital libraries. In *Proc. JCDL*, 2003. 258-260.
18. Zhu, Q. 5SGraph: A Modeling Tool for Digital Libraries, Masters Thesis. Dept. Comp. Sci., Virginia Tech, 2002, <http://scholar.lib.vt.edu/theses/available/etd-11272002-210531>

Integrating Diverse Research in a Digital Library Focused on a Single Author

Neal Audenaert¹, Richard Furuta¹, Eduardo Urbina², Jie Deng¹, Carlos Monroy¹,
Rosy Sáenz², and Doris Careaga²

¹ TEES Center for the Study of Digital Libraries,
& Department of Computer Science,
Texas A&M University,
College Station, TX, 77843
cervantes@csdl.tamu.edu

² TEES Center for the Study of Digital Libraries,
& Hispanic Studies Department,
Texas A&M University,
College Station, TX, 77843
cervantes@csdl.tamu.edu

Abstract. The works of a significant author are accompanied by a variety of artifacts ranging from the scholarly to the popular. In order to better support the needs of the scholarly community, digital libraries focused on the life and works of a particular author must be designed to assemble, integrate, and present the full scope of these artifacts. Drawing from our experiences with the Cervantes Project, we describe five intersecting domains that are common to similarly focused humanities research projects. Integrating the tools needed and the artifacts produced by each of these domains enables digital libraries to provide unique connections between diverse research communities.

1 Introduction

Like many other projects dedicated to a single author, work at the Cervantes Project initially focused on maintaining a comprehensive bibliography of scholarly research and providing access to the works of Miguel de Cervantes Saavedra (1547-1616). We made his works available via the Cervantes Project web site in a variety of editions in several versions (facsimile, old-spelling, modernized, English) along with the interfaces, hypertext links, and search engines to facilitate their use at multiple levels [13]. We have developed an electronic *variorum* edition (EVE) and are populating it with a text collection previously unavailable to the Cervantes scholar [5]. While work is ongoing in these areas, we are expanding the scope of our project to include resources to support historical and biographical research, investigations into the impact of Cervantes' cultural environment on his writings, and studies of popular and scholarly artifacts based on or inspired by Cervantes.

As we expand our focus, we are able to better categorize the breadth of scholarly research activities centered on a single author and the types of resources that digital

libraries need to provide to support those activities. The humanities research involved in this project is characterized by numerous researchers conducting detailed studies of highly focused, inter-disciplinary research questions. For example, some researchers are interested in illustrated editions of *Don Quixote*, others on historical and biographical records and yet others on Cervantes' impact on music. This in turn raises the question of how to provide tight interlinkages among the resources developed by various researchers without requiring large amounts of follow-on customization—an unaffordably labor-intensive effort. While we have focused on the life and works of Cervantes, the practices we have observed in this project typify many humanities research endeavors. The works of a significant author or, more generally, a single artist (author, painter, poet, etc.) are accompanied by a variety of artifacts ranging from the scholarly to the popular that are distributed in time from before the author was born to the present. Consequently, to fully support this research, digital libraries focused on the life and works of a particular author cannot be content merely to present the author's works. Instead, they need to be designed to assemble, integrate, and present the full scope of these artifacts in a manner that facilitates the sophisticated interpretative strategies required for scholarly research [7].

In this paper we discuss our growing understanding of the scope of humanities research practices, drawing on our current work to inform and illustrate our findings. We also describe our current efforts to employ the narrative and thematic structure of Cervantes' most notable work, *Don Quixote (DQ)*, to provide an integrative motif that lends a natural unifying structure to the diverse artifacts in our archive.

2 The Cervantes Project

The Cervantes Project is developing a suite of tools that can be grouped into six major sub-projects: bibliographic information, textual analysis, historical research, music, *ex libris* (bookplates), and textual iconography. Figure 1 provides an overview of how the artifacts from these sub-projects fit into a timeline of Cervantes' life and writings. While each of these sub-projects is individually interesting, the primary contribution of our work stems from the fact that by integrating them, we are able to bring diverse and previously disconnected research together in a way that enhances the value of each sub-project.

2.1 Bibliographies

Bibliographies are perhaps the primary artifact for developing a scholarly “corporate” memory in numerous areas. Since 1996, we have maintained a comprehensive bibliography of scholarly publications pertaining to Cervantes' life and work [14]. This is published periodically in both an online and print form. We have implemented a flexible, database driven tool to manage large-scale, annotated bibliographies. This tool supports the taxonomies and multiple editors required to maintain a bibliography with thousands of records.

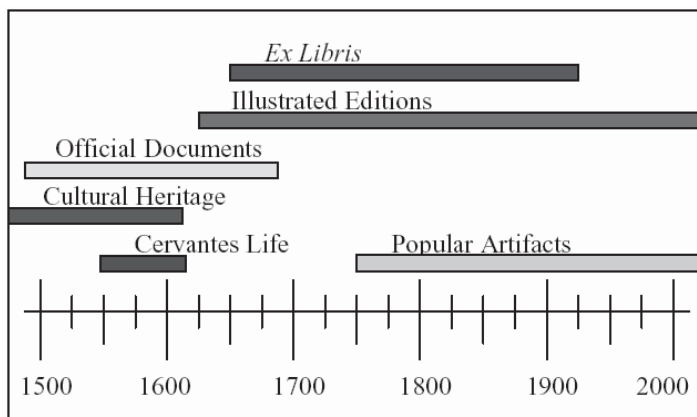


Fig. 1. A lifeline depicting the collections included in the Cervantes Project as they relate to his life

2.2 Textual Analysis

Early in the project we also worked to modernize the primary resources used in traditional textual analysis—the edition and associated commentary. We developed an electronic *variorum* edition (EVE), a reader’s interface (VERI), and a variant editor (MVED) [5], and are in the process of populating a collection of a scope previously unavailable to the Cervantes scholar. Currently, digital images of ten copies of the 1605 *princeps* edition, one copy of the 1615 *princeps* edition and one copy of the three-volume Bowle edition (1781) are available online. Nine copies of the 1605 *princeps* edition have corresponding full text transcriptions linked to the page images and work is currently in progress to add eight copies of the 1615 *princeps* edition (with transcriptions) and about twenty-five copies of later editions. Although the presentation of the texts is centered on the printed version, we are taking advantage of the electronic medium to reshape the form of the books; for example, bringing into proximity related portions of the multi-volume Bowle edition previously published as separate books. We have also employed timelines to visualize the variants and verify the transcriptions [10].

2.3 Historical Research

To better support access to historical and biographical data, we are developing tools to identify dates, people, and places in a collection of approximately 1600 official documents pertaining to Cervantes and his family [11]. Once identified, this information is used to automatically generate dense navigational links and to support collection visualization. Biographies and chronologies of Cervantes and his family will be integrated with this collection, connecting events, people, facts, and places to primary source materials.

2.4 Music

We have recently begun work with a digital collection that explores the intersection of music and Cervantes. It will include detailed information about the instruments Cervantes mentions (images, audio, descriptions, etc.). It will also organize songs, dances, and other musical works inspired by *DQ* around the narrative and thematic elements of the text. This collection will be used to assist scholars investigating Cervantes' awareness of the musical trends of his day, the influence of that music on his writings, and the subsequent interpretation of Cervantes' works by various musicians. Specifically, it will provide them with access to playable scores from musical works written about Cervantes, discussions of themes found in the music of Cervantes' day and how those themes are reflected in his writings, historical notes, bibliographic information, and audio.

2.5 Ex Libris

We have assembled a digital collection of more than 1300 *ex libris* (bookplates) inspired by or based on *DQ*. These *ex libris* are taken from collection loaned to the Cervantes Project by Doctor Gian Carlo Torre. His collection is one of the most important in the world. Now, in its digital form, we are able to offer, for the first time, easy access and reference to a modern artistic corpus that expands the iconography and visual reading of *DQ* while providing an insight into the iconic transformation of the text in the 20th century.

2.6 Textual Iconography

Finally, we have assembled an extensive collection of illustrations from various editions of *DQ*. The Cervantes Project, in collaboration with the Cushing Memorial Library of Texas A&M University, has acquired nearly 400 copies of illustrated editions of *DQ* published between 1620 and 2004. Currently, we have digitized more than 4000 images from 74 of the most significant of these editions. These illustrations are encoded with detailed metadata information pertaining to both their artistic features (e.g. artist, date, size, style, texture) and their literary context (e.g. thematic and narrative elements, characters). The iconography collection facilitates investigations of artists' interpretation of *DQ* throughout history, the cultural, political, and ethical factors that have influenced these interpretations, and the individual artists' unique analysis, techniques, and stylistic flavor.

We plan to enhance this collection will collation tools to facilitate access to these illustrations. These include support for book-based collations that allow the illustrations to be placed in their original physical, narrative or thematic context, natural collations that group illustrations by author, style, size, etc., and custom collations created or tailored by individuals

2.7 Narrative and Thematic Structure as an Integrative Motif

These six lines of research come from distinct scholarly traditions and offer diverse perspectives. They are united, however, in their common goal to better understand

Cervantes' writings and the impact of these writings on the human experience. Historically, the unity of this research has often been lost to the pragmatic difficulties of identifying and accessing the relevant research across the boundaries of academic disciplines. The digital resources we are developing help bridge these boundaries, bringing research results together in a single digital library structured by the narrative and thematic elements of *DQ*.

To bridge these boundaries, it is not enough to simply publish archives of artifacts "on-line" [2]. To adequately support the humanities researcher, the connections between these sub-projects need to be identified and the collections enhanced with tight interlinkages. One key challenge we face as we develop these resources is supporting the integration of these resources without requiring hand coding—a task that would itself be a significant undertaking.

Our approach focuses on identifying and tagging the narrative and thematic elements in the texts themselves, rather than relying on the more traditional positional ties to printed volumes (for example, page and line numbers). In support of this approach, we have developed taxonomies and controlled vocabularies and are encoding *DQ* and Cervantes' other works using TEI standards [6]. The text is naturally divided into chapters that we further sub-divide to indicate narrative units within those chapters. Each of these major and minor narrative divisions is given a short description and tagged with taxonomic categories, principal themes and dominant moods. Examples of taxonomic categories include chapter, place, direction, episode, adventure, action, and character. Examples of themes include madness, love, food, play, enchantment, knighthood, chivalry, justice, freedom, and violence. Examples of moods include parodic, burlesque, ironic, and satiric.

As this is done, other resources can be assigned metadata that identifies their relationships with the structure of the text. This approach allows us to automatically integrate new artifacts as they are added to the collections, providing links from the new artifact to previously existing resources, and from existing data to the new artifact.

3 Characterizing Scholarly Research

Our experiences with this project have encouraged us to carefully reexamine the practices that characterize scholarly research centered on a single author and the types of resources that digital libraries need to provide to adequately support those practices. The work we are currently conducting is clearly not limited to the study of the works produced by Cervantes and commentary on those works. It extends to his life in general, historical documents of his time, the contemporary cultural context in which Cervantes and his works are embedded, and the scholarly and popular artifacts that are based on or inspired by his works. Figure 2 shows a Venn diagram illustrating the relationships between these research domains. We believe that this view is generally descriptive of humanities research efforts centered on the life of a single author or, more generally, a single artist (author, painter, poet, etc.). In this section we describe each of these areas, drawing on our experiences with the Cervantes Project to illustrate the key issues involved.

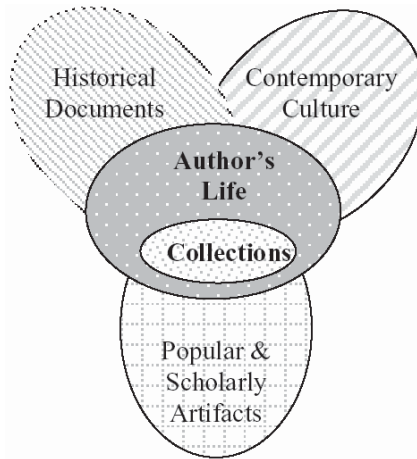


Fig. 2. A Venn diagram illustrating the relationships between the five domains of humanities research focused on a single author: collections of the author's works, the author's life, historical documents, contemporary culture and derivative popular and scholarly artifacts

3.1 Collections

Assembling and presenting a collection of an author's work is central to any project purporting to be a digital library that supports research about a single author. McGann has clearly demonstrated the potential advantages digital resources have for studying texts over traditional print based approaches and has presented a vision for digitally based humanities research [8]. Numerous projects have taken up his vision, including the Rossetti Archive [16], the Canterbury Tales [12], and the Picasso Project [15] among others. While this is the smallest and most well defined domain in a humanities research project, it is by no means simple. Tools are needed to support textual criticism of older texts. Difficult decisions must be made: How should the texts be presented visually? Should the original structure of the book be retained? Which texts should be given primacy when revisions have been made by the author (e.g., should the last version of the text published during the author's lifetime be given primacy, or are all editions of equal value)? Are individual copies important in their own right or only as they serve as an exemplar of an edition? Is it sufficient to provide a textual representation of the original text or are facsimiles of the original pages needed [4]?

Within the Cervantes Project we have found that these decisions need to be carefully evaluated not just on a project-by-project basis, but also within each project on a more fine-grained basis focused on the nature of specific research questions. For the majority of Cervantes' works (e.g. *Novelas ejemplares*, *La Galatea*, etc.) we have been content to provide textual transcriptions of the work in a variety of editions (old-spelling, modernized, English translations). This reflects their relatively decreased prominence in the corpus of Cervantes' work. On the other hand, we have paid considerably more attention to these questions with respect to developing resources for *DQ*. Our facsimile and critical editions form the most significant part of our textual archive of Cervantes' work. For the most part, we have retained the original structure

of the book in presenting these editions, but in the Bowle edition we recast that structure in order to align the commentary section in the third volume with the corresponding pages in the first two volumes. Breaking the structure of the book, for example to bring two different editions into proximity for comparison purposes or to integrate images and commentary that were not originally part of the text, is likely to become a more prominent feature of our approach. Depending on features of a particular edition, we have made different decisions about the primacy of copies. For the *princeps* edition, each extant copy is different and hence important. For the Bowle edition, one copy is sufficient to serve as an exemplar of the edition as an abstract entity. In deciding between textual vs. facsimile representation of the text, our answer has again depended on specific research questions. For the task of textual criticism, both full text transcripts and images of the original pages are required to adequately support the analytical tasks involved. For many of the later editions in which variants between the text of the edition and the “original” are less important, we have been content to provide only a facsimile edition. This reflects scholarly interest in the form of the publication as it changed from edition to edition rather than in the less significant changes in the content of the text.

3.2 The Author’s Life

The works of an author are encompassed within the broader context of his life and nearly all projects focused on a single author make an effort to present at least a minimal amount of information about the author’s life. Access to detailed information about an author’s life is important to scholarly work for two broad purposes. First, a biographical understanding of the author may lead to insights into the motivations for and perspectives influencing his writings. Second, learning more about the author’s life is interesting in its own right. It is the subject both of scholarly research and popular interest (as indicated by the existence of television stations such as the Biography channel). Integrating biographical information into a scholarly archive enhances reader’ understanding promoting a dialog between the focused studies of the author’s works and the biographical information.

Traditional scholarly tools for studying an individual’s life are well recognized and include biographies and chronologies. These are often supported by paintings, maps, and other materials. These are secondary resources, however, and are limited in their ability to support the discovery of new information. Similar to the way in which McGann critiqued the copy-text approach to textual criticism for selecting an authoritative central text and describing its differences with other, marginalized texts [9], a biography can be seen as establishing a centered, “authoritative” narrative of the author’s life, relegating other possibly significant elements to the margins. While this may be useful for many purposes, one of the key advantages of developing a digital archive is to assist scholars (and the public) in finding their own way through the available information developing and utilizing their own unique interpretive perspectives [3]. Set in the context of a digital library, research into the life of the author can benefit not only from the explicitly biographical information commonly provided, but also from the other information in the archives supporting historical research and information about the contemporary cultural context of the author. In this environment, biographies can serve as annotated trails and their authors as trailblazers in

ways similar to those envisioned by Bush [1]. This attitude reflects a different perspective on this domain of scholarly research and affords new possibilities for supporting research without constraining scholars to the single interpretation presented by the biographer.

The Cervantes Project has long provided access to the traditional tools of biographical research (limited by ever-present copyright restrictions). We are beginning to augment this approach with the extensive collection of primary source materials drawn from the official documents collected by Sliwa [11] as well as the results of research into Cervantes' awareness of contemporary musical themes. While these two collections are distinct sub-projects, we are working to integrate them into the overall structure of our archive in order to enable researchers and the public to gain a better understanding of Cervantes' life.

3.3 Historical Documents

Both the author's life and works are embedded in, though not encompassed by, a broader historical context. Providing resources to inform and clarify scholars' understanding of this historical context enhances their understanding of, and engagement with, the author. From a different perspective, exploring the interactions between a particular artist and the historical context in which he and his work are embedded can help scholars better understand significant facets about that historical context. To support research efforts in this domain, digital libraries need to provide access to relevant historical documents and research results. This opens the challenging question of what historical resources ought to be included in a library. While many resources may be potentially helpful, including too much information could detract from the library by unnecessarily diverting it from its ideological focus. One alternative would be to develop supporting collections. For example, one could imagine a digital library focused on collecting historical resources for Spain and the western Mediterranean between the 15th and 17th centuries that could support the Cervantes Project as well as a number of other similar humanities archives.

From a more practical standpoint, this problem is often much simpler than the above considerations. Few projects focused on the life and work of a single author have access to collections of the scope of, for example, the Perseus Project's London collection [3] or the resources to digitize it. Thus, the primary considerations become identifying appropriately sized collections that are available and closely related to the author's life and works. Once a suitable collection is identified, further work will be needed to effectively determine how this collection should be incorporated into the body of artifacts maintained in the digital library. When compared to artifacts collected with the narrower objective of supporting research about the author's life and work, this domain is much more open-ended. The process of determining what artifacts and collections will be most beneficial is heavily dependent on the idiosyncratic needs of particular research communities and on the current availability of materials.

Within the Cervantes Project, we have provided access to the collection of 1600 official documents pertaining to the life of Cervantes and his family as mentioned earlier. The availability and clear relevance of this collection provide an obvious answer to the question of whether or not it merited incorporation into our digital library. Our efforts at integrating it into the larger context of our archive are currently focused on

developing tools to support named entity identification and the identification of other potentially relevant information (the price of eggs in Madrid). As this is done, we will be able to integrate these primary source records better with the biographical information already present in our archive. Questions about how to integrate this information with the texts of Cervantes' remain fruitful areas for future investigation.

3.4 Contemporary Culture

A closely related line of research investigates the nature of the author's engagement with the contemporary cultural context in which he and his works are embedded. To what extent was an author aware of a given cultural theme and how did this awareness affect his writings? What impact did the author have on his contemporaries? How does an author's work inform other questions about his culture? Incorporating cultural artifacts and resources in a single author digital library also assists readers by providing access to information about the author's embedding culture that may be outside the scope of the reader's expertise. Like collections of historical resources, the scope of work that could be incorporated to facilitate an awareness of the author's embedding culture is open ended. Examples of cultural elements that could be integrated include literary, poetic, culinary, societal, dramatic, and musical culture. Again, which resources warrant inclusion is heavily dependent on the characteristics of the author's life and work as well as the pressing issues currently being discussed in the scholarly community.

Our work in developing a music archive related to Cervantes has a significant component that deals with the cultural elements of Cervantes' day. We are incorporating information that connects the texts of Cervantes to information about the musical instruments they mention. Since many of these instruments are not familiar to modern readers or have since significantly altered in form, the connections will improve the depth of our understanding of the texts. From a more scholarly perspective, this collection will describe how major themes and topics found in the music of Cervantes day may be reflected in his writings. This facilitates the exploration of Cervantes' awareness of this aspect of his culture and offers possible insights into his intentions.

3.5 Popular and Scholarly Artifacts

Conceptually, assembling, analyzing, and presenting the body of popular and scholarly artifacts based on or inspired by an author's work is the largest domain for supporting scholarly research. These derivative artifacts lie on a continuum from the results of scholarly research at one end to popular trinkets at the other. At the scholarly end, these artifacts include critical editions, diagrams, and scholarly writings about all aspects of the author's life and works. One clear example of research focused on these artifacts is annotated bibliographies. At the other end of the spectrum are popular artifacts, including souvenirs, trading cards, toys, wrappers, and posters. Between these two extremes lies a tremendously diverse set of artifacts that are unified by their common derivative status.

A few examples from our work within the Cervantes Project will help illustrate some of the possibilities of this class of scholarly research as well as the type of support that a digital library can provide. Our music collection, in addition to providing

resources about music during Cervantes' life, also places a major emphasis on collecting and analyzing the musical compositions that have been based on his works. This resource places the research generated by music scholars into the context of the works on which that music is based. It also provides a unique perspective on the texts of Cervantes by bringing artifacts from this unique interpretive media into proximity to the texts they interpret. Our two collections that focus on artistic elements that been added to the text also fall into this area. The textual iconography project overlaps with research about the text, but since the illustrated editions are all artistic interpretations of *DQ* it is more strongly identified with this area of research. (Contrast this with the illustrated novels of Dickens and Thackeray in which the illustrations were a part of the original published work). The artifacts of the *ex libris* collection are clearly an example of artifacts worthy of scholarly inquiry that are derived from the author's work.

These three examples serve to indicate the breadth and open-ended nature of the scholarly research involved in this domain and of the derivative artifacts that could productively serve in a digital library. This area is the most characteristic of research practice.

4 Conclusions

While our work at the Cervantes Project by no means exhausts the scope of scholarly research that may be motivated by a single author's work, it does begin to suggest the breadth and interdisciplinary nature of that research. In carefully reexamining scholarly practices, we have identified five intersecting domains of that are common across similarly focused humanities research projects: the textual analysis, biographical studies, historical context, contemporary cultural context, and derivative popular and scholarly artifacts. Work in each of these areas is characterized by detailed, thorough investigations with a relatively narrow focus, the engagement of a broad range of humanities disciplines (for example, art, music, publishing, literature, sociology, etc.), large bodies of secondary work developed over long time spans (requiring bibliographies and other tertiary scholarly works), and the need to integrate primary and secondary materials. By integrating the tools needed for and the artifacts produced by each of these five major research domains, a digital library focused on scholarly research pertaining to a single individual is able to support unique connections between diverse and otherwise disconnected research communities.

References

1. Bush, V., "As We May Think", *Atlantic Monthly* (July 1945). 101-108.
2. Crane, G., et al. "Drudgery and Deep Thought," In *Communications of the ACM*, Vol. 44, Issue 5. ACM Press, New York (May 2001). 35-40.
3. Crane, G., Clifford E. Wulfman, and David A. Smith, "Building a Hypertextual Digital Library in the Humanities: A Case Study on London", In *Joint Conference on Digital Libraries, JCDL01*, (Roanoke, Virginia, June 2001). ACM Press, New York (2001). 426-434.
4. Flanders, J. "Trusting the Electronic Edition." In *Computers and the Humanities*, Vol. 31. Kluwer, The Netherlands. (1998). 301-310.

5. Furuta, R., et al. "The Cervantes Project: Steps to a Customizable and Interlinked On-Line Electronic Variorum Edition Supporting Scholarship." In European Conference on Digital Libraries, ECDL2001. (Darmstadt, Germany, September 2001). Berlin: First Springer, 2001. 71-82.
6. Kochumman, R. et al. "Tools for a new Generation of Scholarly Edition Unified by a TEI-based Interchange Format." In Joint Conference on Digital Libraries, JCDL04, (Tuscon, Arizona, June 2004). ACM Press, New York (2004). 368-369.
7. Lynch, C., "Digital Collections, Digital Libraries and the Digitization of Cultural Heritage Information," *First Monday*, Vol. 7, Issue 5. (May 6, 2002).
8. McGann, J. "The Rossetti Archive and Image-Based Electronic Editing." In Finneran, R. J. (ed.): *The Literary Text in the Digital Age*. University of Michigan, Ann Arbor, MI (1996) 145-183
9. McGann, J. "The Rationale of Hypertext." In Sutherland, K. (ed.): *Electronic Text: Investigations in Method and Theory*. Oxford UP, New York, (1997) 19-46.
10. Monroy, C., et al. "Interactive Timeline Viewer (ItLv): A Tool to Visualize Variants Among Documents," In *Visual Interfaces to Digital Libraries, Lecture Notes in Computer Science*, Vol. 2539. Springer-Verlag, Berlin Heidelberg New York (2002) 33-49.
11. Sliwa, K.. *Documentos Cervantinos: Nueva recopilación; lista e índices*. New York: Peter Lang, 2000.
12. "The Canterbury Tales Project." De Montfort University, Leicester, England. <http://www.cta.dmu.ac.uk/projects/ctp/index.html>. Accessed on Feb 7, 2005.
13. "The Cervantes Project." Center for the Study of Digital Libraries, Texas A&M University. <http://csdl.tamu.edu/cervantes>. Accessed on Feb 7, 2005.
14. "The Cervantes International Bibliography Online (CIBO)." Center for the Study of Digital Libraries, Texas A&M University. <http://csdl.tamu.edu/cervantes>. Accessed on Feb 7, 2005.
15. "The Picasso Project", Hispanic Studies Department, Texas A&M University.. <http://www.tamu.edu/mocl/picasso/>. Accessed on Feb 7, 2005.
16. "The Rossetti Archive." The Institute for Advanced Technologies in the Humanities, University of Virginia. <http://www.rossettiarchive.org/>. Accessed on Feb 7, 2005.

A Fluid Interface for Personal Digital Libraries

Lance E. Good, Ashok C. Popat, William C. Janssen, and Eric A. Bier

Palo Alto Research Center,
3333 Coyote Hill Road, Palo Alto, California 94304
{good,popat,janssen,bier}@parc.com

Abstract. An advanced interface is presented for fluid interaction in a personal digital library system. The system employs a zoomable planar representation of a collection using hybrid continuous/quantum treemap visualizations to facilitate navigation while minimizing cognitive load. The system is particularly well suited to user tasks which, in the physical world, are normally carried out by laying out a set of related documents on a physical desk — namely, those tasks that require frequent and rapid transfer of attention from one document in the collection to another. Discussed are the design and implementation of the system as well as its relationship to previous work.

1 Introduction

The persistence of paper as a preferred medium for document interaction, in the face of distinct and growing advantages of electronic representations, has in recent years been well-studied [1]. Perhaps nowhere are the advantages of paper over extant electronic alternatives so strong as in the case where the reader wishes to compare a set of passages appearing in several documents, or to cycle attention among passages from several documents in rapid succession. Such patterns of interaction arise naturally during a variety of knowledge tasks, including the preparation of lectures, the reviewing of papers, the analysis of intelligence briefings, and the writing of a report, among many others. We shall refer to such patterns as “reading from multiple sources,” with “reading” to be taken in its general sense to include such activities as annotation, extraction, summarization, and the like.

We consider reading from multiple sources in the context of a personal digital library. By the qualifier “personal” here we mean that the user (1) already has the right to use all of the data objects in the library, and (2) already has local possession of those objects. An example of such a personal digital library is the result of scanning one’s office filing cabinet onto a local hard drive.

In recent years, the importance of effective visualization tools and visual interfaces to digital library collections has been recognized [2,3,4]. In the case of a *personal* digital library, an opportunity exists to blur the traditional separation between searching for materials and using them. The system presented in this paper pursues this goal.

In the physical world, reading from multiple sources often involves spreading out a set of documents on a large desk or table. During the course of the activity

the reader is at various times absorbed in the close study of one document or another, while at other times, be it in transition or in respite, the reader broadens his or her focus momentarily to regard the workspace as a whole. Fluid transitioning among these modes of attention is achieved, and the continuity of orientation maintained, by virtue of the persistence of the layout of the physical documents in a single visual field, and through the ability of the reader to adjust his or her center and field of attention rapidly and at will.

The system described in this paper is an attempt to simulate and, at least with respect to navigation, surpass in the electronic realm the affordances of reading from multiple paper sources. A uniform and consistent interaction interface is provided to every document in a zoomable virtual workspace, irrespective of its native electronic “format” or “type.” This universality is enabled by a fast, transparent, and automatic means of converting arbitrary documents of interest into a common intermediate working representation. To support the navigational cues that get built up in spatial memory during the course of the activity; advanced visualizations known as *treemaps* are used in the system’s interface. Provision is made for navigation by interactive incremental textual search.

1.1 Universality of Representation and Consistency of Interface

Phelps and Wilensky [5] note that “picking a format is often tantamount to choosing a browser/editor/viewer with its packaged bundle of features and limitations.” The diversity of applications required to interact with different document types leads to a proliferation of user interfaces, placing the additional burden on the user of having to learn and remember their proper operation.

Personal digital library systems can mitigate these problems by allowing documents to be converted into a common intermediate format for purposes of indexing and viewing. Reading applications need then support only this single common format. Because the addition of new information sources is a frequent and integral part of the reading activity, conversion into this common format — including the extraction and processing of any requisite metadata, must be as fast, transparent, and automatic as possible.

1.2 UC: A System for Fluid, Seamless Interaction

This paper describes a system for reading from multiple documents which addresses the above mentioned shortcomings of current digital reading systems. The system, called *UC* (a name deriving from its initial but no longer used internal name “UpLib Client”) is built on the *UpLib* personal digital library platform [6], which provides an extensible, format-agnostic, and secure document repository layer. *UpLib* achieves a degree of universality in accommodating multiple document formats through heavy reliance on two principal transductions of the original document, one into the *image* domain and the other into the *text* domain. From the perspective of *UC*, the image projection facilitates simulation of the visual experience of reading from multiple paper documents, while the text projection enables search-based navigation. By virtue of its leverage of *UpLib*,

incorporating a new document into the *UC* system becomes largely a matter of producing its two fundamental projections. To accomplish this in a manner that imposes the least amount of latency and cognitive load on the user involved in a reading task, an area of the user's file system is designated for scanning and processing, so that any document placed there is automatically incorporated into the workspace.

For each document in the workspace, the *UC* system uses an instance of a single consistent reader application, *ReadUp*, described in detail in a parallel submission to the present conference. Despite the compositional relationship between *UC* and *ReadUp*, the user experiences fluid interaction within what appears to be a single application. This is accomplished by making every document behave as if it were always open in a separate instance of *ReadUp*, independent of whether it is an object of current focus. Thus, from the user's point of view, there is no need to launch a separate viewer application to interact with a document; one merely zooms, pans, and focuses on a document to interact with it.

The remainder of this paper is arranged as follows. Section 2 summarizes relevant prior work. Section 3 motivates key design aspects of *UC* by considering those affordances most valuable when reading from multiple documents. A detailed description of the *UC* is provided in Section 4. Section 5 lists future work and Section 6 concludes.

2 Prior Work

The importance of effective visual interfaces to collections has been recognized by a number of researchers [2,3,4], but in the context of remote digital libraries where finding and using materials are constrained to be separate activities.

Several previous systems have used treemaps and zoomable user interfaces (ZUIs) similar to those described in this paper. The original implementation of this idea was in the PhotoMesa image browser [7]. Many of the principles driving our system design are derived from this work. The International Children's Digital Library (ICDL) [8] builds on PhotoMesa by laying out children's book cover thumbnails in zoomable treemaps. Our work differs from the ICDL in several respects, including the use of hybrid treemaps, the specific modes of navigation supported, the ability to generate the collection automatically, and the accommodation of large, heterogeneous collections.

Scatter/Gather [9] and Visual Relevance Analysis [10] both provide visualization of document collections based on interactive clustering techniques, and the latter presents an interactive treemap visualization of topics. Unlike the present work, neither attempts to support document use (i.e., reading, annotation, etc.) as an integral part of the activity.

The *Web Book* and *Web Forager* [11] uses a 3D book metaphor to organize collections of web pages into a three level hierarchy for navigation and reading. Our work differs in that it provides visualizations of multiple collections simultaneously. It is also not limited to web pages or to a fixed hierarchy. *Data*

Mountain [12], another technique for interacting with collections of web pages, provides a planar 3D workspace in which the user can manually organize web page thumbnails. In contrast, our techniques provide automatic visualization layouts, overviews of multiple collections, and support for a variety of document types.

Much of the prior work on electronic reading has focused on reading individual documents. *Document Lens* [13] presents a fisheye view of a grid of thumbnails of a document's pages. *XLibris* [14] focuses on providing for freeform annotations on an electronic document. The *Multivalent Browser* [5] attempts to create a standardized reading and annotation platform independent of a document's originating format; our work extends this idea in dealing with document collections. In contrast to the above cited work, we focus on multi-document reading tasks where the ability to transition from collection-browsing and overview mode to detailed comparison of the contents of the documents are typical and frequent operations.

3 Effective Interaction

3.1 Fluid Reading

Current document browsers and reading tools have overhead that can be cognitively disruptive for reading. One limitation in these systems is that not all document types are supported by the same reading application. As a result, not all documents support the same functionality, such as thumbnail overviews, highlighting, freehand annotations, text notes, or keyword search. Moreover, even when similar functionality is available in a document's native applications, the user interface controls to that functionality are not standardized.

A related limitation is that document finding and browsing is divorced from document reading. As a result, reading a document typically involves opening a new application external to the one used to find it. This act of opening an application often introduces small delays both for the system to respond and for the user to orient to the change. Moreover, if the new application is displayed in its own window, it can also introduce window management problems. These issues substantially increase the cognitive load of reading.

A more subtle effect of the separation between browsing and reading is that the reader is often forced to prematurely commit to spending time with a document that turns out not to be most relevant. Applications for finding and browsing typically display a fixed amount of information about the document so the user must make a decision whether to incur the costs described above based on limited information.

3.2 Multi-document Reading

Because document finding is typically separated from document reading, multiple document reading tasks are particularly difficult in the electronic realm.

Many applications provide facilities for navigating between multiple documents of the same type, such as tabs in a web browser. However, reading tasks may involve moving between several types of documents such as web pages, word processing documents, presentation slides, etc. This type of task suggests the need for techniques to transfer attention rapidly among documents, regardless of type, in small working sets.

A related problem is being able to view the pages of multiple documents at once. With paper, this is often accomplished by spreading out documents on a table. Reading applications often provide thumbnail facilities for viewing multiple pages from a single document but not from multiple documents. The ability to do so can be important for making comparisons [15] and for getting an overview of a collection of documents [16].

3.3 Finding Documents

Perhaps the most widely accepted method of finding relevant documents is keyword search. This method has a number of limitations in the case of personal digital libraries. Keyword search is fundamentally a recall-based user interface. A classic principle in user interface design is to minimize the memory load on the user [17]. This is often referred to informally as “supporting recognition rather than recall.”

A general problem with keyword search is that a user’s vocabulary often does not match the desired document’s vocabulary. This means that users may not be able to find certain documents or they may have to experiment with several queries before they find what they want. A number of techniques have been introduced to deal with this problem such as term aliasing or personalizations like those in Haystack [18]. Nevertheless, these heuristics are not likely to work in all cases. It is also often the case that the user is not able to formulate a query at all. Some examples include searching for a picture, a specific page layout, or some other visual property of a document. In these cases, the user needs techniques for interacting with documents visually rather than textually.

4 The *UC* System

The *UC* system developed in our research group integrates a number of recent user interface, information visualization, and digital library techniques with the goal of addressing the problems described in the foregoing. The system is also designed to work well with pen-based tablet computers where traditional user interface controls, such as scrollbars and text boxes, are of limited effectiveness. While the foregoing sections have described the motivation and design rationale for this system, the present section and the next describe the system in detail.

UC is in part a user interface for interacting with documents in an *UpLib* [6] repository. As such, it is able to connect to existing *UpLib* repositories, such as those that have been manually constructed. However, it also provides a powerful facility for finding documents in the user’s file system and automatically adding them to the repository.

4.1 Continuous and Quantum Treemaps

UC uses continuous and quantum treemap [19] layouts to present collections of documents. Continuous treemaps are space filling visualizations of trees that assign area to tree nodes based on the weighting of the nodes. In continuous treemaps, the aspect ratio of the cells is not constrained even though square cells are often preferred. Quantum treemaps extend this idea by guaranteeing that cell dimensions are an even multiple of a unit size. These layouts are described in more detail by Bederson et al. [19].

The page thumbnails used in *UC* are based on document icons created by the *UpLib* system. In addition to portraying the often unique appearances of the documents themselves, these icons can also be augmented with overlays such as important text or pictures to further assist users in differentiating one document's icon from another. These unique document icons serve to enhance spatial memory of document locations within *UC*'s spatial treemap layouts.

Users often collect large numbers of documents that are more than can be meaningfully displayed using these thumbnail based visualizations. As a result, the system has a threshold number of documents above which thumbnails are no longer displayed and standard continuous treemaps are shown instead. The continuous treemap attributes of cell size and color can then be used to display quantitative information about the underlying collections such as number of documents, number of pages, file sizes, last modified or last viewed dates, alphabetic ordering, values for manually assigned metadata, or other semantic properties extracted from the documents. Table 1 lists some common operations in *UC* and their allowable starting and ending treemap types. When an operation supports both treemap types, the system chooses between them based on the number of documents in the current view.

A number of algorithms can be used to arrange continuous treemaps. Our system supports both squarified and strip treemap layouts. Because our data is typically ordered by date, file name, etc., the default layout is a strip treemap.

Table 1. The available transitions between the two types of treemaps. In the row header, “C” represents continuous treemaps and “Q” represents quantum treemaps. An “X” in a cell indicates that the column's operation can start in the first treemap type and end in the second.

	Zoom Out	Zoom In	In Context Search	Limit View to Search Results	Explode to Pages	Read Document
C → C	×	×	×	×		
C → Q		×		×		
Q → C	×					
Q → Q	×	×	×	×	×	×

Bederson et al. present evidence that the strip layout is preferable over other available treemap algorithms for presenting ordered data [19]. This layout also has reasonable behavior as documents are added to collections or the window aspect ratio changes.

4.2 Navigation

An important aspect of the interface is the fluidity of navigation. This allows the user to focus on the documents rather than on interacting with the tool. In *UC*, the navigation controls are similar to those in other ZUIs. Left click on an object or group of objects zooms in and either button clicked on the background zooms out.

One problem that arises from combining a zoomable user interface with continuous treemaps involves conflicts with aspect ratios. The cells in continuous treemaps have a range of aspect ratios, as demonstrated in Fig. 1. Each of these different aspect ratios may differ from that of the view window. As a result, zooming in on these cells, as you might zoom into a country on a map, may not increase the amount of screen space devoted to a cell. For example, if a cell consumes the full width of the window but only half the window's height,

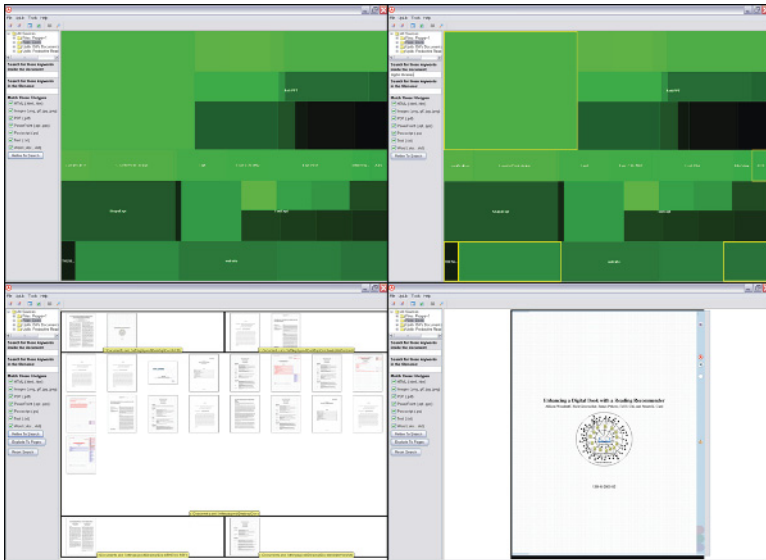


Fig. 1. An example keyword search scenario in *UC*. The scenario begins with a continuous treemap visualization of a document collection (top left). As the user types search terms (top right), interactive highlights appear for groups with matching documents. The user presses a button to limit the view to only matching documents (bottom left). Finally, the user zooms in on a document and begins reading with the *ReadUp* reader (bottom right).

then there is no way to increase the size of the cell while keeping it entirely on screen.

There are two solutions to this problem. The first is to zoom in so that some of the cell is off the screen. In the previous example, this would mean zooming in to the point where the cell's height is equal to the window's height and the cell's width is twice the window's width. The advantage of this approach is that it preserves the kind of standard geometric zooming used in many other systems. The disadvantage is that the change in scale needed to zoom a cell is essentially unlimited. In our previous experience with ZUIs, we have found that this type of large change in scale can be disorienting even when the transition is smoothly animated.

The second solution, and the one used in *UC*, is to zoom and morph the cell to the window size and aspect ratio while leaving the rest of the layout in place. The primary advantage of this approach is that it minimizes the visual disturbance of the display since only a single cell moves. Animating the transition can further help orient the user during the change.

The continuous treemap views also provide previews of the layout at the next level down when the user moves the mouse into a cell. One bit of information this preview provides is whether the layout at the next level down is a continuous treemap or a quantum treemap. The preview can also give a rough idea of the number and structure of groups at the next level.

Importantly, the quantum treemaps in *UC* are navigated with standard view-point animations while the document layouts remain static. This allows the user to build awareness and memories for spatial relationships within smaller working sets of documents.

4.3 Refining and Searching Collections

UC provides controls to refine which documents are displayed. First, the interface provides mechanism to search for specific content within the documents. The system incrementally highlights matching documents as letters are added to the search query to immediately indicate matching documents. The user can also choose to update the view to re-layout with only documents that match the current query. An example of this type of search, ending in reading a document, is shown in Fig. 1.

For *UpLib* repositories generated from a file system, the interface also provides controls for finding patterns in the file path name and for limiting which file types are displayed. We also plan to generalize this to include controls for limiting file size, number of pages, last access or modification times, and other types of document metadata.

4.4 Explode to Pages

For multiple document reading, users need a mechanism to easily make comparisons between multiple documents. This is accomplished in the *UC* by allowing the user to explode a set of documents into their pages. As with collections of

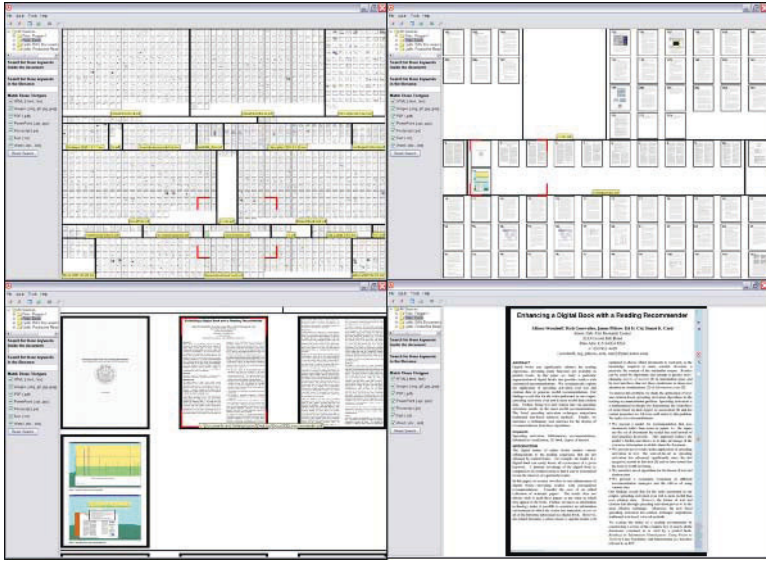


Fig. 2. An example of exploding a document collection to pages. A working set of documents is obtained through a search (top left). Pressing a button explodes the documents to pages. The user zooms in on the pages of several interesting documents (top right). The user continues zooming in to a portion of a single document (bottom left). The user then selects a document page and begins reading at that page in *ReadUp* (bottom right).

document icons in *UC*, these page thumbnails are laid out in quantum tree maps. Using an identical layout at both the document and individual page level allows the user interface to provide a consistent set of functionality and interaction techniques between the two types of views. These exploded document layouts not only support comparisons between pages of multiple documents, they can also provide overviews of collections and a mechanism for quickly jumping between pages in multiple documents. Fig. 2 shows the layouts in use.

4.5 Integrated *ReadUp* Reader

An important principle in the system design is that reading be fluid and not require opening a separate application. This is achieved in *UC* by integrated the *ReadUp* document reader provided by the *UpLib* system. This reader provides thumbnail overviews, freehand pen annotations, highlighting, text sticky notes, bookmarks, and full text keyword search.

Annotations made in the reader are automatically stored in the same *UpLib* repository that stores the image and text projections. This allows the user to fluidly read and annotate documents without having to manage annotated files or explicitly save changes. The user's activity, such as how long the user viewed a particular page, is also stored in the *UpLib* repository which can be

used to inform the visualizations provided by *UC*. For instance, documents that a user reads frequently can be made more prominent to support easily revisiting them.

4.6 Implementation Issues

UC is a Java application that connects to one or more *UpLib* repositories as a network client. The *UpLib* server also runs an extensible set of document analysis and processing operations on each document when it is acquired. These operations extract metadata, create various thumbnail versions of each page image, and perform full-text indexing. The *UC* system requests information from the server, such as thumbnails, word bounding boxes, or document metadata, to enable interaction with documents in the repository. *UC* also includes a utility to scan a portion of the file system specified by the user. This scanner then adds supported document types that it finds to a specified instance of an *UpLib* repository. *UC* extends the set of document types handled by the *UpLib* system to include those produced by popular word-processing, spreadsheet, and presentation applications.

5 Future Work

In this paper we have described the design rationale, feature set, and implementation of *UC*, but not its evaluation in controlled user trials. User evaluation is essential to fully understand the benefits of the approach and to identify weaknesses and opportunities for improvement. A complicating factor in designing such a user study is that the time scale of the sort of multi-document reading tasks for which this system is intended can span days and weeks, involves the interaction of many factors, and is subject to a great deal of variability in both the methods currently employed and their effectiveness. The common expedient of devising a small, artificial task on which to compare and report statistical *p*-values has become *de rigueur* in the user-interface research community, but likely would not lead to much insight. Instead, working with our colleagues we intend to undertake a descriptive ethnographic study in an authentic multi-document reading setting.

UC's user interface currently supports refineable automatic treemap layouts. We would also like to look at combining these automated layouts with user defined layouts such as those in Data Mountain [12] and VKB [20]. The automatic layouts could then bootstrap the users manually constructed working sets for particular tasks.

6 Conclusion

This paper has described an advanced visual interface system for navigating within a personal digital library and interacting with its contents in a fluid and

consistent manner. The system is based on a zoomable planar representation of the collection in which sub-collections and individual items become accessible as the scale is varied. In contrast to most existing collection visualization and management systems, provision is made for interacting with elements of the collection seamlessly and *in situ*, without having to open a separate application. In contrast to most document reading interfaces, the system facilitates rapid and fluid movement from one document to another document, again without leaving the overarching spatial representation of the collection, thus preserving a sense of orientation and reinforcing spatial memory throughout the course of interaction. In addition, the system streamlines the incorporation of new documents within the personal digital library by automating the conversion of those documents into a common compatible format.

Acknowledgments

This research has been funded in part by contract #MDA904-03-C-0404 to Stuart K. Card and Peter Pirolli from the Advanced Research and Development Activity, Novel Intelligence from Massive Data program.

References

1. Sellen, A.J., Harper, R.H.: *The Myth of the Paperless Office*. MIT Press (2003)
2. Chang, M., Leggett, J.J., Furuta, R., Kerne, A., Williams, J.P., Burns, S.A., Bias, R.G.: Collection understanding. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ACM Press (2004) 334–342
3. Perugini, S., McDevitt, K., Richardson, R., Perez-Quinones, M., Shen, R., Ramakrishnan, N., Williams, C., Fox, E.A.: Enhancing usability in citidel: multimodal, multilingual, and interactive visualization interfaces. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ACM Press (2004) 315–324
4. Sumner, T., Bhushan, S., Ahmad, F., Gu, Q.: Designing a language for creating conceptual browsing interfaces for digital libraries. In: JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, IEEE Computer Society (2003) 258–260
5. Phelps, T.A., Wilensky, R.: The multivalent browser: a platform for new ideas. In: DocEng '01: Proceedings of the 2001 ACM Symposium on Document engineering, ACM Press (2001) 58–67
6. Janssen, W.C., Papat, K.: Uplib: a universal personal digital library system. In: ACM Symposium on Document Engineering. (2003) 234–242
7. Bederson, B.B.: Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In: UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology, ACM Press (2001) 71–80
8. Druin, A., Bederson, B.B., Weeks, A., Farber, A., Grosjean, J., Guha, M.L., Hourcade, J.P., Lee, J., Liao, S., Reuter, K., Rose, A., Takayama, Y., Zhang, L.: The international children's digital library: Description and analysis of first use. *First Monday* **8** (2003) http://firstmonday.org/issues/issue8_5/druin/index.html.

9. Cutting, D.R., Pedersen, J.O., Karger, D., Tukey, J.W.: Scatter/gather: A cluster-based approach to browsing large document collections. In: Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1992) 318–329
10. Pediotakis, N., Hascoët-Zizi, M.: Visual relevance analysis. In: DL '96: Proceedings of the first ACM international conference on Digital libraries, ACM Press (1996) 54–62
11. Card, S.K., Robertson, G.G., York, W.: The webbook and the web forager: an information workspace for the world-wide web. In: CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press (1996) 111–117
12. Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D., van Dantzich, M.: Data mountain: using spatial memory for document management. In: UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology, ACM Press (1998) 153–162
13. Robertson, G.G., Mackinlay, J.D.: The document lens. In: UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology, ACM Press (1993) 101–108
14. Price, M.N., Schilit, B.N., Golovchinsky, G.: Xlibris: the active reading machine. In: CHI '98: CHI 98 conference summary on Human factors in computing systems, ACM Press (1998) 22–23
15. Adler, A., Gujar, A., Harrison, B.L., O'Hara, K., Sellen, A.: A diary study of work-related reading: design implications for digital reading devices. In: CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co. (1998) 241–248
16. O'Hara, K.P., Taylor, A.S., Newman, W.M., Sellen, A.: Understanding the materiality of writing from multiple sources. *Int. J. Hum.-Comput. Stud.* **56** (2002) 269–305
17. Shneiderman, B.: Designing the user interface: strategies for effective human-computer interaction. third edn. Addison-Wesley Longman Publishing Co., Inc. (1998)
18. Adar, E., Kargar, D., Stein, L.A.: Haystack: per-user information environments. In: Proceedings of the eighth international conference on Information and knowledge management, ACM Press (1999) 413–422
19. Bederson, B.B., Shneiderman, B., Wattenberg, M.: Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Trans. Graph.* **21** (2002) 833–854
20. Shipman, F.M., Hsieh, H., Moore, J.M., Zacchi, A.: Supporting personal collections across digital libraries in spatial hypertext. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ACM Press (2004) 358–367

MedioVis – A User-Centred Library Metadata Browser

Christian Grün, Jens Gerken, Hans-Christian Jetter, Werner König, and
Harald Reiterer

Workgroup Human-Computer-Interaction
Department of Computer and Information Science
University of Konstanz
78457 Konstanz, Germany

{gruen, gerken, jetter, koenigw, reiterer}@inf.uni-konstanz.de
<http://hci.uni-konstanz.de>

Abstract. MedioVis is a visual information seeking system which was designed especially for library data. The objective target was to create a system which simplifies and optimizes the user's information seeking process and thus further motivates the user to browse in the library stock. To enhance the motivation special attention was given to consider joy of use aspects during the design of the user interface. The primary user interface design is based on multiple coordinated views to offer a great variety of exploration possibilities in a direct-manipulative manner. To accomplish a self-explanatory usability of the system for non-expert users, the development was accompanied by continuous user tests with casual and regular library users. At the end of the development process a comprehensive summative evaluation was conducted, comparing efficiency and joy of use of the existing web-based catalogue system KOALA of the library of the University of Konstanz with the MedioVis system. The results of this comparative evaluation show a significant improvement of the efficiency of the information seeking process with the help of MedioVis. The users also rated MedioVis significantly better in all dimensions of its hedonic quality and appeal compared with KOALA.

1 Motivation

Retrieving relevant information on library catalogues has long time been quite a tedious job as the visual presentation of bibliographic metadata ignored most of the rules of usability and attractiveness which we are facing today and many systems did not match the users' retrieval behaviour as discovered by Borgman [1]. First improvements could be observed when early, purely text-based interfaces were replaced by graphical representations. Nevertheless, in most of the cases the internal systematic metadata structures were only visually reproduced, disregarding the user's need to get non-technical, more inviting views on the desired information. Moreover, as a means of retrieving information libraries have to compete with the internet. Although the internet is often criticised for its unreliable information space, especially by information experts, it should be accepted that more and more information seekers

choose the advanced retrieval technologies of web engines to get quick access to relevant information.

This observation was decisive for us to develop a JAVA-based application that motivates the user to explore the complete information space which a library has to offer. We are convinced that library stocks still have and will always have their very own qualities regarding the intellectual choice of relevant works and consistent metadata editing, and we tried to create a browser interface, enabling the user to work with a more web-oriented browsing paradigm instead of applying the simple search and finding strategy. Moreover, we took advantage of the broad availability of online sources to enrich the metadata of the library with additional textual and multimedia data.

Chapter 2 outlines the theories and concepts which have influenced the development of the application. Evaluation has played a central role during the development of MedioVis. The evaluations of our preceding projects INSYDER [2], INVISIP [3] and VisMeB [4] (all of them metadata browsers) represented a good base for crucial design decisions. All beta versions of the system were evaluated in order to get creative support from real users and to confirm or influence the design rules. The results of our evaluations, finalized by a summative evaluation, are worked out in Chapter 3. Conclusions are given in Chapter 4.

2 MedioVis System Design

2.1 Joy of Use

A usability aspect which plays a minor role in library catalogues is the attractiveness of user interfaces. However, the role emotion and aesthetics plays for people has already been pointed out as a central psychological issue by James in 1884 [5] and has widely been discussed since. Nevertheless it had not been applied to usability until Kurosu and Kashimura [6] noted that the ergonomic quality of a product does not implicitly coincide with the usability perceived by the user. Considering Glass' thesis from 1997, "I predict that joy of use will become an important factor in product development and development success." [7], we can observe that many commercial products like computers, cellular phones, operating systems, etc., are already very fun-oriented, aiming at the users' need for aesthetics and entertainment. Tractinsky even argues that "For many users, other aspects of the interaction [than aesthetics] hardly matter anymore." [8].

A major question still to solve is how aesthetics can be measured and parameterized in general. Although we have a common idea that aesthetic objects should be symmetric, balanced or well proportioned, there is – fortunately? – no general instruction set prescribing how to create aesthetic interfaces. Jordan [9] proposes some helpful methods and guidelines for the design of pleasurable products whereas Hassenzahl et al. [10] underline the importance of hedonic quality (novelty, originality) of a software product. Other interesting, more formalized approaches are in progress to construct mathematical concepts [11] and find mathematical explanations [12] for accessing aesthetic phenomena, but they are still in an early stage.

Our aim was to create a simple attractive and pleasurable interface, for which we collaborated with communication designers of the University for Applied Sciences Konstanz. By regularly testing our prototypes with users we could quickly react on possible flaws of the system. A consistent colour and font range and animations were applied to give the system its own identity, and the results of our evaluations proved that users seem to appreciate and enjoy the overall appearance of MedioVis.

2.2 Visual Information Seeking

Most of the conventional library web catalogues offer three or four successive steps to control the user's information seeking process:

1. Search for Keywords (simple / advanced input forms)
2. Display of the results as overview (10 / 20 / 50 per page, tabular / list oriented)
3. Display of single results (list oriented / full text)
4. (Non-Standard) Overview of all selected results

All steps are normally visually separated from each other, i.e. the user is often forced to return to the first page to start a new search. This sequential approach reminds of real library visits: first the book titles are researched in the catalogue systems, then the correct book shelves have to be found in the library to finally review the book. It is interesting to see that real life drawbacks have been transferred to computer logics, as it may seem obvious that all steps can easily be combined. Although computer screens are limited by their resolution, the integration of all panels on one screen can easily be implemented, even in web sites, by cleverly partitioning the available space.

The approach to combine several views with semantically similar data is known as "Multiple Coordinated Views" (MCVs) [13] [14], and the visualizations in MedioVis follow several design principles expressed by Shneiderman [15]. As seen in Fig. 1, the input for query terms is located on the top area of the window. We used a table-based view to give an overview of all queried documents. Tables are a popular visualization concept as they can display a huge amount of data in a consistent way. We can establish an interesting analogy between browsing real and digital libraries by using tables [16]: real libraries can be browsed by walking along the shelves to spot material which is similar in content. Titles presented in a virtual table can be spatially separated in the library, but they can have other common attributes such as title keywords, media types or publication years. Hence one of the most important features of a table and a big advantage over physical libraries is the capability to sort data by specific attributes as the users can decide for themselves which attribute seems most important to them. The original shelf order can still be simulated by restricting the search to a certain subject area and sorting the titles by their signature. Columns can simply be sorted in MedioVis by clicking on the headers.

Following the principle of coordinated views, the dataset which is currently focused in the table is focused in all other visualizations. Detail-on-Demand is given by the "Title View" which lists all attributes of the focused title. If a relevant title is selected, it is moved into the "Selected Media" view. Titles in this view can be printed, saved on disk or sent via e-mail. Alternatively the Media Location is displayed in this area. The currently focused title is marked in a floor plan of the

library which should help the user to locate it in the real library (Fig. 2). The user interactions in MedioVis were reduced to basic mouse actions. A title in the table view can be focused by simply moving the mouse across it. If the mouse button is clicked, titles are being selected.

An alternative view to the table is a scatterplot-like visualization which was termed “Graphical View” for simplification (see Fig. 2). As it was not included in the evaluation, it is not further described in this paper. An advanced zoomable version named ZUIScat will be integrated in the near future [17].

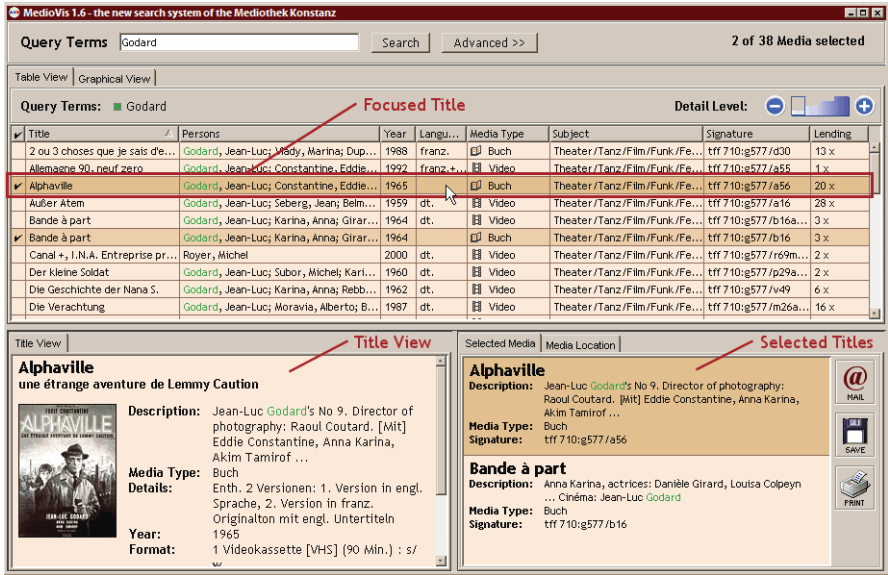


Fig. 1. MedioVis main screen

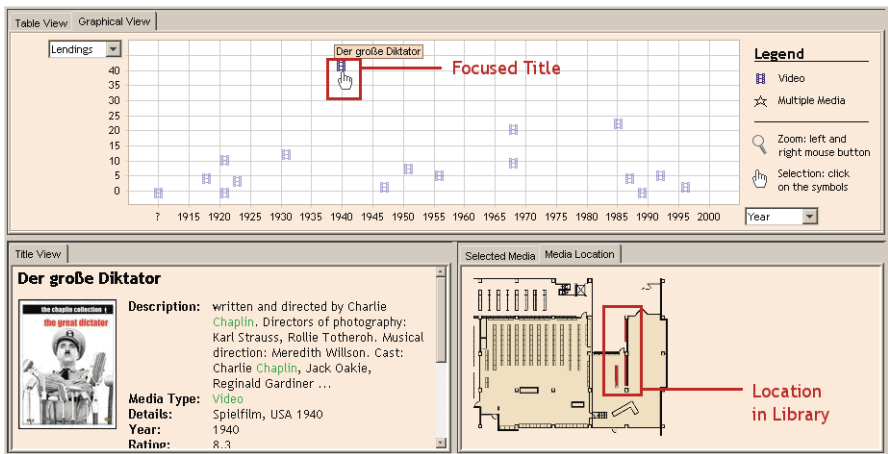


Fig. 2. Scatterplot, Location Panel

2.3 Information Retrieval

Many conventional library catalogues still cling to their internal meta-data structure and transfer its logic to the search forms which are to be handled by users that do not know about bibliographic issues and technical limitations which often are obsolete. To give an easy example, the existing local library catalogue KOALA expected a fixed format to search for names defined by “Surname, First Name”, which was why some of our test persons did not manage to find any results at all. Today’s retrieval operations should therefore be uncoupled from formal issues, allowing the user a high level of flexibility and freedom. Early tests of our prototypes confirmed the assumption that most users are accustomed to the presentation of one single search form as it is used in common web engines. In fact, many users used query terms like “DVD”, “Movie” or an ISBN. So we implemented a single input area as standard which queries most important metadata attributes such as Title, Author, Year, Media Type, Full Text Description, etc. Whereas the amount of data which has to be checked by such a query would have disallowed such a procedure in the past, it makes no difference for today’s technology and for indexing algorithms.

One of the most striking advantages of library catalogues opposed to web indexes, as earlier mentioned, is the consistency and structure of the data behind. So it still makes sense to include an additional advanced search form to offer the user explicit searches for persons, titles, etc. Such a form is also implemented in MedioVis and can be invoked by an “Advanced” button. It allows searching for title keywords, persons, year ranges, media types, library sections and languages. Moreover search terms can be AND/OR combined (“Find all terms” / “Find at least one Term”), and the choice is given between querying for complete strings and substrings. The search form stays visible if the result set is returned and visualized.

3 Evaluation

No matter how usable a software system proves to be during formative evaluations, its usefulness in a real world environment depends on the alternative software systems available and of course on their quality. On this account it is essential to compare the developed software system with its competitors in order to be able to determine whether the development was successful and communicate this to possible customers. Accordingly we decided to conduct a quantitative Usability Experiment to compare MedioVis with KOALA which is the retrieval system of the library of the University of Konstanz currently in use. We put our main focus on the questions whether users were able to solve realistic tasks more efficiently (objective measurement) and how they would rate MedioVis in comparison to KOALA (subjective measurement).

3.1 Experiment

The Experiment took place within the scope of a lecture “Usability Engineering 2”, given by Prof. Dr. Harald Reiterer, Workgroup HCI at the University of Konstanz, from December 2004 to January 2005. After some detailed training we selected four students of the lecture as test monitors. Each of them conducted six test sessions so that altogether we had 24 subjects. To avoid test monitor effects we standardized and

structured the test procedure to the greatest possible extent. Since our test monitors had not been involved in the development of MedioVis we could furthermore exclude test monitor effects due to possible personal interests in specific results. We wanted to test two main hypotheses, expressed as null-hypotheses:

1. In terms of task completion time there is no significant difference between MedioVis and KOALA.
2. In terms of subjective user rating there is no significant difference between MedioVis and KOALA.

To check the first hypothesis we measured the time it took our 24 participants to complete realistic tasks with the help of either MedioVis or KOALA. The second hypothesis was checked with the help of two questionnaires, SUS [18] and Attrakdiff [19]. The latter does not only measure the pragmatic quality of a software product but furthermore hedonic quality and appeal – two aspects which are, in our understanding, extremely important for a software system which users like to work with. We did not especially concentrate on task accuracy since this is always to some extent contradictory to the measurement of task completion time, and this was where we put our main focus.

3.2 Method

3.2.1 Participants

We chose 24 students of the University of Konstanz as subjects, since they would be potential end-users. In addition we concentrated on students in the fields of “Literature, Art and Media Science” (11 subjects, 9 female & 2 male), as we expected them to be an above average part of our target group. Since one of our main aspects during the development of MedioVis was to develop an easy to learn system, we did not involve any students in the field of computer science. The subjects were between 18 and 28 years old, with the medium age at 22.

3.2.2 Software Systems

The benchmark for MedioVis (see chapter 2) was KOALA, a web based catalogue system, which allows the user to search the library stock of the University of Konstanz. It offers a “simple” and an “advanced” search dialogue. The “simple” mode asks the user to specify at least author or title, but does not include a general search field. The “advanced” mode includes more specific search fields such as signature or year. The result is presented in a list, showing ten hits per page. The user can get a detail view of each result or several results at once (see Fig. 3).

3.2.3 Data Base and Tasks

Our test data was based on a copy of the library stock of the Mediothek, a department of the library of the University of Konstanz. The data was enriched with additional heterogeneous data such as posters and movie ratings of the Internet Movie Database (IMDB). KOALA was restricted to the Mediothek inventory, but used the up-to-date data, which slightly differed from our copy used in MedioVis. We were able, however, to consider this aspect with our task design.

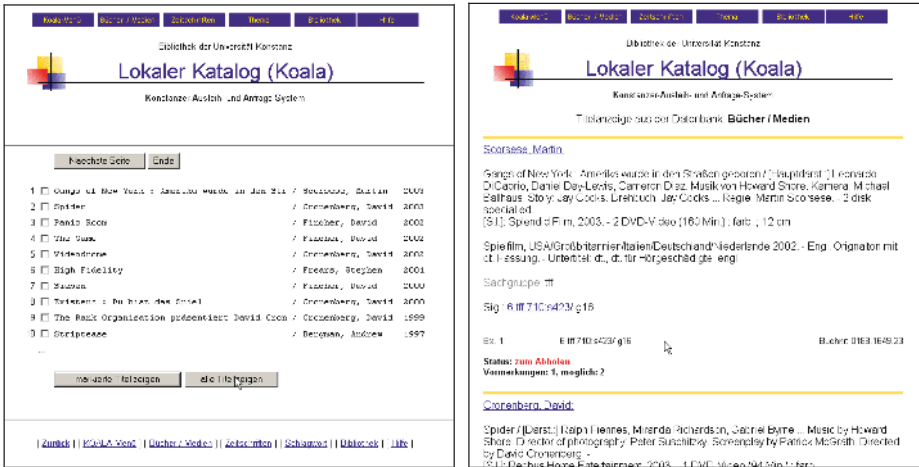


Fig. 3. KOALA: list-based overview, detail view

The main focus of our task design was to simulate realistic seeking processes which would appear in the Mediothek. In order to achieve this we consulted an expert of the Media Science faculty of the University of Konstanz. Together we developed twelve tasks which were separated into two task sets, each including six tasks. Each task set contained three specific fact finding tasks and three extended fact finding tasks [20]. All tasks were designed in a way that they could be solved with both systems in an appropriate time frame and without major problems. In addition we developed six introduction tasks (three for each task set), which weren't considered in the statistical analysis. The purpose of those was to assure that all test persons would start with a comparable knowledge of each system, especially since none of the users was familiar with MedioVis.

3.2.4 Procedure

The experiment was conducted in our HCI usability lab. We used an Intel Pentium IV with 3 GHz, 1 GB RAM and a 19" TFT display. Each session was recorded with the help of Techsmith Morae 1.1 recording software [21]. During the session three persons were present in the office: The test-subject, the test-monitor and one minute taker. The experiment started with a Pre-Test questionnaire. In the following each subject started to work with task set 1 and the first system and afterwards switched over to the second system and task set 2. In addition the order of the systems was alternated: Fifty per cent of the participants started with KOALA, the other half started with MedioVis. Task sets were not alternated (see Fig. 4). In front of each task set each subject concluded the introduction tasks to get to know the system. After having worked with a system, the subjects filled in the SUS and Attrakdiff questionnaires. Each session lasted about 45-60 minutes.

Subject	System 1	Task- Set	Questionnaire	System 2	Task-Set	Questionnaire
1	KOALA	1	SUS, Attr.	MedioVis	2	SUS, Attr.
2	MedioVis	1	SUS, Attr.	KOALA	2	SUS, Attr.
3	KOALA	1	SUS, Attr.	MedioVis	2	SUS, Attr.
4	MedioVis	1	SUS, Attr.	KOALA	2	SUS, Attr.

Fig. 4. Experiment Procedure, 4 example subjects, Attr. = Attrakdiff

3.2.5 Experimental-Design

We used a combination of repeated-measures design (for questionnaire analysis – “system” as within-subjects factor, “order” as between-subjects factor, questionnaire score as dependent variable) and between-subjects design (for task completion time analysis – “system” as between-subjects factor, task completion time as dependent variable). As a result, we had 24 subjects who rated both KOALA and MedioVis with SUS and Attrakdiff. For task completion time analysis we had 12 subjects working on task-set 1 with KOALA and 12 working on task-set 1 with MedioVis, analogue task-set 2. We did not choose a repeated-measures design for task completion time analysis since we think that working on the same or at least very similar task with two different systems can influence the results in a significant manner. Based on our hypothesis that this would not be the case with questionnaires, we designed the experiment in this way to combine the advantages of both designs.

3.3 Results

3.3.1 Task Completion Time and Task Accuracy Overview

The average task completion times were 87 seconds (task set 1) and 96 seconds (task set 2) for KOALA users. In contrast it took our MedioVis subjects on average 49 seconds (task set 1) and 40 seconds (task set 2) respectively. This difference is highly significant ($p < 0.01$). Levene’s Test for the homogeneity of variances furthermore showed that the assumption made by the Oneway ANOVA, that variances of the groups are all equal, is justified ($p = \text{not significant, n.s.}$).

3.3.2 Task Completion Time in Detail

Since the time differences were that significant, we were primarily interested if this was caused by a specific type of tasks or if it was just based on the different systems. We will look at the two task sets separately in order to keep everything in mind.

As mentioned above, we clustered the tasks in specific fact finding and extended fact finding tasks, the latter being expected to be the more complex tasks. Regarding task set 1, the average completion time for specific fact finding tasks was 78 seconds for KOALA users and 35 seconds for MedioVis users (significant, $p < 0.01$). Extended fact finding tasks took KOALA users 95 seconds on average and MedioVis users 62 seconds. Although this seems slightly closer, the difference is still highly significant ($p < 0.01$). Analyzing every task of the specific fact finding tasks for itself, we again have significant differences for each of them (at least $p < 0.05$). Especially task 3 was outstanding, taking KOALA users on average 139 seconds and MedioVis users only 50 seconds to complete it. The task demanded the users to search for a film, available on a multilingual video cassette. Since KOALA users could not limit their search on

video cassettes nor multilingual versions, they had to check each search result manually (see Fig. 3). MedioVis users could benefit from the table visualization, the sorting features and the multi coordinated views (see chapter 2.2). They could directly decide whether they found the correct film or not by checking out the searched attributes (see Fig. 3).

The second task set took the subjects 50 seconds on average to complete the specific fact finding tasks with KOALA and 29 seconds with MedioVis (significant, $p < 0.01$). Looking at the extended fact finding tasks, this gets even more explicit, taking KOALA users 143 seconds on average and MedioVis users only 51 seconds (significant, $p < 0.01$). A detailed analysis reveals that the same advantages as in task set 1 are responsible for the better performance of MedioVis. However, there were two tasks where both systems performed nearly equal ($p = n.s.$). In both cases the user did not have to formulate a new search query but could work on the prior result set. The extended fact finding task asked the users to find out the screenplay author for three of the films in the result set. Although KOALA users had to load a new web page with the detailed information, they were nearly as fast as the MedioVis users (87 seconds vs. 77 seconds). It seems to be the case that it takes MedioVis users more time to find a specific title in the result set, probably due to the rather small fonts within the table view.

Next we tried to identify other variables which might have an effect on the results. We conducted multivariate ANOVAs with gender (male, female), semester (first semester, higher semester) and field of study (seven characteristics) but could not detect any significant impact of those variables on our dependent variable task completion time. Another interesting aspect is the use of the scatterplot. We did not include the scatterplot specifically in our test design but did not bar our users from trying and using it. Although most of the users were confused by the mathematical appearance, five subjects used it successfully for some tasks. This shows that we still have to work on the appearance of the scatterplot to get it used by more people.

3.3.3 Questionnaire Analysis

As stated before we used a repeated-measures ANOVA to analyze the results of our two questionnaires, Attrakdiff and SUS. The analysis revealed that the system order had no significant effect on the results, which proved our hypothesis that such a design is suitable for questionnaires. Analyzing the Attrakdiff scores we noticed that MedioVis scores significantly higher on each of the four scales ($p < 0.01$) – with particular high scores on the pragmatic quality (PQ) and on the appeal scale (APP, both times 5.5 compared to 3.9 and 3.7 for KOALA on average on a 7 point scale). The values of hedonic quality (HQ) measurement are also clearly above KOALA with 5.0 on average on the identity (HQ-I, KOALA 3.8) and 4.7 on average on the stimulation scale (HQ-S, KOALA 3.0). This is remarkable since it clearly proves our initial design concept of a system which is easy to learn, to use and which has a high aesthetic value.

The SUS scores were again very clear in favour of MedioVis with a score of 83 compared to 55 for Koala on a scale from 0-100. This difference is highly significant ($p < 0.01$) and confirms the PQ score of Attrakdiff. Altogether the subjective user ratings with our two questionnaires underline the clear performance results once more and lead to the definite statement that MedioVis is superior to KOALA in nearly all aspects of our experiment.

4 Conclusion

MedioVis was developed with the aim to proof the concept of a new user interface paradigm for online library catalogue systems. To reach this goal the design of the system was based on the following principles:

- Design of an **easy to learn** system that allows novice users a quick use without any training.
- Design of an **easy to use** system that supports all information seeking activities in an effective and efficient manner.
- Design of an aesthetical pleasing system offering **joy of use** experiences.
- Simple **formulation of the query** following the well established convention of web search engines, extended by an advanced search.
- Offer a quick and insightful **overview** about all search results to find the “needles in the haystack”.
- Offer the right **amount of information** in the **context** where the users need it.
- Present **different aspects** of interest at the **same time** to compare them or to get more information at a glance.
- Offer possibilities of **restricting** the amount of **information** to selected topics of interest.
- Offer the possibility to **customize the system** reflecting the user’s personal needs (e.g. kind of result presentation, placement or amount of information).

To fulfil all these principles we followed a multiple coordinated view approach, offering the user a simple and advanced search view known from web search engines, a powerful table and a simple graphical view for the presentation of the search results, a title, a selected media and a media location view to present more details about selected titles. Special attention was given to the aesthetic appearance of the system, replacing default widgets by graphical ones and improving the consistency of used colours, font sizes, etc.

The results of our extensive evaluation experiment lead to the conclusion that MedioVis is clearly superior to conventional web-based retrieval system like KOALA (online catalogue system of the University of Konstanz). Our test users completed nearly all tasks significantly faster working with MedioVis. In addition the results of the Attrakdiff questionnaire (measuring the appeal of the system) and the comments made by the users during the experiment (e.g.: “I would have preferred to work on all tasks with MedioVis”, user after working on the second task-set with KOALA) also confirm that users seemed to like the appearance of MedioVis and were comfortable working with it although they had never seen or used it before. We will continue to work with communication designers in order to further improve the visual appearance of MedioVis and talk to library users in order to find out which features are still missing but would be welcome.

If you take a look at the range of products available for internet search, list-based approaches are clearly in the lead. Nowadays users even expect Google-like interfaces and result presentation whenever searching information with their PC. Nevertheless our results show that a table based result presentation is superior to list-based

approaches, and that users are able to quickly adapt to our new interface resulting in an immediate enhancement of the time needed to complete typical search tasks. Since the market of internet search systems has become a multi-billion dollar business, innovations are not easy to introduce. In our opinion, libraries offer the perfect sub-market to slowly establish alternative forms like our table-based MedioVis and thus could lead to a paradigm shift.

References

1. Borgman, C.: Why are online catalogs still hard to use? *Journal of the American Society for Information Science* 47(7) (1996) 493–503
2. Reiterer, H., Tullius, G., Mann, T. M.: INSYDER: a content-based visual-information-seeking system for the Web. Springer-Verlag GmbH, *International Journal on Digital Libraries* (2005). <http://www.springeronline.com/sgw/cda/frontpage/0,11855,5-148-70-1118744-0,00.html>, 2005/02/25
3. Klein, P., Müller, F., Reiterer, H., Eibl, M.: Visual Information Retrieval with the Supertable + Scatterplot. In: Sixth International Conference on Information Visualisation IV02 (2002) 70-75
4. Klein, P., Reiterer, H., Müller, F., Limbach, T.: Metadata Visualization with VisMeB. IV03, 7th International Conference on Information Visualization, London (2003)
5. James, W.: What is an Emotion? *Mind*, 9 (1884) 188-205. <http://psychclassics.yorku.ca/James/emotion.htm>, 2005/02/25
6. Kurosu, M., Kashimura, K.: Apparent usability vs. inherent usability. In: Companion of CHI '95 Conference on Computer Human Interaction (1995) 292-293
7. Glass, B.: Swept Away in a Sea of Evolution: New Challenges and Opportunities for Usability Professionals. *Software-Ergonomie '97. Usability Engineering: Integration von Mensch-Computer-Interaktion und Software-Entwicklung*, R. Liskowsky, B.M. Velichkovsky, and W. Wüschmann, eds., B.G. Teubner, Stuttgart, Germany (1997) 17-26
8. Tractinsky, N.: Towards the Study of Aesthetics in Information Technology, 25th Annual International Conference on Information Systems, Washington, DC, December 12-15 (2004). http://www.ise.bgu.ac.il/faculty/noam/papers/04_nt_icis.pdf, 2005/02/25
9. Jordan, P. W.: *Designing Pleasurable Products – An Introduction To The New Human Factors*. Taylor & Francis, London (2000)
10. Hassenzahl M., Platz A., Burmester M., Lehner K.: Hedonic and Ergonomic Quality Aspects Determine a Software's Appeal. *Proceedings of CHI'2000*. ACM, The Hague (2000)
11. Ngo., D. C. L., Teo, L. S., Byrne, J. G.: A Mathematical Theory of Interface Aesthetics. *Visual Mathematics* N 1 (2001). <http://www.mi.sanu.ac.yu/vismath/ngo>, 2005/02/25
12. Bálek, M., Nešetřil, J.: *Towards Mathematical Aesthetics*. ITI-Series 2004, Institute for Theoretical Computer Science (2004)
13. North, C., Shneiderman, B.: *A Taxonomy of Multiple-Window Coordination*. University of Maryland, Computer Science Dept, Technical Report #3854 (1997)
14. North, C., Shneiderman, B.: Snap-together Visualization: Can Users Construct and Operate Coordinated Visualizations?. *Int. J. Human-Computer Studies* 53 (2000) 715-739
15. Shneiderman, B.: The Eyes have it: A Task by Data Type Taxonomy. In: *Proc. of IEEE Symp. Visual Languages* 96 (1996) 336-343
16. Wake, W., Fox, E.: SortTables: A Browser for a Digital Library. In: *Proc. 4th Int. Conf. on Information and Knowledge Management, CIKM'95*, Baltimore, MD (1995) 175-181

17. Buering T., Reiterer H.: ZuiScat - Querying and Visualizing Information Spaces on Personal Digital Assistants. In MobileHCI 2005. Human Computer Interaction with Mobile Devices and Services, ACM Press (2005)
18. Brooke, J.: SUS: A Quick and Dirty Usability Scale. In: Jordan, P.W., Thomas, B., Weerdmeester, B.A. & McClelland, I.L. (Eds.), Usability Evaluation in Industry. London: Taylor & Francis (1996)
19. Hassenzahl, M., Burmester, M., & Koller, F.: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In J.Ziegler & G. Szwillus (Eds.), Mensch & Computer 2003. Interaktion in Bewegung (2003) 187-196
20. Shneiderman, Ben: Designing the User Interface. Strategies for Effective Human-Computer Interaction. 3rd edition Reading, MA, Addison-Wesley (1998)
21. Techsmith Morae 1.1: The Digital Solution for Usability Analysis. <http://www.techsmith.com/products/morae/default.asp>, 2005/02/25

Effectiveness of Implicit Rating Data on Characterizing Users in Complex Information Systems

Seonho Kim, Uma Murthy, Kapil Ahuja, Sandi Vasile, and Edward A. Fox

Department of Computer Science,
Virginia Tech,
Blacksburg, Virginia 24061 USA
{shk, umurthy, kahuja, sandi, fox}@vt.edu

Abstract. Most user focused data mining techniques involve purchase pattern analysis, targeted at strictly-formatted database-like transaction records. Most personalization systems employ explicitly provided user preferences rather than implicit rating data obtained automatically by collecting users' interactions. In this paper, we show that in complex information systems such as digital libraries, implicit rating data can help to characterize users' research and learning interests, and can be used to cluster users into meaningful groups. Thus, in our personalized recommender system based on collaborative filtering, we employ a user tracking system and a user modeling technique to capture and store users' implicit ratings. Also, we describe the effects (on community finding) of using four different types of implicit rating data.

1 Introduction

As two-way World Wide Web services such as blogs, wikis, online journals, online forums, etc. became popular, more people were able to express themselves and play more active roles in online societies [1, 2]. This trend changed WWW users from passive anonymous observers to visible individuals with personalities. Such users, in increasing numbers, are patrons of digital libraries (DLs), e.g., researchers and distance learners. Studying users of DLs is providing opportunities for research on collaborative filtering, personalization, user modeling, and recommender systems. Most such studies consider users' ratings on the information they select, as well as users' preferences – e.g., on research areas, majors, learning topics, or publications – which are entered explicitly [3, 4, 5]. However, obtaining explicit rating data is difficult. Further, terminology associated with the broad topical coverage of most DLs poses serious challenges regarding the identification of users' research and learning interests. Even people with the same research interests express those interests with different terms, while the same terms sometimes represent different research fields. For these reasons, we need other evidence to help distinguish users' research interests, without depending on their written comments. Thus, Nichols [6] and the GroupLens team [7] showed the great potential of implicit rating data when it is combined with existing systems to form a hybrid system. Further, we utilized users' implicit rating data for collaborative filtering in DLs [8]. However, the effectiveness of implicit rating methods still remains unproven. Consequently, we explore user tracking and

implicit ratings in Section 2 and then propose hypotheses about the use of such data in Section 3. Section 4 describes our initial experiments and their results, while Section 5 concludes the discussion and outlines future work.

2 User Tracking and Implicit Ratings

Gonçalves et al. proposed an XML based log standard for DLs [9] which helped pave the way for this study. However, originally it emphasized the interoperability and reusability of logging, based on a minimal DL metamodel. A more detailed characterization and analysis of user societies and scenarios is needed. Accordingly, we developed a user tracking system [8, 10] to collect histories of users' interactions. The user tracking system, which collects and sends all phrases and sentences clicked and typed by a user to a server, is embedded in the interface, so that the user is not aware of its existence. Instead of using HTTP web logs like most DLs, we employed a user modeling technique to record interactions, in XML. Each user model also contains demographic information, personal interests, and similarities with research interest groups in DLs. The most distinctive feature of the model is that it contains a user's implicit ratings along with explicit ratings and statistics, as needed for collaborative filtering and recommendation. This not only increases the completeness, interoperability, and reusability of user model data, but also decreases the complexity of the process. User interests are entered implicitly by using a document clustering algorithm, LINGO [11, 12], and a user tracking interface. Also, sending a query, selecting or skipping an anchor, and expanding a node – all are considered implicit ratings.

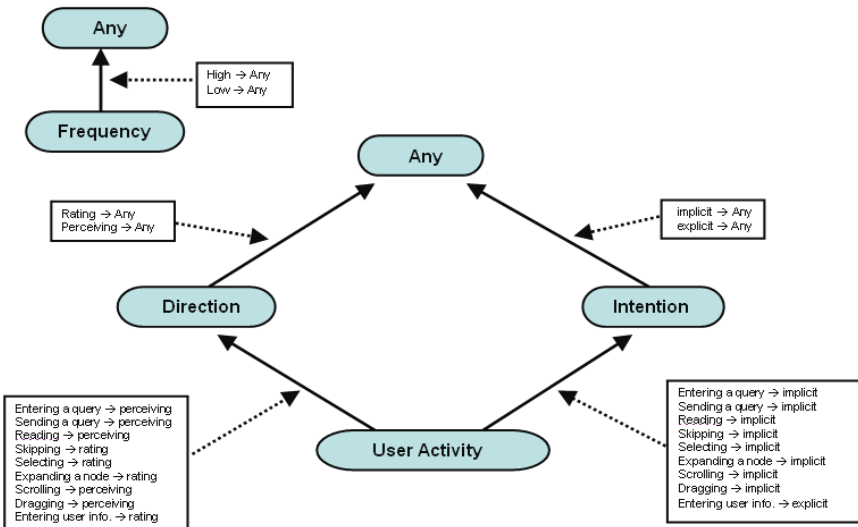


Fig. 1. A DGG for "User Activity" attribute

An implicit rating is captured from a user activity. Figure 1 is our Domain Generalization Graph (DGG) for the user activity attribute in our model; DGGs are more commonly used in connection with data mining targeted on sales or transaction data [13] to represent the comprehensive relations between attributes. Each node in the graph represents a partition of the values that can be used to describe the attributes. The discrete attribute “frequency” is independent of other attributes of user activity. Edges between adjacent nodes describe the generalization relations between the nodes. Each user activity has a direction, where:

- “rating” means the user gives some feedback to the system;
- “perceiving” means the user doesn’t give feedback to the system; and
- regarding intention, “implicit” means the user gives feedback implicitly while “explicit” means feedback is given explicitly.

Thus, sending a query and reading a title are not “rating,” since we don’t give any feedback. However, expanding and skipping a cluster are “rating” – by which we indicate whether the cluster is interesting or not. For an example of intention, note that entering a query is “implicit,” because our purpose is not to characterize ourselves. However, entering user information or preferences is “explicit.”

3 Hypotheses

To assess the effectiveness of implicit rating data for characterizing DL patrons according to their research interests, we developed a special interface for the CITIDEL system [14], part of the NSF-funded National Science Digital Library. Our interface was based on Carrot² [11], coupled with our user tracking system, which together support and record selection of clusters (i.e., the output of the system) [8]. Hence, we test three hypotheses about proper human-computer interaction and document clustering.

1. **H₁**: For any serious user with their own research interests and topics, show repeated (consistent) output for the document collections referred to by the user.
2. **H₂**: For serious users who share common research interests and topics, show overlapped output for the document collections referred to by them.
3. **H₃**: For serious users who don’t share any research interests and topics, show different output for the document collections referred to by them.

4 Experiments and Data Analysis

We collected implicit ratings from 22 students at both the Ph.D. and Master’s level in Computer Science at Virginia Tech; CITIDEL [14] contains documents in the “computing” field. 18 of the students successfully completed the experiment and so were selected to be analyzed for this study. Each subject was asked to perform searches with CITIDEL using 10 queries in his/her (hereafter, “her”) research field and allowed to browse the search results to find interesting documents. All subjects were required to register into the system and so provided explicit preferences.

By session end, each subject had an XML formatted user model in our system. The recommender, which is a software module in charge of collaborative filtering, manages and updates models whenever users log-out. Figure 2 is a simplified sample of a user model. The model consists of four highest level elements (in addition to a log of queries submitted): 1) “userInfo” and “userInterests” (not expanded) are for explicit answers to a questionnaire, 2) “community” is for the communities of the user found by the recommender, 3) “proposed” is for document topics which are shown to the user and skipped, and 4) “selected” is for document topics which are selected or expanded by the user. Therefore, (1) is explicit rating data, (2) reflects computer inference, and (3) and (4) are implicit rating data. Each entry has accompanying statistics (e.g., frequencies, probabilities).

```

<?xml version="1.0" ?>
- <user>
  <userID>seonho</userID>
+ <userInfo> (1)
+ <userInterests>
- <community> (2)
  <member score="0.743">sig001</item>
  <member score="0.510">sig004</item>
  <member score="0.183">sig003</item>
</community>
- <query>
  <item freq="3">Educational Library</item>
  <item freq="2">User modeling</item>
  <item freq="1">Log System</item>
</query>
- <proposed> (3)
  <item freq="3">Curriculum in Computer</item>
  <item freq="3">Distance learning</item>
  <item freq="2">Computer Communication</item>
  <item freq="2">Computer and Computer Education</item>
  <item freq="1">Computer Security</item>
  <item freq="1">Computer Integrated Manufacturing</item>
  <item freq="1">Computer and Public</item>
  <item freq="1">Computer Anxiety</item>
  <item freq="1">Data Parallel</item>
  <item freq="1">IEEE Computer Society</item>
</proposed>
- <selected> (4)
  <item freq="3">Curriculum in Computer</item>
  <item freq="2">Distance learning</item>
  <item freq="2">Computer and Computer Education</item>
  <item freq="1">Computer and Public</item>
  <item freq="1">IEEE Computer Society</item>
</selected>
</user>

```

Fig. 2. An example of user model generated by the recommender

We employed hypothesis testing [15] as follows. Because the data collected from the user tracking system is independent and identically distributed (i.i.d.), we use inference processes to verify hypotheses and estimate properties, starting with HT1.

HT1: Hypothesis testing and confidence intervals for H_1 .

1. H_0 (Null hypothesis of H_1): Mean (μ) of frequency of document topics proposed by the Document Clustering Algorithm are NOT consistent ($\mu_0 = 1$) for a user.

Hypothesis Testing about $H_0 : \mu = \mu_0$ vs. $H_1 : \mu > \mu_0$

2. Conditions: 95% confidence (test size $\alpha = 0.05$), sample size ‘n’ < 25, unknown standard deviation ‘ σ ’, i.i.d. random sample from normal distribution, \rightarrow estimated z-score t-test.
3. Test statistics: sample mean ‘ \bar{y} ’ = 1.1429, sample standard deviation ‘s’ = 0.2277 are observed from the experiment.
4. Rejection Rule is to reject H_0 if $\bar{y} > \mu_0 + z_{\alpha/2} \sigma/\sqrt{n}$
5. From the experiment, $\bar{y} = 1.1429 > \mu_0 + z_{\alpha/2} \sigma/\sqrt{n} = 1.0934$
6. Therefore decision is to Reject H_0 and accept H_1 , 95% Confidence Interval for μ is $1.0297 \leq \mu \leq 1.2561$, and P-value = 0.0039

Although we separated H_2 and H_3 as different hypotheses to emphasize the ideas, they can be understood as identical and can be proven and estimated together by one hypothesis test, with confidence intervals as described below. So, we consider HT2:

HT2: Hypothesis testing and confidence intervals for H_2 .

1. H_0 (Null hypothesis of H_2): A user’s average ratio of overlapped topics with other persons in her groups over her total topics which have been referred, μ_1 , is the same as the average ratio of overlapped topics with other persons out of her groups over her total topics which have been referred, μ_2 .

Hypothesis Testing about $H_0 : \mu_1 = \mu_2$ vs. $H_2 : \mu_1 > \mu_2$

Because a user can belong to multiple groups, population means μ_1 and μ_2 are calculated as in the formulas below, respectively,

$$\mu_1 = \frac{\sum_{k=1}^G \sum_{i=1}^{n_k} \sum_{j=1, j \neq i}^{n_k} O_{i,j}}{\sum_{k=1}^G n_k (n_k - 1)}, \quad \mu_2 = \frac{\sum_{k=1}^G \sum_{i=1}^{n_k} \sum_{j=1, j \notin K}^N O_{i,j}}{\sum_{k=1}^G n_k (N - n_k)}$$

where $O_{i,j}$ is user i’s topic ratio overlapped with user j’s topics over i’s total topics, G is the total number of user groups in the system, n_k is the total number of users in group K, and N is the total number of users in the system. One instance of random variables in this testing, one user’s overlapped topic ratio with other persons in her group and overlapped topic ratio with other persons out of her group, is illustrated in Figure 3.

2. Conditions: 95% confidence (test size $\alpha = 0.05$), two i.i.d. random samples from a normal distribution, for two sample sizes n_1 and n_2 , $n_1 = n_2 < 25$, standard deviations of each sample σ_1 and σ_2 are unknown \rightarrow two-sample Welch t-test.
3. Test statistics: Welch score ‘ w_0 ’ = $\frac{(\bar{y}_1 - \bar{y}_2)}{(s_1^2/n_1 + s_2^2/n_2)^{1/2}}$, where \bar{y}_1, \bar{y}_2 are sample means of each sample and s_1, s_2 are sample standard deviations of each sample.
4. Rejection Rule is to reject H_0 if the $w_0 > t_{df_s, \alpha}$ where t refers to the t-cutoff of the t-distribution table, and df_s is the Satterthwaite’s degree of freedom approximation [15] which is calculated by

$$df_s = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}}$$

5. From the experiment, $\bar{y}_1 = 0.103$, $\bar{y}_2 = 0.0215$, $df_s = 16.2$ and $w_0 = 4.64 > t_{16.2, 0.05} = 1.745$
6. Therefore decision is to Reject H_0 and accept H_2 , 95% Confidence Intervals for μ_1 , μ_2 and $\mu_1 - \mu_2$ are $0.0659 \leq \mu_1 \leq 0.1402$, $0.0183 \leq \mu_2 \leq 0.0247$ and $0.0468 \leq \mu_1 - \mu_2 \leq 0.1163$, respectively, and P-value = 0.0003

Although the confidence intervals found in HT1 and HT2 are broad because of a relatively small set of participants, all values in the intervals still prove our hypotheses. Therefore, supported by HT1 and HT2, we argue statistically that our hypotheses are correct. We conclude that each DL user will engage in consistent activities in response to consistent output of the DLs according to their research interests and learning topics. Also, DL users who share common research interests and learning topics will share the same output from the DLs as well. Therefore, we conclude that using implicit rating data is highly effective in characterizing users, according to our experiment.

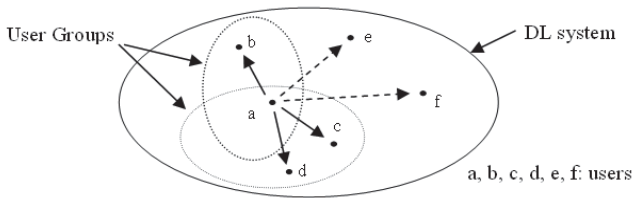


Fig. 3. An instance of the random variables for user a 's in-group overlappings and out-group overlappings, in the Hypothesis Testing HT2. All overlapping ratios are directed. \overline{ab} means the overlapping ratio from user a to user b . Because the ratio is the number of topics overlapped over the total number of topics in her user model, $\overline{ab} \neq \overline{ba}$. In this case, the in-group overlapping ratio of user a is the average of \overline{ab} , \overline{ac} , and \overline{ad} , and out-group overlapping ratio is the average of \overline{ae} and \overline{af} .

Studies on the effect of different types of data on the performance of user cluster mining have highlighted a basic problem caused by the variety of academic terms, as we mentioned in the introduction section. However, we can explore user cluster mining more objectively, because we can obtain user groups without depending on user's subjective answers to questionnaires about their research interests or preferences [8]. We conducted an ANOVA test to compare the effectiveness of four different user rating data types on the performance of user cluster mining by using implicit rating data and user groups collected from experiments in [8].

Figure 4 shows the result; ANOVA statistics $F(3, 64) = 4.86$, $p\text{-value} = 0.0042$ and the least significant difference (LSD) = 1.7531. Topics mean noun phrases generated by LINGO. Terms indicate single nouns contained in the original documents, queries, and topics. Although we gained a relatively large LSD because of the small number of participants, we still found statistical significance in this test. Figure 4 shows that the test using proposed terms performs significantly worse.

Except for the test using the proposed terms, the other three tests that use selected topics, proposed topics, and selected terms don't show statistically significant differences from each other even though the test using proposed document topics shows slightly higher performance. We believe that this is because using proposed terms causes too sensitive overlapping both in the in-group testing and out-group testing (to distinguish proper relations between users). This leads us to conclude that term-frequency based approaches to user cluster mining are not as efficient as document-topic based approaches using user rating and document clustering.

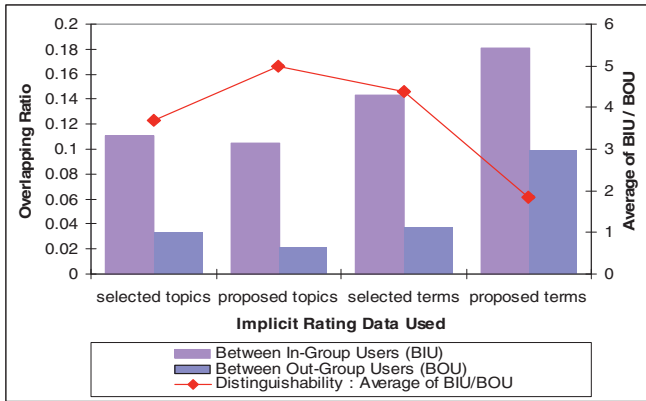


Fig. 4. Effects of implicit data type used, on the average topic overlapping ratios, between in-group users and between out-group users

5 Conclusions and Future Work

We designed and implemented a user modeling and user tracking system for a DL to capture and maintain a user's ratings and preferences. We then proved that implicit rating data in a complex information system is highly related with the user's research interests, learning topics, and preferences – through two statistical hypothesis tests. The test results support the claim that implicit ratings are good information for studies on user analysis, personalization, collaborative filtering, and recommending. Finally, we tested the effect of different types of rating data on the performance of user cluster mining and found that using proposed terms performs worst, because of sensitive overlapping ratio of appearance. From this test we conclude that user's activities of selecting something on the interface, and extracting document topics of returned documents from searches with a document clustering algorithm, represent the user's characteristics. These results are more meaningful in complex information systems like digital libraries because such systems have dynamic contents and sparse rating data, and thus implicit rating data is more feasible to collect than explicit rating data.

Future work will include more advanced data mining techniques using implicit rating data and a wider deployment of a visualization front-end for CITIDEL [16].

Acknowledgements

We thank the: people and organizations working on CITIDEL, student participants in our experiments, developers of the LINGO algorithm, and developers of the CitiViz [16] visualization tool. Thanks go to the National Science Foundation for support of grants NSF DUE-0121679, DUE-0121741, IIS-0307867, and NSF IIS-0325579.

References

1. Ravi Kumar, Jasmine Novak, Prabhakar Raghavan and Andrew Tomkins: Structure and Evolution of Blogspace. In *Communications of the ACM*, Vol. 47, No. 12, December 2004, 35-39
2. Cass R. Sunstein: Democracy and Filtering. In *Communications of the ACM*, Vol. 47, No. 12, December 2004, 57-59
3. Thomas W. Malone, Kenneth R. Grant, Franklyn A. Turbak, Stephen A. Brobst, and Michael D. Cohen: Intelligent information sharing systems. In *Communications of the ACM*, Vol. 30, No. 5, 1987, 390-402
4. David M. Nichols, Duncan Pemberton, Salah Dalhoumi, Omar Larouk, Clair Belisle and Michael B. Twidale: DEBORA: Developing an Interface to Support Collaboration in a Digital Library. In *Proceedings of the Fourth European Conference on Research and Advanced Technology for Digital Libraries (ECDL '00)*, Lisbon Portugal, September 2000, 239-248
5. Kai Yu, Anton Schwaighofer, Volker Tresp, Xiaowei Xu and Hans-Peter Kriegel: Probabilistic Memory-based Collaborative Filtering, *IEEE Transactions on Knowledge and Data Engineering*, 2004, Vol. 16, No. 1, 56-69
6. David M. Nichols: Implicit Rating and Filtering. In *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*, Budapest Hungary, November 1997, 31-36
7. Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon and John Riedl, GroupLens: Applying Collaborative Filtering to Usenet News. In *Communications of the ACM*, Vol. 40, No. 3, 1997, 77-87
8. Seonho Kim and Edward A. Fox: Interest-based User Grouping Model for Collaborative Filtering in Digital Libraries. In *Proceedings of the 7th International Conference on Asian Digital Libraries (ICADL'04)*, Shanghai, China, December 2004. In *Springer Lecture Notes in Computer Science* 3334, 533-542
9. Marcos André Gonçalves, Ming Luo, Rao Shen, Mir Farooq and Edward A. Fox: An XML Log Standard and Tools for Digital Library Logging Analysis. In *Proceedings of Sixth European Conference on Research and Advanced Technology for Digital Libraries*, Rome, Italy, September, 2002, 16-18
10. Kapil Ahuja, Uma Murthy and Sandi Vasile: In Virginia Tech report, available at <http://collab.dlib.vt.edu/runwiki/wiki.pl?MmProjectUserMod>, 2004
11. Carrot² Project, A Research Framework for experimenting with automated querying of various data sources, processing search results and visualization, available at <http://www.cs.put.poznan.pl/dweiss/carrot/>, 2005
12. Stanisław Osiński and Dawid Weiss: Conceptual Clustering Using Lingo Algorithm: Evaluation on Open Directory Project Data. In *Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference*, Zakopane Poland, 2004, 369-378

13. Aaron Ceglar, John Roddick and Paul Calder: Guiding Knowledge Discovery Through Interactive Data Mining, *Managing Data Mining Technologies in Organizations: Techniques and Applications*, Idea Group Publishing, 2003, 45-87
14. CITIDEL: Available at <http://www.citidel.org/>, 2005
15. R. Lyman Ott and Michael Longnecker: *An Introduction to Statistical Methods and Data Analysis*, Fifth Edition, Wadsworth Group, 2001
16. Saverio Perugini, Kathleen McDevitt, Ryan Richardson, Manuel Perez-Quñones, Rao Shen, Naren Ramakrishnan, Chris Williams and Edward A. Fox: Enhancing Usability in CITIDEL: Multimodal, Multilingual, and Interactive Visualization Interfaces, in *Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries (JCDL '04)*, Tucson Arizona, June 2004, 315-324

Managing Personal Documents with a Digital Library

Imene Jaballah, Sally Jo Cunningham, and Ian H. Witten

Department of Computer Science, University of Waikato,
Private Bag 3105, Hamilton, New Zealand
[ibaj1, sallyjo, ihw]@cs.waikato.ac.nz

Abstract. This paper presents a desktop system for managing personal documents. The documents can be of many types—text, spreadsheets, images, multimedia—and are organized in a personal “digital library”. The interface supports browsing over a wide variety of document metadata, as well as full-text searching. This extensive browsing facility addresses a significant flaw in digital library and file management software, both of which typically provide less support for browsing than for searching, and support relatively inflexible browsing methods. Three separate usability studies of a prototype—an expert evaluation, a learnability evaluation, and a diary study—were conducted to suggest design refinements, which were then incorporated into the final system.

1 Introduction

For nearly four decades, personal computers have been using the desktop and folder system metaphors. These metaphors use a hierarchical structure to allow users to store and access documents in their personal file space. This approach worked quite well as long as the number of items was in the range of hundreds, but it does not scale to thousands or ten of thousands of files. The challenge has shifted from deciding what to keep, to finding specific documents when they are needed [11]. The result is too many folders for the users to organize, remember and access when seeking information within their personal collection of files.

Currently, the ability of users to browse and search through their files is limited by conventional hierarchical structure and location-based browsing. Strict hierarchies map poorly to user needs. The restriction that a document can appear only in one place at any given time, and using document locations as the principle of organization structure, forces computer users to create strict categorizations for their files. Previous studies of filing practices of computer users have suggested that such restrictions to a hierarchical structure can hinder rather than help users in quickly finding desired documents. Providing other means for browsing would give users more flexibility when looking for information in their personal electronic collections.

The system described here is an attempt to provide better support for information seeking within personal information collections, through a Desktop Digital Library (DDL). Although the DDL supports both searching and browsing, the emphasis is on browsing based on document properties and document contents—that is, those

features of a document that are meaningful to users. The implementation is based on a digital library solution, Greenstone, and uses a metadata-based approach.

Previous attempts to provide different and better ways of browsing include Tree-maps, which present the relationships between two dimensional images and their representation in hierarchical tree structures [10]. Alternatively, Boardman [1] proposed a technique to organize resources at the workspace level, by sharing one hierarchy between all applications. Freeman and Gelernter have proposed the Lifestreams project which provides a complete file management system based on time stamps [4]. Lifestreams generates a visualization of documents organized by time, forming a personal history. However, all these solutions escape one fixed organizational scheme, the folder-hierarchy, to fall into another, such as the time-line. Users need not be restricted to two dimensional representations, hierarchical structures or temporal organizations.

The closest related work to this project is UpLib [5]—a personal digital library system. The system could be accessed through an active agent via a Web interface (similar to the Greenstone’s collection access method). In addition, like Greenstone it provides a full-text index of the collection documents. The system uses both document images and document text; however, it adopts an image-centric approach that produces a visual interface based on page images. Compared to the work presented here, UpLib handles smaller collections of documents than the Desktop Digital Library system. DDL aims to support very large scale collections that reflect the number of documents in actual personal information collections. Unlike the image-focused approach embraced by UpLib, DDL provides a variety of browsing methods. Documents images form an interesting navigation technique; however, users might want to navigate using other attributes.

The Greenstone digital library construction software that underpins the DDL is described in Section 2, and the DDL interfaces and sample interactions are presented in Section 3. The DDL system underwent two rounds of usability studies—an expert evaluation and a ‘learnability’ study with prospective users—and the results of these studies were used to modify the DDL design to improve its usability. Usability of the modified prototype was further evaluated through a diary study. The results of these studies are described in Section 4.

2 Implementation

The term “digital library” is used to describe the use of digital technologies to acquire, store, preserve, and provide access to information and material originally published in digital form or digitized from existing print, audio-visual or other formats [12].

The Desktop Digital Library was implemented using the Greenstone digital library software, described in Section 2.1. Although Greenstone supports storage, searching, and browsing of document collections, it is not ideally suited to organizing a personal document collection—Greenstone’s drawbacks in this regard are described in Sections 2.3, 4.1, and 4.2.

2.1 Greenstone Overview

The Greenstone digital library software (www.greenstone.org) is a comprehensive system for the construction and presentation of document collections [12]. Greenstone was created by the New Zealand Digital Library research group (<http://www.nzdl.org>) at the University of Waikato (Hamilton, New Zealand).

Collections built by Greenstone become maintainable, searchable, and browsable. They can be large: Greenstone collections can comprise millions of documents and require gigabytes of storage. Documents in a collection can include text, images, sound, and multimedia. Greenstone facilitates the process of indexing files to make them fully searchable, by associating metadata stored in the file system and by producing browsing indexes that reflect multiple hierarchies, thereby allowing collection creators to tailor collection presentation to the needs of users.

The Greenstone system is public, extensible, and well documented. It is issued under the Gnu public license and users are invited to contribute modifications and enhancements. In addition, the system is multilingual, as it was used to construct collections in different languages. This supports the ability to extend the Desktop Digital Library interface into different languages. Moreover, Greenstone works under different platforms and only small proportions of the Desktop Digital Library system needs to be upgraded for the application to support multiple platforms.

2.2 Greenstone Browsing Facilities

The browsing facilities provided by Greenstone are supported via structures generated by software ‘classifiers’. These browsing structures are generated automatically from the metadata associated with each documents. Currently, there are five main types of classifiers provided by Greenstone: the list classifier, which produces an alphabetic display of selected metadata (for example, document titles); the alphabetic list classifier, which splits the metadata up into alphabetic groups for ease of browsing; the date classifier, which groups documents by date metadata (for example, date of publication, or date of file creation); the hierarchic classifier, which can display hierarchic categorizations of documents such as the Library of Congress Classification System or the Dewey Decimal system; and the key-phrase classifier Phind [8] that automatically extracts keyphrases and supports the user in browsing and searching by keyphrase.

2.3 Creating and Maintaining a Greenstone Collection

The Greenstone Librarian Interface is intended for use primarily by digital librarians crafting large-scale public collections. Collections are organized and built on a local machine. In keeping with the needs of its primary users—digital librarians—the Librarian Interface provides a rich set of options for adding and editing document metadata and specifying interface details. For lay users interacting with their personal documents, the Librarian Interface facilities are dizzyingly complex.



Fig. 1. Desktop DDL icons

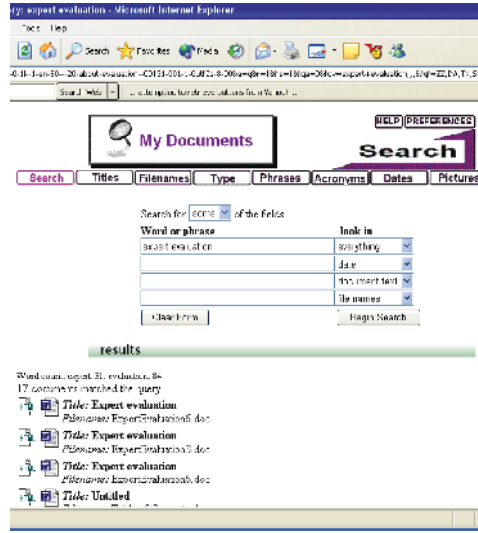


Fig. 2. Searching in the DDL

3 The Personal Digital Library Desktop System

The Desktop Digital Library is designed to assist users in carrying out their browsing tasks. Currently, the ability of users to browse through their personal file space is limited to folder-based access. Documents can only be browsed with respect to their location in the file system. The Desktop Digital Library system, however, provides computer users with additional browsing methods. For instance, it enables users to navigate personal documents by their type, titles, filenames, date of modification, and so on. This section will illustrate how the Desktop Digital Library is used and present the different browsing options available.

3.1 Creating and Organizing a Personal Collection

The Desktop Digital Library is designed to allow the user to interact with the application without any prior knowledge of Greenstone and its infrastructure. The following steps need to be followed in order to create a collection of documents that is managed by the DDL (Figure 1):

- 1) The user needs to select a set of documents—selections could be in the form of a single document, multiple documents, a single folder or multiple folders.
- 2) After deciding what documents to select, the user drags the selection into the “Drag Documents” icon.

- 3) Then, users need to double click on the “Organize My Documents” icon—which will display a command prompt window showing all the documents being processed.

Users are immediately able to view their personal collection of files by clicking on the “View My Documents” icon.

3.2 Searching a Personal Document Collection

As well as browsing, the DDL, the system also offers searching facilities. Using the Greenstone capabilities, the system offers full-text searching of the documents’ text in the collection. The interface also allows the user to search filenames, date of modification, document titles, document type, and a combination of these options. In Figure 2, a user is viewing the results of a search for the words “expert evaluation”.

3.3 Browsing a Personal Document Collection

DDL currently supports eight browsing mechanisms: by Title, Type, Phrase, Acronym, Date, Picture, and Folder (Figure 3):

- The *Titles* interface is based on the Greenstone alphabetic list classifier (not shown in Figure 3). Title metadata is automatically extracted as the first few lines of text in a document, similar to the Microsoft Word convention of suggesting a filename for a newly created document from its initial text.
- The *Filenames* structure displays documents in alphabetic order by file name. It is also based on the Greenstone alphabetic list classifier.
- The *Type* interface allows the user to view and browse their personal documents grouped by document type. Each type is displayed with the appropriate icon and the total number of documents of that type in the collection. Clicking on a type displays those associated documents.
- The *Pictures* interface displays thumbnails of image files within a collection, sorted by file name.
- The *Folders* interface allows users to browse according to the file paths for documents within the collection. This view is similar to that provided by the user’s operating system, but is not cluttered with files outside the collection.
- The *Dates* browsing scheme sorts documents by their latest modification date. Dates are organized by year, and further divided into ranges of months (not shown in Figure 3).
- *Phrase* browsing is based on the Phind classifier [8], which automatically identifies noun phrases within the collection (not shown in Figure 3). These phrases can be searched and browsed, and by selecting a phrase a user can drill down to its context within the documents.
- The *Acronyms* interface allows users to view acronyms occurring in the text of collection documents (not shown in Figure 3). A compression-based algorithm automatically identifies acronyms and associates each one with its likely expansion [13].

4 Usability Evaluation

Two usability studies were conducted on the initial DDL design: an expert evaluation by usability specialists (Section 4.1) and a ‘learnability’ study whose participants were prospective users of DDL (Section 4.2). A nearly fully functional prototype was created for these studies, so that participants could gain a feeling for the interactions possible with the system rather than restricting their assessments to interface issues as visible through, for example, a paper prototype. The findings from these two studies were used to refine the initial DDL interface. A third usability analysis, a diary study, was conducted to examine the usability in real world contexts of this revised DDL prototype (Section 4.3).

4.1 Expert Evaluation

The first study was an expert evaluation—specifically, a heuristic evaluation. In an expert evaluation, two or more usability specialists apply their expertise in human factors to independently evaluate a system. The evaluators examine the interface and judge its compliance with recognized usability principles (the heuristics). Skilled specialists can produce high-quality results in a limited time because the method does not involve the detailed scripting or time-consuming participant recruiting of laboratory usability testing. Thus this type of evaluation is especially valuable when time and resources are short, or as an initial overview of system usability.

Two local experts in both usability and digital libraries/information management participated in this study. Previous research indicates that while ‘single’ experts are likely to find over 40% of usability problems, ‘double experts’—those with “expertise in both usability in general and the kind of interface being evaluated”—are likely to find a higher proportion of the problems [7]. The experts each spent an hour interacting with the initial prototype of the DDL. At the beginning of each session the evaluator was given a brief introduction to the DDL through a demonstration of the typical steps a user would follow to interact with the system. Each session was video recorded for later analysis, and a researcher facilitated the sessions, answering questions as they arose.

This study uncovered a number of usability problems, both minor—for example, small inconsistencies in font or icon between the browsing facilities—and major. Many of the usability issues stemmed from what in hindsight is excessive adherence to the ‘library’ metaphor of the underlying Greenstone implementation. For example, a label in the initial search facility allowed the user to specify that a search term could match in “all fields”—where “fields” is a term familiar in the online library context, but not when looking through the contents of one’s own file space.

A more fundamental problem stemming from the library metaphor lay in Greenstone’s view of a document as being potentially part of many different collections. When a collection is built with Greenstone, it generates a copy of each document and stores the copy in the collection index; this copy is viewed when a library user searches or browses to locate a particular document, and any changes made to the copy are not saved to the original location of the document. This architecture is sensible in a digital library, where the expectation is that users will be

reading rather than modifying documents, and where it could be catastrophic to have a user's casual annotations to a document ripple through all collections containing it.

In a personal document management, however, users naturally expect that after locating a document through the DDL search or browse facilities, clicking on the document will open the document itself—and not a mere copy. As one of the evaluators noted, this is crucial because, *“Usually when users search or browse for documents, they want to perform a further action in relation to the document such as editing, modifying, deleting and so on.”* A more suitable model to follow here is the standard folder system, where clicking on a file allows users to work directly on the selected document. To resolve this problem, DDL was modified to attach the original path of each document to the document representation within DDL as metadata.

Another area in which the library metaphor poses usability issues is the use of multiple icons for the DDL: specifically, the “Drag files” and “Organize My Documents” icons (Figure 1). In a physical or digital library, there is usually a distinction between selecting documents (the acquisitions process) and indexing or organizing them (creating the collection and its interface). Within a personal document collection, however, this distinction is artificial—the user should not be forced to think within the library paradigm, but rather should be allowed to concentrate on their tasks with minimal interruption by a need to manage their documents.

Within a library, the acquisitions process selects a subset of the potentially many available documents to include in a collection. For a personal document collection, acquisitions should be automatic, with any document created or saved in the user's file space automatically added to the DDL; the user has essentially decided that the document is relevant to the personal collection by creating or downloading it. Similarly, users will wish to always interact with the latest version of their personal collection; the DDL should automatically re-build its indexes whenever new files are added, rather than requiring a separate “Organize My Documents” stage. Unfortunately, creating a single-stage acquisitions/build or an automated document addition facility remains a direction for future work.

4.2 Learnability Evaluation

The second usability study focused primarily on ‘learnability’: the extent to which a user can get started with a system and use it appropriately without first undergoing training [7]. A high rate of learnability is crucial for software acceptance by users. Given the reluctance of users to consult manuals or help files, a system that can be immediately useful will be more likely to see future use.

As recommended by [6], this was a small-scale study involving six participants; the usability research literature indicates that using more than five or six participants does not necessarily gain significantly greater insight into usability issues for a system—instead, the same problems tend to be identified again and again.

Assessment of learnability includes studying system predictability—that is, the ability of users to predict system reactions [2]. Participants were asked to predict what would happen if they clicked on buttons or filled in text boxes in DDL; they were then asked to interact with these searching and browsing facilities of DDL and to comment on whether or not they achieved the predicted response. Participants were

also invited to comment on the interface design in general, and to discuss any aspect that they found confusing, unusual, or difficult to understand. Participant sessions lasted between 1 1/2 and 2 hours, and were video-recorded for later analysis.

Many—but not all—of the interface usability problems identified by these participants had also been noted by the expert evaluation. This high degree of overlap between the two studies is encouraging, as this is evidence that the experts were indeed able to tune in to the sorts of difficulties that potential DDL users might experience. The strongest—negative—reaction was to the Phrase and Acronym browsers. The purpose of the Phrase browser as distinct from a keyword/phrase search mechanism was not clear to the users, and most users could not even recall the meaning of the word ‘acronym’, let alone imagine a scenario in which they would wish to search or browse for documents containing specific acronyms. The case of the phrase browser is particularly interesting, as an earlier usability study had concluded that participants in the study liked this scheme and believed that it would be useful [3]. However, it was determined to be suitable for supporting exploratory tasks rather than at supporting more targeted searching or browsing. While these two browsing facilities may be useful in exploring a digital library whose contents are novel to a user, they are less useful in managing a personal collection where the user is likely to be familiar with the significant phrases and acronyms contained in the documents.

These issues aside, the participants found that most features of the interface were self-explanatory or could be induced through brief experimentation with DDL. The simplicity of the interface was appreciated by most participants.

4.3 Diary Study

The third study was a “diary study” examination of the usability of the usability of the DDL prototype, as refined through insights gained in the first two studies. Participants recorded their daily interactions with a system on preprinted log forms (diaries) [9]. This type of study provides insights into the use and usefulness of a system in real world contexts, over a more extensive period of time than is possible in laboratory experiments.

Six participants took part over a one week period. Corresponding with the target population for the system, the participants were computer users with moderate to advanced computer skills. They used computers on a daily basis, to manage large amounts of electronic information. For the duration of the study, participants were requested to give preference to DDL whenever they need to browse through their personal file space, and to record their interactions on at least a daily basis. Participants were also asked to fill in a concluding questionnaire about their personal experience with the system and attend to a debriefing interview to discuss their recorded comments in the workbook.

Users reported that the application gave them the opportunity to explore their personal documents in a different way. Navigating through their documents using different browsers provided users with the ability to see documents that they have forgotten about or have misplaced and thought they have lost them. One of the participants said “*I have totally forgotten about this document*”. Another one stated “*This picture here... I never thought I still have it*”.

A special interest was shown by most of the study participants in the *Pictures* browser. One of the participants noted that, *“It is nice to be able to see all these photos listed in one place... regardless of what folder they belong to”*. Another participant made the comment, *“... thumbnails allow me to have a quick look to decide which ones [image documents] I would like to go ahead and open”*.

One participant complimented the concept of being able to drag documents from any arbitrary location on the file space to the application to organize and process. This participant dragged documents from a USB key and then clicked the “Organize My Documents” icon. They were able to browse these documents after being included in the application. It was mentioned that even when the USB key is not plugged in the computer, it is still possible to navigate through these documents. However, one problem arises; when the USB key is unplugged the document cannot be retrieved—when the document is clicked an error message is displayed because no link can be provided to a document that does not exist.

Overall, however, the participants reported that they preferred the Windows folder system to the DDL. The DDL presents documents differently from the folders scheme—in particular the concept of presenting documents grouped by alternative methods other than their location. Users are familiar with folders and need to spend some time before becoming familiar with the application mode of presentation. Furthermore, the application needs to be upgraded to accommodate the sorting capabilities that folders provide—being able to sort files by documents’ properties.

Further, the majority of the diary study participants found the action of dragging documents and clicking on icons to be prohibitively expensive. Despite the steps taken to simplify the users’ communication load with the application and ensure that they don’t have to know about Greenstone and its internal structure, they expect an effortless and lighter style of interaction. Documents should be automatically organized and processed by the application without the user having to click on “Organize My Documents” icon. As one of the participants commented, *“I don’t have to do that [clicking on icons] when looking at my files using the Windows Explorer [the folder system]”*.

Perhaps the greatest barrier to the DDL replacing a file management system is that it does not support the deletion of documents. This lack of a deletion facility is Greenstone legacy. In Greenstone’s original domain—maintaining a sizeable public document collection—documents are rarely, if ever, deleted; public digital libraries tend to grow, not reduce in size. If a document must be removed from a digital library, then the deletion is effected by removing the file from the document set prior to a ‘rebuild’ of the entire digital library. Similarly, if a document is modified then the Greenstone indexes are updated by removing the document from the collection’s files, adding the modified version, and re-building (re-indexing) the collection. The situation is radically different in a personal document collection, where documents are modified and deleted on a daily basis.

5 Conclusions

It may seem to stretch the meaning of the term to view the collection of files on one’s own personal computer as a “digital library.” However, although today their importance has long been eclipsed by large institutional and national collections,

personal libraries have a venerable history. For example, on his death in 1661 the renowned Irish man of letters Archbishop Ussher had a personal collection of 10,000 books—which could well have been Ireland’s largest library at the time.

This paper has explored the use of a standard digital library software system, Greenstone, to organize one’s own personal file space. In Greenstone each collection is designed individually by determining what searching and browsing facilities it should provide to the user, and deciding on what the pages it generates should look like. In principle, producing DDL, the Desktop Digital Library, simply amounted to creating a suitable collection design and installing it on the target computer.

However, things were not quite as easy as that. To provide a suitable interface it was necessary to augment some aspects of Greenstone with specially-tailored facilities. These included

- the ability to drag documents directly into the collection
- harvesting metadata such as file name, type, creation date, last modified date, etc.
- shortcuts for organizing documents and viewing them by browsing the collection
- a new way of displaying hierarchical metadata as file hierarchies
- altering some interface terms from library jargon to file space terminology
- storing the original path of each document as metadata so as to give users direct access to the document rather than a library copy of it

The system design was honed by two rounds of user studies: an expert evaluation with two interface expert, following which the system was improved, and a learnability study with six prospective users, following which it was improved further. This proved a valuable methodology, yielding a large volume of high-quality design feedback from only a few subjects. Although some duplicate points arose from the two groups, each contributed its own very different perspective. A third, diary study of the refined prototype gave further insight into the DDL’s usability in real world conditions, over a more extended period of time.

The use of a standard digital library system had some disadvantages. Two notable ones stem from different notions of “immediacy” in a library context vis a vis a personal computer context. Most libraries can tolerate a lag of a day or two between when a new document is received and when it appears in the collection. And most library browsers can tolerate the network and browser delays induced by using a shared system over the World-Wide Web. Personal computer users, however, rightly expect more. In particular, a Web browser is a cumbersome way of interacting with a locally-running software system.

Personal digital libraries differ from personal library collections of old in the amount of effort that their owners are prepared to expend in organizing them. Users want to be able to drag files into the collection having to add metadata to them manually. A certain amount of metadata (filename, type, modification date, etc) can be gleaned from the operating system; other information (title, acronyms, phrases) can be extracted from the document itself, if it is textual. This project has shown that this automatically harvested metadata is enough to provide a rich and useful browsing structure within a standard digital library software application.

The expected lifecycle of documents also differs between personal and public document collections. In a private collection, documents are volatile; they are created, modified, and tossed away, sometimes over very short time periods. In a public digital library, documents are typically not modified or deleted from the collection—

the collection tends to grow monotonically. The facilities to delete or modify documents in Greenstone are ponderous, forcing the collection maintainer to remove documents from the Greenstone file space and re-index the collection.

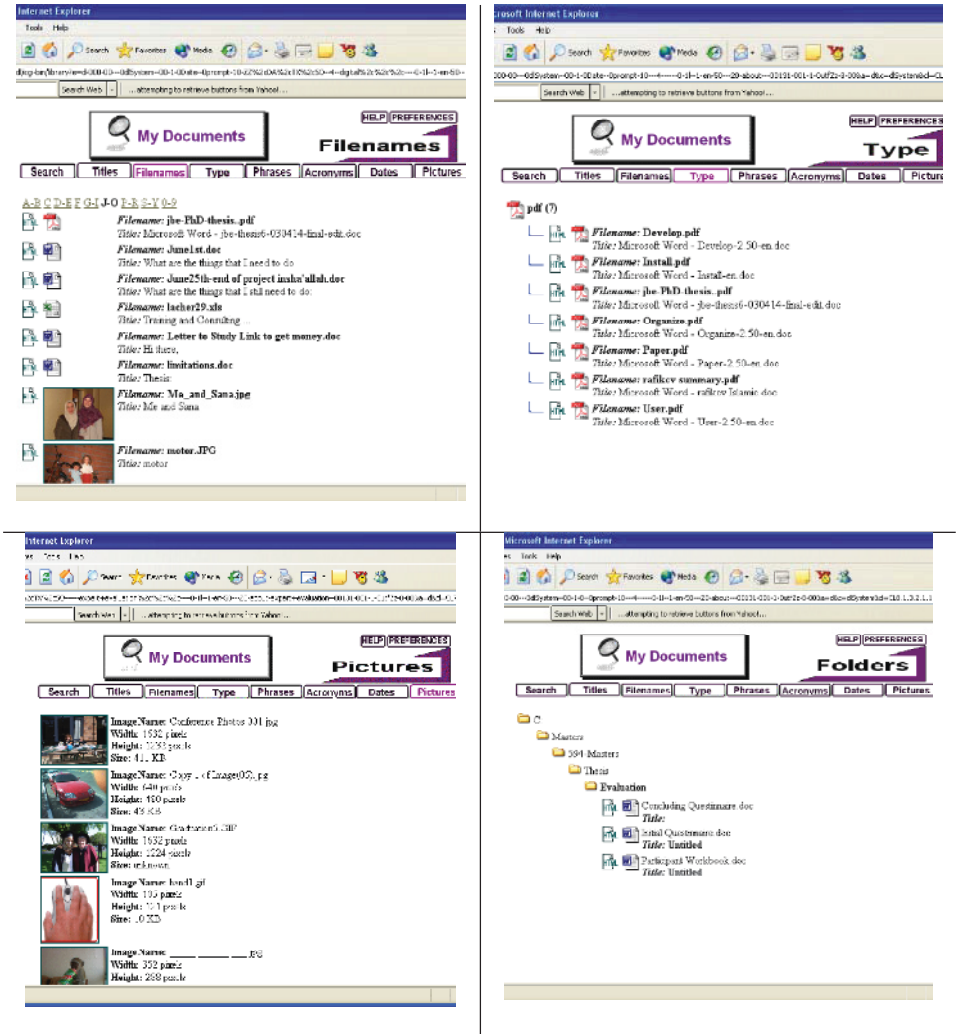


Fig. 3. Browsing interfaces for the Desktop Digital Library

References

- [1] Boardman, R. Multiple hierarchies in user workspaces. In *Proceedings of CHI Conference on Human Factors in Computing Systems*, (Seattle, WA). ACM Press, 2001.
- [2] Dix, A., Finlay, J., Abowd, G., and Beale, R., *Human-Computer Interaction*. 2 ed. 1997, Glasgow: Prentice-Hall.

- [3] Edgar, K.D., Nichols, D.M., Paynter, G.W., Thomson, K., and Witten, I.H. A user evaluation of hierarchical phrase browsing. In *Proceedings of European Conference on Digital Libraries*, (Trondheim). 2003, 313-324.
- [4] Freeman, E. and Gelernter, D., Lifestreams: A storage model for personal data. *ACM SIGMOD Bulletin, March*, (1996), 80-86,
- [5] Janssen, W.C. and Popat, K. UpLib: a universal personal digital library system. In *Proceedings of ACM Symposium on Document Engineering*, (Grenoble, France). ACM Press, 2003, 234-242.
- [6] Nielsen, J. and Landauer, T.K. A mathematical model of the finding of usability problems. In *Proceedings of INTERCHI'93*, (Amsterdam, The Netherlands). ACM, 1993, 206-213.
- [7] Nielsen, J., *Usability Engineering*. 1994, San Francisco, CA: Morgan Kaufmann.
- [8] Paynter, G.W., Witten, I.H., Cunningham, S.J., and Buchanan, G. Scalable browsing for large collections: a case study. In *Proceedings of Digital Libraries 2000*, (San Antonio, Texas). ACM Press, 2000, 215-223.
- [9] Rieman, J. The diary study: a workplace-oriented research tool to guide laboratory efforts. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, (Amsterdam, The Netherlands). ACM Press, 1993, 321-326.
- [10] Shneiderman, B., Tree visualization with tree-maps: A 2-D space-filling approach. *ACM Transactions on Graphics*, 11, 1, (1992), 92-99,
- [11] Soules, C.A.N. and Ganger, R. Why can't I find my files? New methods for automating attribute assignment. In *Proceedings of 9th Workshop on Hot Topics in Operating Systems*. 2003, 115-120.
- [12] Witten, I.H. and Bainbridge, D., *How to build a digital library*. 2002, San Francisco, CA: Morgan Kaufmann.
- [13] Yeates, S., Bainbridge, D., and Witten, I.H. Using compression to identify acronyms in text. In *Proceedings of Data Compression Conference*. 2000, 582.

The Influence of the Scatter of Literature on the Use of Electronic Resources Across Disciplines: A Case Study of FinELib

Pertti Vakkari and Sanna Talja

Department of Information Studies, University of Tampere, FIN-33014, Finland
{Sanna.Talja, Pertti.Vakkari}@uta.fi}

Abstract. This paper reports on how disciplinary variation in the scatter of literature affects the searching and use of electronic information services (EIS) by university faculty. The data consist of a nationwide web-survey of the end-users of FinELib, The Finnish National Electronic Library. The results show that discipline and scatter of literature are significantly associated with the number and types of electronic databases used. The scatter of literature across several fields activates researchers to more frequently search for and use various types of EIS. Especially the results concerning search methods challenge previous hypotheses and suggest important changes brought by the digital environment.

1 Introduction

Various types of user studies can support the design and evaluation of digital libraries. Studies may range from experimental evaluations testing particular system features to large-scale surveys on users and usage patterns. Most helpful are studies that not only measure use of electronic information services but also connect specific characteristics of user communities to behaviors in digital environments (Hert 2001). Such studies enable us to better understand patterns of use and help in developing services to better match users' work practices and information habits.

The supply of electronic information services (EIS) continues to grow and diversify. It seems that this brings about related changes in scholars' ways of searching and using literature. The evidence of the nature of these changes is not conclusive, however. Tenopir, King and colleagues (2003a, b,c, 2004) have conducted a series of studies in different fields. These show that the increased provision of electronic journals has increased the amount of articles read, time spent on reading, and range of journals used by scholars. Results also hint that scholars' methods of searching literature may be changing in the electronic information environment (Tenopir & al 2003a). However, studies focusing on disciplinary differences in the use and uptake of EIS show the persistence of domain-specific patterns of searching and using literature (Covi 2000; Kling & McKim 2000; Talja & Maula 2003). Thus, the evidence on the impact of EIS on scholars' search methods and use of literature is not conclusive.

Earlier results do clearly show that the use of EIS varies considerably across disciplines (Eason et al. 2000, Tenopir 2003, Talja & Maula 2003). Disciplinary differences are naturally in part due to the uneven provision of EIS between disciplines (Borgman 2000; Törmä & Vakkari 2004). Also evident is that work practices and habits of using literature vary between disciplines and influence the need for and use of various types of EIS (Borgman 2000; Talja & Maula 2003; Fry & Talja 2004). It is not totally clear, however, which characteristics of disciplines underlie these differences.

Talja and Maula (2003) found that a central disciplinary feature influencing patterns of searching and the use of EIS is the degree of the scatter of literature within a discipline. Scholars in high scatter fields used EIS more, and different types of EIS, than scholars in low scatter fields (*ibid.*). However, there exists no large-scale studies on how variation in the scatter of literature influences the use of EIS and scholars' search methods.

The aim of this study is explore how variation in the scatter of literature is associated with the number and types of EIS used by university faculty across disciplines, and with the methods they use for finding electronic journal articles relevant for their work. The data of our study consists of a web-based survey of the use of FinELib, The Finnish National Electronic Library. Understanding changes in scholars' information behavior and literature use in the electronic information environment requires relatively large projects (Banwell & al 2004). This survey provides comprehensive data on the use of EIS in disciplinary groups. Studies on the use of networked resources have often focused on single disciplines, while large-scale cross-disciplinary studies have been relatively uncommon.

2 Earlier Results

2.1 Scatter of Literature and Search Patterns

According to numerous studies, representatives of science and medicine use electronic resources more than humanists and social scientists (Borgman 2000, Talja & Maula 2003, Tenopir 2003, Törmä & Vakkari 2004). Bates (2002, 138) has suggested that one important factor influencing search patterns and the use of EIS is the degree to which information on a subject is distributed (scattered) among the resources where such information may be expected to be found.

The distinction between low and high scatter domains was originally made by Mote (1962). Scholars in low scatter fields are served by a small number of highly specialized journals, whereas in high scatter fields, relevant materials are scattered across several disciplines and published in a large number of different journals. Literature searches in low scatter fields need not extend to other subject areas or fields, whereas in high scatter fields, the researcher must typically cross several disciplines to locate all relevant materials (Mote 1962; Bates 1996). Findings show that scholars in high scatter fields put more effort on searching than colleagues in low scatter fields (Mote 1962; Packer & Soergel 1979, Bates 1996). Studies conducted in the print environment showed that scholars in high scatter areas made more use of

current awareness methods and spent more time on searching (Packer & Soergel 1979).

In all, scatter of literature seems to affect EIS, however, we lack knowledge of how variation in the scatter of literature influences patterns of searching and EIS use across disciplines.

2.2 Search Methods

Studies conducted in the print environment have shown clear differences to exist in the relative importance of different search methods - directed searching, browsing and chaining - across disciplines (Bates 2002). In the print environment, chaining from seed documents and browsing were the methods most used especially by humanists and social scientists. Scholars in these fields rarely conducted directed searches in reference databases (cf. Bates 2002; Case 2001). Some empirical studies have suggested that electronic journal services are used more in disciplines where directed searching - doing subject searches in databases whose materials have been indexed, catalogued and classified - is the predominant search method (Eason et al. 2000; Talja & Maula 2003). Those relying more on browsing and chaining techniques were found to use these services less.

The existing results of the influence of the availability of full-text electronic journals on search behaviors suggest the increased popularity of directed searching especially in large mixed-journal title databases (Tenopir & al 2003a). Browsing of core journals remains important (*ibid.*), but findings from the eJUST project (Institute for the Future 2002) suggest that users initially familiarize themselves with the supply of EIS via search-and-retrieval of content. Especially younger scholars may increasingly grow into a pattern of searching that bypasses the journal as a boundary delimiting a distinct body of ideas and content. The ease of backward and forward chaining are major changes in the electronic environment, and according to the findings of the eJUST a majority of e-journal users used links within and between journal articles.

A consistent finding in studies of scholars' information behavior is that scholars discover a high proportion of essential materials through their colleagues and scholarly networks (Case 2001, Talja 2003). Recent studies focusing on scholars' collaboration in document retrieval show that the importance of colleagues in the discovery of relevant articles has not diminished (Talja 2003). However, "word of mouth" was far more vital for discovering relevant materials in the print environment, because in the EIS environment scholars can conduct relatively comprehensive searches in minutes without leaving their desks.

In sum, there is evidence about how the new search possibilities are changing scholars' patterns of searching and using literature in the electronic environment, but few studies have taken a comparative cross-disciplinary approach.

3 Research Design

The National Electronic Library, FinELib, was established in 1997 (<http://www.lib.helsinki.fi/finelib/english/>), and is operated by the Finnish National

Library. The FinELib consortium negotiates user-rights agreements and acquires electronic resources on a centralised basis for its member organisations: universities, polytechnics and research institutes. FinELib is clearly the major supplier of electronic materials at universities in Finland. Most of the electronic literature used by Finnish university faculty is provided by FinELib that offers about 8200 full-text online journals, reference databases, dictionaries and reference works. Academic libraries purchase their own collections from this supply of resources.

3.1 Data Collection

The data used in this study was collected by FinELib as its annual user survey via a www-questionnaire. It was posted in FinELib's homepage during April and May of 2004, and advertised in university libraries' main pages, and in some institutions, through e-mail. The target population of the study consists of faculty and full time PhD students in Finnish universities. In all, 900 faculty members and PhD students filled the questionnaire. We conducted a detailed analysis of the disciplinary representativeness of the sample compared to the target population using the Kota database of university statistics. Despite the self-selection of respondents, the sample is representative by disciplinary categories. Natural sciences and economics are somewhat over-represented in the data, while engineering is somewhat underrepresented (20 % in the sample, 26 % in the target population). In addition, the sample proved to be representative in terms of faculty members' status compared to the target population.

There were respondents from all 20 Finnish universities except from the Academy of Fine Arts and Theatre Academy. Respondents from the University of Helsinki were over-represented in the data (29,1 % vs. 21,5 %). A relatively small university, Turku School of Economics was over-represented (3,8 % vs. 1,3 %) which was reflected in the over-representation of economics in the disciplines (11 % vs. 7 % in the target population). Naturally, only users of FinELib resources responded the survey. However, the size and representativeness of the data gives a reliable point of departure for exploring the use of FinELib.

3.2 Used Variables

The dependent variables are the number of various types of important databases used and the methods of searching for electronic journals or articles. The independent variables are discipline and the scatter of literature. The influence of academic status, age and gender was also tested. An elaboration showed only a weak association between them and the dependent variables. Therefore these variables were not analyzed in greater depth.

The first group of dependent variables consists of the number of electronic 1) full-text journal databases, 2) reference databases and 3) reference works used and rated as important. The number of databases used was chosen as the indicator because of its relatively high reliability and non value laden nature: the respondents ticked from the list of databases provided by FinELib those which they used. They also rated their

centrality to their work on a four-point scale: very important - important - not particularly important - do not use. The use measure was formed by counting in each category the number of databases which were rated important or very important. In 2004, FinElib offered 20 journal databases such as EbscoHost, JSTOR, ScienceDirect, and OVID; 12 reference databases such as Cambridge Scientific Abstracts (CSA) and Chemical Abstracts; and reference works such as Encyclopedia Britannica Online, and Oxford English Dictionary.

The second group of dependent variables measure the importance of methods of searching for electronic journals and articles. The respondents were asked how they had found the electronic articles or journals relevant for their work from the FinElib materials. The options given were: 1) browsing key journals of the field, 2) chaining from the reference lists of publications, 3) keyword searching in reference databases, 4) keyword searching in full-text journal databases, 5) through colleagues or other persons or 6) through alert services. They rated the importance of these methods on a three-point scale: important - rather important - do not use.

Scatter was measured by asking the respondents whether they use publications 1) mainly from their own field, 2) to a certain extent also from other fields, or 3) mainly from several fields. In beginning of the web questionnaire, field was specified as "a discipline or field of study."

In the questionnaire, respondents were asked to place themselves into a disciplinary grouping. This categorization of respondents was used in the 2004 FinElib questionnaire to ensure comparability of results and degrees of use with the results of earlier FinElib user surveys. The questionnaire's grouping of disciplines into six broad categories corresponds to the official categorization by the Ministry of Education (Table 1).

Table 1. Disciplinary categories

Name	Disciplines
Humanities	history, folklore, education, theology, psychology, linguistics, fine arts, music, theatre and dance
Natural sciences	mathematics, physics, chemistry, agriculture and forestry, dietetics, food industry and home economics
Economics	economics
Engineering	engineering, computer science and architecture
Medicine	medicine, nursing science and physical education
Social sciences	social sciences, law and administration

These disciplinary categories are evidently not internally homogeneous regarding their research culture and literature orientation. Within the humanities group, psychology and education may share more features with social sciences than humanities. This kind of within group variance may decrease the between group variance of disciplinary categorization, reducing its explanatory power.

4 Results

4.1 Discipline and the Use of Electronic Resources

In the following section we analyze variation across disciplines in how many full-text journal databases, reference databases and reference works were used and rated as important.

Table 2. The average number and types of databases used in disciplinary groups

Group	The number of databases rated as important		
	Journal databases	Reference databases	Reference works
Humanities (n=175)	2,8	1,3	2,1
Natural sciences (n=265)	4,1	1,8	2,0
Economics (n=95)	4,5	1,1	1,4
Engineering (n=178)	3,9	1,2	1,8
Medicine (n=92)	4,6	0,9	1,7
Social sciences (n=95)	4,0	0,9	2,1
Total (n=900)	3,9	1,3	1,9

There are significant differences between the disciplinary groups in the number and types of databases used (Table 2). An ANOVA shows that the number of journal databases ($F=14,5$; $p=0,000$), reference databases ($F=9,2$; $p=0,000$) and reference works ($F=3,2$; $p=0,007$) used was significantly associated with discipline. Humanities scholars used significantly less journal databases than scholars in the other disciplinary groups ($p<0,05$: Dunnett C). Both the nature of research conducted in the humanities and the weaker availability of suitable journal databases in humanities are likely to influence this difference. Humanities scholars used on the average about three journal databases while representatives of medicine and economics used about four and a half journal databases. Natural sciences, engineering and social sciences formed an intermediary group rating about four databases as important.

The profiles of the disciplinary groups varied also in the types of databases used. The importance of reference databases varied so that natural scientists rated nearly two reference databases as important while the other groups used on the average one. This difference was statistically significant ($p<0,05$: Dunnett C). In humanities and social sciences reference works are relatively important electronic sources, reflecting the information needs and literature use in these groups. Natural scientists also rated on the average two reference works as important.

In sum, in all fields journal databases were the most important type of resource. Scholars in the medicine, economics, natural sciences and social sciences groups used on the average about four journal databases. Compared to other fields, humanities scholars used less journal databases, and relatively more reference works and reference databases.

4.2 Scatter of Literature and Use of Electronic Resources

Next we analyze the relationship between scatter of literature and use of FinElib databases. There were significant differences between the disciplines in the use of literature from other fields ($p=0,000$; X^2). In medicine 52 %, in economics 44 %, in natural sciences 42 %, and in engineering 40 % of the researchers used literature mainly from their own field. In contrast, in social sciences 21% and in humanities only 13 % of scholars used literature solely from their own field. Over one third of humanists and social scientists used literature from several fields. In economics one quarter, in engineering one fifth, and in natural sciences and medicine about one seventh used literature from several fields. In medicine, natural sciences and engineering the use of literature was more focused on own field, whereas in humanities and social sciences relevant publications were more frequently scattered across several fields.

Table 3. The number of various types of important databases used by discipline and scatter.

I use publications in my work from	The number of important		
	Journal databases	Reference databases	Reference works
Own field mainly (n=312)	3,7	1,2	1,7
Other fields to some extent (n=371)	3,7	1,3	1,9
Several fields mainly (n=206)	4,6	1,7	2,3
Total (n=889)	3,9	1,4	1,9

Table 3 shows that scatter is significantly associated with the number of journal databases ($F=14,9$; $p=0,000$), reference databases ($F=8,7$; $p=0,000$), and electronic reference works used ($F=7,8$; $p=0,000$). As can be expected, scholars in low scatter fields rated fewer source types used as important compared to those who use literature from several fields ($p<0,05$: Dunnett C). The scatter of literature across several fields activates the researchers to use databases of all types more. This corroborates earlier results that keeping up-to-date is more difficult in high scatter areas (Packer & Soergel 1979) and that that scholars in high scatter fields must engage in considerably more seeking and search more databases than their colleagues in low scatter fields (Bates 1996).

An ANOVA showed, however, that the degree of scatter of literature did not produce similar patterns in the use of databases types across the disciplinary groups. In medicine, economics, natural sciences and engineering, high scatter of relevant literature led researchers to use all resource types more compared to their colleagues in low scatter fields ($p<0,05$: Dunnett C). In social sciences, there were no differences in the number of journal databases used by scholars in high and low scatter fields ($p>0,05$: Dunnett C). In contrast, in the humanities, scatter was associated with the number of journal databases used, but not with the number of reference databases used. Overall, however, the results clearly show that a higher degree of scatter requires a higher number of different resources to be used, and the relatively high degree of interdisciplinary literature use in social sciences and humanities.

4.3 Discipline, Scatter of Literature and Search Methods

Keyword searching in journal databases and keyword searching in reference databases were the two most important means of discovering relevant electronic materials (Table 4). The importance of different search methods varied to some extent between the disciplinary groups. Keyword searching in journal databases was significantly more important in natural sciences than in the other groups. Keyword searching in reference databases was the most used method in economics and engineering, and searching in journal databases the second most important route to journals. This reflects the ready availability of monodisciplinary reference databases in these groups.

Keyword searching was the most important method of discovering relevant materials in all disciplinary groups. It is surprising that there is no more variation in importance of different search methods across the groups. For instance, we would expect the importance of chaining and browsing to be higher as both can easily be done in electronic journal services.

Table 4. The average rating (1= important - 3= do not use) of search methods by discipline

Discipline (n)	The rating of search methods					
	Searching ref DB	Searching journ. DB	Colle- agues	Browsing journals	Chaining	Alert services
Hum (175)	1,8	1,7	2,4	2,0	2,3	2,8
Nat (265)	1,8	1,3	2,1	1,9	2,0	2,4
Econ (95)	1,6	1,8	2,1	1,8	2,0	2,4
Eng(178)	1,5	1,7	2,1	2,0	1,9	2,6
Med (92)	1,9	1,5	2,3	1,9	2,3	2,5
Soc (95)	1,8	1,8	2,2	1,8	2,0	2,6
Tot (900)	1,7	1,6	2,2	1,9	2,1	2,5
p.	0,002	0,000	0,000	0,11	0,000	0,000

Browsing of key journals was the third important method of finding relevant journal articles from FinElib, and it was not significantly associated with discipline. Chaining was rated as less important, as were colleagues. Surprisingly, humanities scholars rated chaining significantly less important than scholars in other fields ($p < 0,05$: Dunnett C). Numerous earlier studies conducted in the print environment show that chaining and browsing are the methods most frequently used by humanists for identifying relevant sources, and that humanists rarely conduct keyword searches (cf. Bates 2002).

Colleagues were rated as a less important means for discovering electronic journal articles. In humanities, colleagues were significantly a less important means of discovering relevant materials than in the other groups ($p < 0,05$: Dunnett C). Humanities and social sciences scholars also used alert services least.

Contrary to what we expected, the scatter of literature was not associated with the importance of various search methods. Although researchers use literature to varying degrees from several fields, scattering was not significantly connected in their methods of searching except keyword searching in reference databases (reference

searching: $F=5,3$, $p=0,005$; full text searching: $F=2,4$, $p=0,094$; colleagues: $0,3$, $p=0,75$; browsing journals: $F=0,4$, $p=0,68$; chaining: $F=0,01$, $p=0,99$; alert services: $F=0,5$, $p=0,62$). Those who used publications mainly from several fields did more keyword searching in reference databases than their colleagues ($p<0,05$: Dunnett C). As stated earlier, scholars in high scatter fields use significantly more databases of different types than their colleagues in low scatter fields.

5 Discussion and Conclusions

Our results show that in all disciplines full-text journal databases were clearly the most important electronic sources of information for work. Natural scientists were the heaviest users of all database types. In medicine and economics researchers focused mainly on journal databases, whereas the use of various source types varied less in the humanities. Compared to others they used less journal databases, but more reference works.

The findings show that scatter has a significant influence on the use of EIS, corroborating earlier findings and hypotheses (Talja & Maula 2003). A high scatter lead to a more intensive use of both journal and reference databases. Researchers using literature mainly from several fields used more databases of all types compared to colleagues who used literature mainly from their own field.

Using publications across several fields is increasingly easy in the electronic environment through aggregated mixed-journal databases containing materials from several fields. Earlier findings (eJUST, Institute for the Future 2002; Tenopir 2003a) suggest an increased use of literature from various fields as a consequence of the increased supply of mixed-journal databases. Our results also showed that over one third of humanities and social science scholars in our sample used literature mainly from several fields. It is evident that the need of tools for supporting interdisciplinary use of literature is greatest in the disciplinary groups in which the supply of EIS is not the most comprehensive.

Our results corroborate the findings of Tenopir and colleagues (2003a) according to which searching is increasingly used as a means of discovering journal articles in the electronic environment. Browsing key journals, discovering relevant materials through colleagues, and chaining did not emerge as equally important means of finding relevant electronic journal articles. It is evident that scholars did not yet use all the possibilities provided by the EIS environment. That browsing and chaining were not more important methods of discovering relevant articles is surprising and deviates from the findings of the eJust project (2002). Especially surprising was the low priority given to browsing and chaining in humanities. The sample of the study consisted of active users of FinElib resources, therefore, some caution is necessary in the interpretation of the results. It is evident that humanities and social sciences scholars using mainly Finnish literature and working in areas where the availability of suitable electronic resources is not yet very high did not respond the survey. It may be that those using traditional "artisan" approaches to searching (Bates 2002) such as chaining from seed documents were not equally represented in the sample.

Another finding that is clearly inconsistent with earlier findings (Case 2001) is that colleagues had a minor role for finding literature in FinElib, especially in the

humanities group. Talja's (2003) study of collaboration in document retrieval indicates that natural scientists - especially those working in groups - may share a considerable amount relevant documents directly within the group. This may reduce these scholars' searching within EIS. In humanities, sharing references to relevant materials is a qualitatively different activity. Humanities scholars share references to theoretical literature, especially books, similarly decreasing the need for EIS use. Although colleagues were a minor source of information about FinElib materials, we do not believe that the role of colleagues in discovering in relevant materials in general would have diminished.

The degree to which scholars used literature from other fields was only slightly connected to their methods of searching literature across the disciplinary groups. Our findings thus did not lend support to the hypothesis presented by Bates (2002) that scholars in high scatter fields use chaining and browsing as their primary search methods, whereas directed keyword searching is a more effective method for finding relevant materials in low scatter fields. The increase in scatter increased only the importance of searching in reference databases. The number relevant databases is higher in high scatter fields implying greater effort in keeping up-to-date. In some fields, especially those with a higher degree of vocabulary control, directed searching across fields is greatly facilitated by mixed-journal databases containing journals from several fields. In other disciplines, researchers in high scatter fields probably reduce their search load by first searching databases for references and then continue to the full-text journals. The provision of multisearch tools that facilitate searching across databases will not reduce the problems of scattering in fields where the consistency of vocabularies across journals and databases is not so high.

This study tested several hypotheses concerning the associations between scatter and the use of electronic resources through a large-scale representative web-based survey, showing the scatter of literature to be a significant context for information behavior. Next in our project we will analyze patterns of searching and literature use within the disciplinary categories. It is evident that fields within the broad categories differ to some extent in their work and information practices. In addition, the refinement of the survey instrument for tracking search methods will yield an even more detailed picture of the use of digital libraries. Given the high amount of cross-disciplinary literature use especially in humanities and social sciences, both qualitative and quantitative research linking patterns of searching and literature use to the nature of topics and problems studied in cross-disciplinary and transdisciplinary research will add to our understanding of the contexts of digital library use.

References

- Banwell, L., Ray, K., Coulson, G., Urquhart, C., Lonsdale, R., Armstrong, C., Thomas, R., Spink, S., Yeoman, A., Fenton, R., Rowley, J., The JISC User Behaviour Monitoring and Evaluation Framework. *J. Doc.* **60** (2004) 302-320.
- Bates, M., Learning About the Information Seeking of Interdisciplinary Scholars and Students. *Library Trends* **45** (1996) 155-164.

- Bates, M. J., Speculations On Browsing, Directed Searching, and Linking in Relation to the Bradford Distribution. In: Bruce, H., Fidel, R., Ingwersen, P., Vakkari, P. (eds.): *Emerging Frameworks and Methods: Proceedings of the Fourth International Conference on Conceptions of Library and Information Science (CoLIS4)*. Libraries Unlimited, Greenwood Village (2002) 137-149.
- Borgman, C. L., *From Gutenberg to the Global Information Infrastructure: Access to Information in the Networked World*. The MIT Press, Cambridge, Mass. (2000).
- Case, D. *Looking for Information*. Academic Press.(2001)
- Covi, L. M., Debunking the Myth of the Nintendo Generation: How Doctoral Students Introduce New Electronic Communication Practices into University Research, *J. Am. Soc. Inf. Sci.* **51** (2000) 1284-1294.
- Eason, K. Richardson, S., Yu, L., Patterns of Use of Electronic Journals. *J. Doc* **56** (2000) 477-504.
- Fry, J., Talja, S., The Cultural Shaping of Scholarly Communication: Explaining E-journal Use Within and Across Academic Fields. In: *ASIST 2004: Proceedings of the 67th ASIST Annual Meeting*, Vol. 41. Information Today, Medford NJ. (2004) 20-30.
- Hert, C., User-centered Evaluation and Its Connection to Design. In: McClure, C., Bertot, J. (eds.) *Evaluating Networked Information Services*. Medford, N.J., Information Today (2001) 155-173.
- Institute for the Future, *E-Journal User Study eJUST: Research Findings* (2002). Retrieved 27 February from <http://ejust.stanford.edu/SR-786.ejustfinal.html>.
- Kling, R., McKim, G., Not Just a Matter of Time: Field Differences and the Shaping of Electronic Media in Supporting Scientific Communication. *J. Am. Soc. Inf. Sci* **51** (2000) 1306-1320.
- Kota Database of University Statistics. www.csc.fi/kota/aihelista1.html.
- Mote, L.J.B., Reasons for the Variation of Information Needs of Scientists. *J. Doc.* **18** (1962) 169-175.
- Packer, K. H., Soergel, D. The Importance of SDI for Current Awareness in Fields with Severe Scatter of Information, *J. Am. Soc. Inf. Sci.* **30** (1979) 125-135.
- Talja, S., Information Sharing in Academic Communities: Types and Levels of Collaboration in Information Seeking and Use. *New Review of Information Behavior Research* **3** (2003) 143-159.
- Talja, S., Maula, H., Reasons for the Use and Non-use of Electronic Journals and Databases: A Domain Analytic Study in Four Scholarly Disciplines. *J. Doc* **59** (2003) 673-691.
- Tenopir, C., Use and Users of Electronic Library Resources: An Overview and Analysis of Recent Research Studies. Council on Library and Information Resources (2003a). URL:<http://web.utk.edu/~tenopir/eprints/index.html>. (Referenced 2.2.2005).
- Tenopir, C., King, D.W., Boyce, P., Grayson, M., Zhang, Y., & Ebuen, M., Patterns of Journal Use by Scientists Through Three Evolutionary Phases. *D-Lib Magazine* (2003b) **9**. Retrieved 7 January, 2004 from <http://www.dlib.org/dlib/may03/king/05king.html>.
- Tenopir, C., King, D.W., Boyce, P., Grayson, M., Paulson, K-L., Relying on Electronic Journals: Reading Patterns of Astronomers. Preprint (2003c) Available at http://web.utk.edu/%7Etenopir/eprints/tenopir_jasist_article_042503_preprint.pdf
- Tenopir, C., King, D.W., & Bush, A. Medical Faculty's Use of Print and Electronic Journals: Changes Over Time and Comparison With Other Scientists. *J. Am. Med. Assoc. (JMLA)* **92** (2004) 233-241.
- Törmä, S., Vakkari, P. Discipline, Availability of Electronic Resources and the Use of Finnish National Electronic Library - FinELib. *Information Research* **10** (2004) paper 204 [Available at <http://informationr.net/ir/10-1/paper204.html>].

Information Seeking by Humanities Scholars

George Buchanan, Sally Jo Cunningham, Ann Blandford, Jon Rimmer, and
Claire Warwick

University College London, United Kingdom
{g.buchanan, a.blandford, c.warwick}@ucl.ac.uk
University of Waikato, New Zealand
sallyjo@cs.waikato.ac.nz

Abstract. This paper investigates the information seeking of humanities academics and scholars using digital libraries. It furthers existing work by updating our knowledge of the information seeking techniques used by humanities scholars, where the current work predates the wide availability of the Internet. We also report some of the patterns observed in query and term usage by humanities scholars, and relate this to the patterns they report in their own information seeking and the problems that they encounter. This insight is used to reveal the current gap between the skills of information seekers and the technologies that they use. Searches for 'discipline terms' prove to be particularly problematic.

Keywords: Digital Libraries, Human-Computer Interaction, Information Seeking, Humanities.

1 Introduction

Information retrieval research has made significant improvements in the quality of results from interactive searches. Simple approaches such as the log rule [18] and query term expansion [9] give users much better results than earlier techniques. However, in the complex arena of digital libraries, where users have to choose from a rich variety of search criteria - e.g. metadata field, Boolean logic - problems arise in the users' choice of effective criteria. In particular, users seem to have problems interpreting causal relationships between their inputs and the returned results.

This gap of interpretation requires further study. Our research is founded on the approach of information seeking research. In contrast to information retrieval, which focuses on improving search performance by technical means, information seeking describes human behaviour. Improvement is achieved through a better understanding of user abilities and expectations, leading to changes in underlying system mechanics or the human-computer interaction as appropriate. The objective of our research is to reveal more about users' intentions in their information seeking behaviour, and the strategies and tactics they choose that they believe will improve their outcomes. In understanding their anticipations, we hope to provide interactions that better bridge the gap between the system's underlying operation and the users' conception of the system.

This study focuses on humanities scholars – intellectually able seekers who are not technical in orientation. The body of the paper commences with an examination of

information seeking research - both general material and findings particular to the humanities. Having reviewed the current state of knowledge, we introduce our user study in Section 3. The findings of that study are presented in Section 4 and discussed in Section 5, before we conclude with a summary of our contribution and future plans.

2 Information Seeking

A key theme in user-centred information retrieval research has been the investigation of the choices users make when searching: what terms they choose, how many, and which other features (e.g. phrase search or Boolean logic) they naturally use.

One immediate distinction that can be made between users is their level of experience, in terms of either interactive search or the subject domain. Lucas and Topi found that experienced, trained searchers use more query terms and exploit Boolean logic more frequently [10]. Hsieh-Yee [7] and Wildemuth [17] studied the impact of domain knowledge; they found that underlying search skills have a more significant effect on query formation than domain knowledge.

The overall picture from studies of choice patterns within user queries, particularly on the web (e.g. [8]), has been that users use few terms and seldom employ Boolean logic or other advanced search criteria. The consistent picture of interactive search is that few expert searchers exist, and most seekers use simple two or three term queries.

In addition to studies of users' behaviour when engaged in interactive search, other researchers have investigated the broader patterns of users' information seeking strategies. One key model is that of Ellis [5,6], who identifies actions such as *Starting*, where the seeker identifies initial sources of information, and *Chaining*, where references are followed forwards or backwards to extend the scope of the covered area. In Ellis' model, users may move from one action to another and no order is assumed. Below, we relate our findings to the work of Ellis.

2.1 Information Seeking in the Humanities

The available literature on information seeking in the Humanities is relatively limited. Wiberley et al's [15] early work disputed the then well-accepted view that humanities search terms were frequently general and imprecise. He identified the frequent use of items such as the names of persons and places. This work was later extended by Bates [2]. This body of evidence has revealed that humanities scholars in fact frequently use specific and highly selective query terms.

Recently, comparisons have emerged with other academic disciplines. Whitmire†[14] reports that humanities seekers demonstrate a significantly higher use of library facilities than other academics. They more often use catalogues, turn to librarians for assistance, browse, use reserve collections and journal indexes, etc. The collaborative aspect of information seeking has frequently proven significant; Watson-Boone [13] identifies the importance of the professional network of neighbouring and distant colleagues in the information seeking of humanities scholars. Thus, humanities information seeking demonstrates a strong use of human support, as well as a more intensive use of printed or mechanised seeking tools. However, though

humanities scholars do turn more frequently to librarians, they do so with some reluctance [16].

The impact of the digital information seeking environment in the workplace is particularly poorly understood. Tibbo, [12] in her investigation of the information seeking of historians, states that “we have no idea if they succeeded in finding these materials based on web searches”, and similarly questions electronic catalogues. Wiberley’s later report [16] itself faces the problem of being a retrospective over the previous ten years. The availability of electronic resources on the researchers’ work table was only becoming a factor towards the end of that period.

Though this material is helpful in informing the design of library services for humanities scholars, the picture is far from complete. For example, little has emerged about the use of new electronic sources of information such as the Web, and other recent developments such as the widespread introduction of online journals also have yet to be systematically studied. For the technical development of digital library (DL) systems, the distinctions made in [15] and [2] between different query terms do not fit well with search indexes for text, where distinguishing the role of a particular word is extremely difficult. In other words, there is a gap between our information seeking insights and our information retrieval technologies.

2.2 Information Seeking: Summary

This paper reports our findings which are helping to narrow this gap, and bringing information retrieval technology and interaction closer to the information seeking of humanities scholars. In the next section, we describe a user study that we undertook to update our picture of the information seeking strategies of humanities scholars, and to reveal the parts where the misfit between technology and user skill is greatest. In [3], we used information retrieval evaluation techniques to identify patterns in the textual properties of documents gathered by users in the course of their information seeking. In this paper, we again use information retrieval measures to scrutinise the information seeking of users - in this case, analysing the types of terms that they use to search for documents. Through this, we identify the relationship between the types of search terms identified by Bates [2,11] and the frequency of occurrence of words associated with each type of term.

3 User Study

To improve our understanding of the information seeking strategies of humanities academics we conducted interviews with eighteen members of the Faculty of Arts and Social Sciences at the University of Waikato in New Zealand.

The interviews were conducted over a six week period, in the participants’ own offices. We wished to elicit their experiences of using digital libraries and electronic library catalogues, and their perceptions of the problems and successes they experience in using them. Participants were encouraged to demonstrate their strategies using the University’s electronic catalogue, which provides direct access to a number of electronic resources as well as the books physically in the library. The use of a familiar

system was intended to reduce the effects of learning, and to focus their explanation of their information seeking in a context where they were able to demonstrate their own expertise. In addition to the traditional interview approach, the search terms used were recorded and later analysed for their effect.

Our goals were to identify areas where strategies are already well formed, areas where seeking was commonly perceived as difficult, and the types of query choices made (terms, options, etc.). In the latter case, we wished to identify any correlation between problematic areas of information seeking and the sorts of query choices made.

3.1 Participants

In this paper a two-letter subject code and number identify each participant. Participants in our study were academic faculty, ranging in age from 28 to 65; their backgrounds are outlined in Table 1. We compared experiences across different levels of experience, background, seniority and age. The participants were recruited through direct contact and a circular through the faculty newsletter.

Due to the relatively difficult access to large reference libraries in New Zealand, we might expect a higher adoption rate of technology compared to academics in Europe or North America.

3.2 Study Method

The interviews occurred in the participants' own offices and they used their own computer equipment when interacting with the library systems. After a short briefing on the objectives of our research, participants engaged in semi-structured interviews concluding with a demonstration by them of their use of the information seeking tools available to them. During the interview, only the investigator and the participant were present, the investigator noting the participant's responses and explanations. Their interactions with the library system and/or Internet search engines were also noted, and search terms recorded for later analysis. In addition to library systems, the academics were also encouraged to describe their use of the web in general as a research and information resource.

After the interview, the university catalogue index was used to identify the frequency of the terms that the participants had used. Where another source was used, the result list for each individual term used was inspected to deduce the effect of each word and the default search criteria recorded. The university catalogue system produces results in a number of forms, including ranked list (default) and alphabetic title order. Search criteria include "Keyword anywhere" and "Title begins with". The participant's explanation of the search - i.e. their intention - was then compared against the achieved effect. Terms were also compared against the search term taxonomy of Bates [2]. Thus, for each term we identified its semantic form (e.g. geographical name) and its rate of occurrence (document frequency).

Query (or browsing) intentions were related to Ellis' information seeking strategies [5,6]. This applied to information seeking activity described by the interviewee during both the interview and the demonstration parts of the study.

Table 1. Brief details of participants

Identity	Subject	Post	Years in Field	Gender	Use
AN1	Anthropological History	Lecturer	15	Female	Low
AN2	Anthropological History	Assoc. Professor	21	Female	Low
AN3	Anthropological History	Professor	20	Male	Low
ED1	Education	Dean	26	Male	High
ED2	Education	Research fellow	20	Female	Med
EN1	English	Senior Lecturer	12	Male	High
EN2	English	Assoc. Professor	20	Male	Low
EN3	English	Senior Lecturer	20	Female	Med
HS1	History	Lecturer	10	Female	Med
HS2	History	Senior Lecturer	35	Male	Med
HS3	History	Senior Lecturer	27	Female	Low
LI1	Linguistics	Assoc. Professor	25	Male	Med
LI2	Applied Linguistics	Senior Lecturer	10	Female	High
MS1	Media Studies	Professor	24	Male	High
PL1	Philosophy	Lecturer	8	Female	High
PL2	Philosophy	Professor	30	Male	Med
PO1	Politics	Professor	25	Male	Low
EV1	Environmental Mgmt	PhD Student	3	Female	Med

4 Findings

Our participants reported a wide variety of experiences with digital libraries, electronic library catalogues and the web generally. In this section, we discuss the role of each of these types of resources in turn. Subsequently, we identify common strategies described by our readers, and then finally discuss the insights gained through a detailed examination of their searches.

4.1 Digital Libraries

Our readers were not explicitly aware of the concept of digital libraries. Commonly, they referred to DLs as “online databases”, or referred to the electronic library catalogues (e.g. the university’s own catalogue) that themselves linked to the content of actual DLs. The use of electronic journals also confused this issue. However, all the participants used digital libraries in one form or other. For these users, DLs as a concept were subsumed by digital forms of familiar paper-based services.

Actual use of digital libraries varied. Participant AN1 described herself as “a technophobe”, seldom using digital resources although she described how she “repeatedly tried to use online journals”. In contrast, participant ED1 described himself as an “enthusiastic user of online material”. Each of these participants saw themselves as being at an extreme of the digital use continuum.

These differences were not, however, simply personal. Both education users (ED1, ED2) reported high levels of computer use, and stated that colleagues in other institutions were similar to them. ED1 and ED2 both argued that the strong uptake of

computer technology in schools had driven them to be early adopters of online information systems: ED1 describing his use of a dial-up connection to a document database in San Diego in the mid 1980's, and ED2 her use of online material from the UK towards the end of the same decade. Conversely, the anthropological history users (AN1/2/3) identified themselves as low-frequency users and also reported that their counterparts elsewhere shared their own preferences.

One marked difference in technology arose around the issue of access. Once material was found online, educationalists reported that they seldom had difficulty retrieving it. On the other hand, Anthropological History and English users regularly found themselves "barred" by access controls - i.e. they were expected to pay for access. A few individual users - e.g. LI1 - were directly involved in projects that gave them privileged access to material of high value to them, whilst others such as EN1 personally paid for valued resources. There was a strong correlation between access difficulties and take-up: all readers who reported access problems as being acute were infrequent users of online libraries and journals.

4.2 Internet Use

Across all participants and degrees of digital library use, the Web was seen as a useful substitute for a traditional encyclopaedia. However, identifying the source was seen as an important step in verifying the trustworthiness of the information. For example, AN1 said "I use the Internet to look up definitions of words, rather than turn to a dictionary like the OED. I know that we've got access to the OED online, but I prefer to search for it instead." Similar views were reported by the other readers.

The Internet was also a good starting point for more specific strategies. For example, it could provide initial sources for chaining, quotations, and checking bibliographic data.

4.3 Career and Community Effects

Most participants reported that when they were new to an area of research or teaching, their DL use soared, especially when they were at an early stage of their career. This pattern was repeated on the smaller scale; e.g. when a new project was starting, or a new sub-field of study was explored, participants reported a peak of digital library use. For instance, EV1 reported that the start of her PhD studies resulted in an eighteen month period of extensive literature seeking, followed by another eighteen months of fieldwork, in which all she did was track for later publications. LI1 also reported the same pattern in regard to a book he was writing and, in common with EV1, was now returning to online sources to check for literature that had appeared in the meanwhile to ensure that all the references were up to date.

Findings vary with the age group and experience of the researchers interviewed. More experienced researchers relied more on personal contact and domain knowledge; they would know of many developments before they are formally reported [1]. For these users, personal contact was a more important source of new data than direct information seeking in any medium. Literature was invoked more to back up arguments than to develop them. One particular Internet strategy did emerge in this area: fifteen participants reported 'Googling' for the homepages of known researchers to either update their list of publications or check specific citation details.

4.4 Information Seeking Strategies and Tactics

In this subsection, we report on the higher-level information seeking strategies and tactics used by the participants in our study. As shall be seen, there was generally a strong relationship between the behaviours we observed and those found in information seeking models such as those of Ellis mentioned earlier.

Choosing to Search or Browse: “Browsing” in digital environments was noted as a particular problem - highly effective in physical libraries, but difficult in an electronic environment. EN1 suggested a reason for this: “with an online search you have to be more targeted as the structure is linear. On the other hand, in a real library I go to a physical place and find stuff alongside what I am looking for under the same call number; I think I can do that in the DL but I am less likely to do that.” During search, subject classifications were even more rarely used; indeed, any mention was negatively critical. For example, PL2 commented: “I’ve encountered these things in the past... I don’t think of subjects that way. I’m always surprised where books are classified.” Our readers demonstrated varying awareness of the existence of browsing in a DL, and none reported doing it regularly. This correlates with findings from our earlier studies. Search also proved an ineffective replacement for browsing, participants observing that it is easy to have lots of hits or no hits. Thus, there was little satisfaction when searching broadly in a field understood by the humanities academic as well-defined. One alternative is the use of chaining instead of browsing in the digital domain.

Choosing where to Seek: Participants reported difficulties early in their use of digital resources in identifying appropriate sources for their areas of interest. Later, having selected sites or journals that met their needs, they would return to these regularly. However, they often did so directly, not interacting with the institutional catalogue that indexed the external material. ED2 explained: “When I started, I took a long time to find what was there. Now, I just go direct - you know, not using the uni catalogue - straight to the journal or database itself.” A different approach was taken by PL1, who said “When I am searching, I just use Google; it usually finds the article I want and I go straight from there to the database - when I locate a paper often I have online access through the library... it will come up and when I click on it and access if good it will just say ‘You are logged on at the University of Waikato’ or something like that.”

Ellis’ Patterns: Ellis’ principle of chaining - using references and citations in known works to find unknown ones - was noted by all our participants. This was a dominant form of behaviour, which was often reported as the common context within which effective search occurred. Thirteen participants also reported their tracking of particular journals - described by Ellis as ‘monitoring’, and eleven participants reported the same behaviour in regard to known researchers in their field. Twelve reported behaviours Ellis described as ‘browsing’¹. ‘Verifying’ previously found information was reported by ten participants.

When returning to journals, a serial reading of the latest issues was common. For example EV1 reported: “There are certain journals that do publish a lot of useful things

¹ N.B. Ellis defines this as scanning known sources of information - this is not quite the common usage in information seeking.

and I sometimes trawl through those issue by issue - I thought it would be a waste of time, but it wasn't: I found things I wouldn't have by search."

4.5 Query Term Effects

In our study, we observed forty six different searches with 121 query terms. Previous research [2,15] has identified key query term types that appear in the searches of humanities users: names of individuals, geographical names, chronological terms and discipline terms (understood terminology within the field – e.g. in computer science, 'recall and precision'). It is argued that these query term types have a specific meaning within a discipline and therefore should be "good" query terms. However, these term types have not been studied from the perspective of information retrieval, to understand their usefulness when input to a search engine. A specific term may carry strong semantic cues, but information retrieval techniques focus instead on the rate of occurrence of a word or phrase. For example, 'precision' can occur in many more contexts than 'recall and precision' and mean different things even in a computer science corpus. Thus, there is a gap between the human understanding of the term and the treatment of the same term by the computer. In addition, we wished to see how these types of terms are used by humanities academics when performing actual queries. The effect of single words is strongly influenced by the criteria with which they are used. For example, a word occurring in a phrase plays a different role to a word used on its own.

Bates [2] and Wiberley [16] identify a range of term types that are particular forms of 'proper' words; e.g. names of places and people. Proper names were reported by all our participants as being a commonplace form of term in their searching – e.g. when searching for a journal article by a particular author. In addition, fifteen of our participants used one or more proper term in their description and demonstration of their own searching. Thus, this particular type of query term is of particular interest.

When a person's name was used, the number of document matches in the university library was typically under 100, and these were grouped by name, so the number of displayed results was usually under 25 (one page of results). Only in two cases was this number exceeded: when the name was matched in any field. In one of these cases, the results for 'Caxton' soared to just over 3000 matches from a body of over 1,000,000 works.

Geographical names were not as selective, particularly when they referred to New Zealand locations. The University of Waikato specialises in New Zealand specific research, so even regional name searches on title return hundreds or thousands of hits; for instance, a search for 'Otago' - a region of NZ - in the title of a book would return 1268 entries; when executed on 'keyword anywhere', 3174 items were returned.

What we observed in the case of common proper names was that 'keyword anywhere' represented a poor choice of search criteria. Unfortunately, this was the default used by the library catalogue, and was seldom changed by moderate and low users of DL resources. However, both forms of proper (person and place) name, particularly when used with additional terms, resulted in smaller result sets that our participants reported positively. Combined with a default relevance ranking, a satisfactory match was consistently found near the top of the search result list. Furthermore, in the case of the six participants who used phrase searching when using

people's names (all but one reporting themselves as a 'high frequency' user), the selectivity of the term prove even stronger. Each of these users also reported the need to try alternative forms of the same name.

Another form reported by Bates was the chronological period. This occurred on only six occasions within our participants' searching, but was reported in the course of eight interviews. Where chronological periods were used, a search typically returned around 1000 hits on title, with more modern periods being increasingly common (e.g. 1869 hits for "twentieth century").

One further form of term that Bates [2] specified was the 'discipline term'. Again, these terms were both reported and demonstrated by our seekers. Such terms are often small phrases that consist of two or three individual words.

Examining these terms through the online catalogue's index revealed a clear problem. When using the default "keyword anywhere" search of the university library, very high numbers of results were returned. The library indexer automatically restricts searches to returning at most 10,000 matches. This limit was consistently exceeded when such discipline terms were used; the best case was "mercantile economy", and even that resulted in over 4,000 hits. The participants in these cases found little of relevance even at the top of the search result list. This problem seldom occurred for proper name searches unless, again, several terms were used together. Conversely, use of these terms as a phrase, particularly against individual fields, often resulted in few or no hits (less than 25 hits in 18 of 21 examples, 2 being no hits). Such search criteria prove imprecise unless treated as a phrase. Participants, such as ED2, who were satisfied with digital libraries specifically checked that they used these terms in a phrase. Naive users, on the other hand, performed no checks for this and consistently obtained very large result sets.

This problem with discipline terms seems to correlate with problems reported by seekers when searching more broadly, and also ties with the strategies that they use.

When terms are combined, the effects vary widely depending on the search criteria used. For example, a combination of 'Otago' and 'twentieth century' can range from one hit to 10,000 hits, depending on the options used. The default search option for the library catalogue (keyword anywhere) in fact returns the highest figure.

Participants who reported satisfaction with their experience of online systems (e.g. LI1, ED1, ED2, EV1, PL1) all regularly described not only the search terms that they used, but which fields those terms would be used in. For these users, term and field are closely related. On the other hand, AN1 represents those with a less positive experience: "When I look stuff up, it just is far too much. My colleagues are overwhelmed, two of them anyway, by finding lots of stuff and I don't think they find much of it helpful; I don't. It is much easier just to go to the shelves and browse there". In the case of these users, they reported finding too much, and in our interview only demonstrated "keyword anywhere" searching. Given the example above, it can be seen that effective use of fields in search is critical to consistent satisfaction.

Similarly, satisfied readers articulated both a pattern of changing their search terms flexibly and, even in the case of ED1, an awareness of different strategies in different libraries: "...because I have got a, generally got a clear sense of what I am looking for. My knowledge and fluency in what I am looking for lets me get good descriptors. There's always a learning curve with a new repository. The familiarising process takes time - what works in each library." Again, dissatisfied users did not express this

understanding. However, most participants expressed little or no awareness of different strategies being needed in different contexts.

When comparing between our seekers, we sought any relationship between their attitudes and experience with digital library resources and their use of queries. As has been touched on a couple of times in this section, query criteria were used more selectively by more frequent and enthusiastic readers. Naive or unenthusiastic users only used the basic search criteria, and appear to be more frequent users of 'concept terms'. As noted above, concept terms require the use of phrase searching to improve precision in results sets; low frequency users AN1 and SS1 used one or more concept terms in each search and never used phrase search. In contrast, highly satisfied intensive users (e.g. PL1, EN1) used concept terms in only one case, and immediately applied phrase search, citing the problems that they found if they forgot to use this. Experienced and keen users of digital resources also exploited fielded querying, and proper names more frequently occurred in their search criteria. All five users who identified themselves as high intensity users described using Boolean search; of the moderate and low frequency users, only two noted an awareness of this option, and both had previously been high intensity users. However, our samples are too small to determine any statistical significance.

Strangely, where naive strategies were applied, readers would often report positive experiences with searching on the Internet as opposed to disappointment with specialised catalogues and libraries. Google was consistently identified as the Internet search engine used by our readers. Given the design choices of Google that maximise precision at the expense of recall (e.g. all search terms are required in matched documents) and positively weight query term proximity in matching documents (providing a simile to phrase searching), there may be a relationship between the design choices of the search engines, strengthening poorly focussed searches, and the users' positive remarks.

5 Discussion and Conclusions

We studied the information seeking skills and strategies of eighteen humanities academics. Across our participants, there was a correspondence between high usage (present or historic), strong search skills (e.g. use of fielded search) and a greater degree of satisfaction with DL systems. Unfortunately, it is difficult to ascertain what empowers a user to move from being a low-skill, low or moderate frequency user to a more flexible, satisfied user. Our high-frequency users could pinpoint a specific event, e.g. the beginning of a project, as a turning point in their use of DLs and this, in turn, suggests that the change can occur over short periods of time. As with our observation of career effects, Wiberley [16] draws similar conclusions. This means that observing this change in a given user study will probably prove extremely difficult.

Through comparing participants' behaviour to Ellis' information seeking model, we discovered that a few basic strategies had a central role in their use of DLs. Citation chaining was one key strategy that formed the common approach to finding contemporary academic research literature and, often, initial leads into archives.

Humanities academics, particularly those established in their field, used the academic community as an important source of recommendations. They also tracked

the work of individual known researchers, e.g. by monitoring the researchers' personal home pages – a form of chaining. Where skills of information seeking on computers were limited, it was often because the academic network was strongly developed and minimised the need for active, independent information seeking.

Our participants believed that they were successfully locating information that answered their needs. However, it appeared that on occasion the effort required to achieve success was high. For example, user AN3 succeeded in exploring a new area of interest through an extensive use of chaining, built on an initial set of documents obtained through personal contact. However, even he described this strategy as “time consuming and exhausting”.

When information seeking moved from a strongly-defined goal – e.g. ‘author and title’ search – into more uncertain areas such as conceptual searches related to a discipline term, problems rapidly emerged. Precise searches required the careful selection of search criteria that we only observed in a few users. Classification structures of the library could help locate information about a particular topic. However, our participants seldom used classifications to browse in DLs, even though they often reported browsing in physical libraries. Furthermore, only two mentioned using classifications when searching.

Future work is needed to identify techniques to better support less experienced users by assisting them in selecting appropriate search criteria (e.g. use of fields and phrases). Similarly, setting default search criteria to emphasise precision over recall should improve the results when naïve search criteria are used with discipline term queries (Google takes this approach). Support for chaining in many DL systems (including the ones studied) is poor. Given the significance of this approach for humanities academics, better citation chaining tools should considerably improve their experience of DL systems.

Acknowledgements

This work was supported by EPSRC Grant (GR/S84798). We would also like to thank the study participants and the University of Waikato for its support of the study.

References

1. A. Adams and A. Blandford. Digital libraries in a clinical setting: Friend or foe? *Procs. European Conference on Digital Libraries, LNCS 2163*, pages 213–224, 2001.
2. M. J. Bates. The Getty end-user online searching project in the humanities: Report no. 6: Overview and conclusions. *College & Research Libraries*, 57:514–523, November 1996.
3. G. Buchanan, A. Blandford, M. Jones, and H. W. Thimbleby. Spatial hypertext as a reader tool in digital libraries. In *Visual Interfaces to Digital Libraries*, pages 13–24. Springer-Verlag, 2002.
4. G. Buchanan, M. Jones, and G. Marsden. Exploring small screen digital library access with the greenstone digital library. In *ECDL '02: Proceedings of the 6th European Conference on Digital Libraries*, pages 583–596. Springer-Verlag, 2002.
5. D. Ellis. A behavioural model for information retrieval system design. *Journal of Information Science*, 15(4/5):237–247, 1989.

6. D. Ellis and M. Haugan. Modelling the information seeking patterns of engineers and research scientists in an industrial environment. *J. of Documentation*, 53(4):384–403, 1997.
7. I. Hsieh-Yee. Effects of search experience and subject knowledge on the search tactics of novice and experienced searchers. *JASIS*, 44(3):161–174, 1993.
8. B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage.*, 36(2):207–227, 2000.
9. J. Koenemann and N. J. Belkin. A case for interaction: a study of interactive information retrieval behavior and effectiveness. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 205–212. ACM Press, 1996.
10. W. T. Lucas and H. Topi. Form and function: The impact of query term and operator usage on web search results. *JASIST*, 53(2):95–108, 2002.
11. S. Siegfried, M. J. Bates, D. Wilde. A profile of end-user searching behavior by humanities scholars: The Getty Online Searching Project Report #2. *JASIS*, 44(5):273–291, 1993.
12. H. R. Tibbo. Primarily history: historians and the search for primary source materials. In *JCDL '02: Procs. Joint Conference on Digital libraries*, pages 1–10. ACM Press, 2002.
13. R. Watson-Boone. The information needs and habits of humanities scholars. *RQ*, 34(2):203–216, 1994.
14. E. Whitmire. Disciplinary differences and undergraduates' information-seeking behavior. *J. Am. Soc. Inf. Sci. Technol.*, 53(8):631–638, 2002.
15. S. E. Wiberley and W. G. Jones. Patterns of information seeking in the humanities. *College and Research Libraries*, 50:638–645, 1989.
16. S. E. Wiberley and W. G. Jones. Time and technology: A decade-long look at humanists' use of electronic information technology. *College & Research Libraries*, 61:421–431, 2000.
17. B. M. Wildemuth. The effects of domain knowledge on search tactic formulation. *Journal of the American Society for Information Science and Technology*, 55(3):246–258, 2004.
18. I. H. Witten, A. Moffat, and T. C. Bell. *Managing gigabytes: compressing and indexing documents and images*. (second edition). Morgan Kaufmann, San Francisco, CA., 1999.

ReadUp: A Widget for Reading

William C. Janssen

Palo Alto Research Center,
3333 Coyote Hill Road,
Palo Alto, California, 94304 USA
janssen@parc.com

Abstract. User interfaces for digital library systems must support a wide range of user activities. They include search, browsing, and curation, but perhaps the most important is actual reading of the items in the library. Support for reading, however, is usually relegated to applications which are only loosely integrated with the digital library system. One reason for this is the absence of toolkit widget support for the activity of reading. Most user interface toolkits instead provide support for either text editing or text presentation, making it difficult to write applications which support reading well. In this paper we describe the origins, design, and implementation of a new Java Swing toolkit widget called *ReadUp*, which provides support for reading page images in a digital library application, and discuss briefly how it is being used.

1 Introduction

The UpLib personal digital library system [11] is designed for storage, organization, retrieval, and use of an individual's digital documents. As part of this system, we originally included a simple Web-based document reader which presented each page of the document being read as an image on a Web page, along with a sidebar that provided access to operations on the document. Page images were clipped to increase effective resolution, using the techniques described in [10]. The image had a superimposed HTML imagemap, so that clicking with the mouse on the right side of the page would turn to the next page by following a link to a Web page that had the image of the next page on it. Similarly, clicking on the left side of the page would take the reader to the previous page. On the side of the page image, small icons provided the page index number of the page, explicit following-page/preceding-page buttons, and an UpLib logo button which when clicked would take the user to the UpLib repository overview.

This document reader implementation had many deficiencies. On a fast machine, the page turn could happen quickly enough that users hadn't realized it *had* happened, and would frequently double or triple click, turning multiple pages inadvertently. The trail of pages cluttered the Web browser's history stack, making effective use of the "back" button difficult. There was no provision for annotation or markup of any kind, even simple bookmarks. There was no ability to search the text of the page or the document. Web browsers often ignored the caching instructions provided in the HTTP headers, and cached potentially sensitive page images to disk. The sidebar was generated statically, used frames, and

contained seldom-used page thumbnails, display of which was not synchronized with the particular large page image currently visible. Other clients for UpLib were being developed, with similar reading interfaces, and they shared some of these problems.

To address these deficiencies, we have implemented a reading widget as a Java Swing `JComponent`, which we use in both an applet form, and directly in other applications. We call it the *ReadUp* widget. In the rest of this paper, we describe the widget's design and interface, then discuss some implementation issues, and finally consider the ways in which it is being used.

2 Related Work

A number of studies of subjects reading both paper and virtual paper documents indicate several important requirements for reading interfaces. In [14], looking at readers of both paper and virtual documents, the authors remark that “critical differences have to do with the major advantages that paper offers in supporting annotation while reading, quick navigation, and flexibility of spatial layout.” In [15], they examine annotation more fully, citing the differences between making marks directly on the source and making notes about the source, finding both to have value. In [13], Marshall examines direct annotation of books in great detail, stressing the importance of support for informal, fluid, *in situ* markings, integrated with the reading activity.

Navigation in digital documents is also an issue. The authors of [1] point out that linear reading is “an unrealistic characterisation of how people read in the course of their daily work”, a finding reinforcing the ideas of reading in [2]. Another survey [7] suggests difficulty with page manipulation and navigation when reading from screen devices, compared to paper.

A common approach to this need for better support for annotation and navigation in digital documents is to write a separate reading application which supports page manipulation, search, and annotation. Some of these applications, such as Lectrice [5] and XLibris [17], have been applications directly supporting annotation and navigation, but somewhat embedded in tablet PC hardware. Others, such as Lectk [5], Multivalent [16], 3Book [4], and “Open the Book” [6], also address the support issues, and run as applications on more conventional computing platforms. Building such an application gives full control over page presentation, and allows for additional experimentation. However, this approach does not support a Web-based interface, and has the further disadvantage of having to be installed on potential clients before any documents can be read with it. In addition, other client applications cannot take advantage of this functionality.

3 The ReadUp Widget

We eventually realized that if the custom application was designed as a new toolkit widget, many of these problems could be addressed. It could be bundled up in either a Web browser plug-in or in an applet, and used to support the

Web interface to UpLib. It could be extended to support experimental reading operations. Other applications could use the widget to support reading from an UpLib repository.

It is possible to write a browser plug-in *sui generis*, but it would have to be downloaded and installed before it could be used. Two existing plug-in technologies, Flash and Java, provide useful interface toolkits and are widely pre-installed. The use of Java seemed particularly appealing, despite the somewhat clumsy nature of the language, as much of the work in our group is being done in Java. The default user interface toolkit in Java, called Swing [21], is also a solid and portable, if unimaginative, base on which to work. Additionally, the Java applet is a well-known way to present client interfaces to users in Web pages.

A widget as complex as ReadUp contains a complete embedded user interface, which must be neutral enough to mesh well with the various applications it is used in. This interface, which perhaps should be thought of as the *reader* interface, is designed to be flexible enough to experiment with new modes of interaction for reading, but also in its basic form provide support for two major reading activities identified by the previously cited user studies of reading: anno-

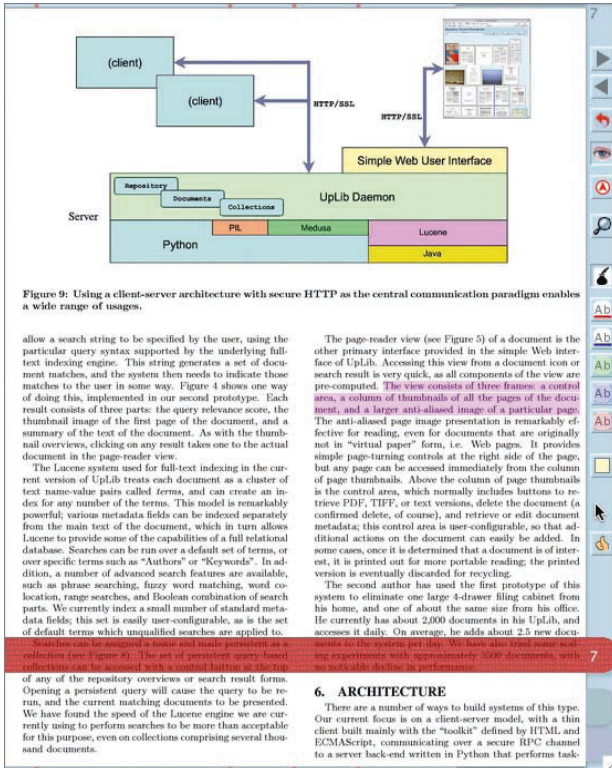


Fig. 1. A document open in an UpLib widget. The user has turned to page 7 by using a bookmark, and selected a region of text.



Fig. 2. Two-page viewing mode is useful for documents where the page design is spread across both pages

tation and document navigation. In addition, it supports two experimental reading modes, rapid serial visual presentation and animated dynamic highlighting.

Figure 1 shows a document open in a ReadUp widget; the widget itself may be embedded in a larger user interface, or simply in a stand-alone top-level window. The current page is displayed as an image in the center; either one or two pages (figure 2) may be displayed. On the side, a toolbar provides direct access to commonly used functions: page-turning, search, annotation, and bookmarks. Thin *page-edge* indicators at the top and bottom provide visual indication of where the reader is in the document. The toolbar and page-edge controls are optional; their display may be suppressed by the application. Both are “skinnable” so that their appearance can be adjusted by applications.

3.1 Document Navigation

The variety of ways in which people manipulate paper pages and navigate through documents is noted continuously in studies of readers. Many of the previously mentioned reading systems have attempted to support this kind of manipulation and navigation for virtual documents. ReadUp brings support for these reading activities into the toolkit. First, it provides several ways to simply turn pages, and provides a page-turn animation to provide feedback to the user on the event. Second, it supports operations on the document as a whole through overviews, bookmarks, and page-edges. Third, it supports search through the text of the document. Fourth, it extends some of the other page-turning mechanisms to support flipping back and forth between pages, something we refer to as *page twiddling*.

Turning pages. To turn from one page to the following page, a user can click on the “following page” button, an arrow pointing to the right, in the toolbar.

Similarly, the “preceding page” button will turn to the preceding page in the document. The user can also left-click on the right side of the page to advance to the following page, or left-click on the left side of the page to move to the preceding page. With a keyboard, the arrow keys can be used, as can the “Page Up” (preceding page) or “Page Down” (following page) keys. Emacs keybindings are also supported; “Control-V” turns to the following page, “Meta-V” or “Control-P” turns to the preceding page.

When the reader turns from one page to another, a page-turn animation can be drawn. This consists of a vertical blue bar that moves smoothly across the widget either from right to left on a “following page” turn, or from left to right on a “preceding page” turn. On one side of the bar, the old page is drawn, horizontally scaled to fit in the space. On the other side, the new page is drawn, also horizontally scaled. The bar moves across the widget in a specified page-turn animation time, typically 300 to 400 milliseconds, specified by the application. Using a page-turn animation time of 0 effectively disables page-turn animations. Our experience to date, though, is that readers like to have the animations, probably to establish visually the change in context associated with a page-turn.

Non-sequential navigation. The ReadUp widget also provides several mechanisms for document navigation, which can be thought of as occasions when non-sequential page-turns are desired. At the top and bottom of the page image is an optional narrow horizontal sub-widget called a *page-edge*. The page-edge provides context to the user, visual feedback on where they are in the document. It can also be used for direct access to another part of the document. Clicking on a location in either page-edge turns to the page at a corresponding proportional location in the document.

Another mechanism provided for navigation is a bookmark system. Three bookmark “ribbons” are provided for each document, in three different colors. They are drawn as semi-transparent textured strips that protrude as tabs into the side toolbar, intended to suggest to the reader a thin silk ribbon bound into the spine of the “book”. Each may be slid up and down the page, to the vertical position the reader finds most useful. A user “binds” a bookmark to a page by clicking on the tab of an unused bookmark when the document is open to the desired page. When a bookmark is bound – set to a particular page – its tab is drawn as a saturated end with a page number on it and a drop-shadow in the side toolbar (see figure 1). Unused bookmark tabs are drawn as faded, without page numbers. When the reader clicks on the exposed tab of a bound bookmark, the document flips to the page the bookmark is set to. Clicking the tab of a bound bookmark while at that bookmark’s page will unbind the bookmark, and it will disappear from that page; conceptually, it’s pushed to the back of the book.

A third mechanism provided is document overview. The reader can switch to an alternate view of the document, shown in figure 3(a), either by pressing the “Alt” key on a keyboard, or by clicking on the small thumbnail button in the side toolbar. In this display, each page is shown as a small numbered thumbnail image. The current page is highlighted. The reader can switch to a

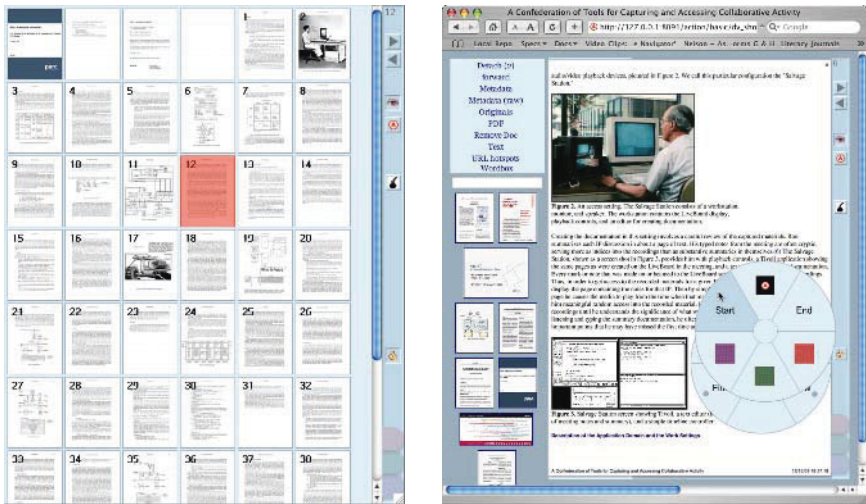


Fig. 3. (a) Page thumbnails shown in alternate view; the current page is highlighted. (b) A document displayed in a ReadUp applet in the UpLib Web interface; the reader is using the pie-menus to jump to the start of the document.

different page simply by clicking on it, then switch back to full-page mode by either releasing the “Alt” key, double-clicking the page thumbnail, or clicking the thumbnail button again. This is often a useful way to find a particular page with distinctive graphics on it.

Text search. A major advantage of document navigation in virtual documents is the ability to search for text in the contents of the document. ReadUp provides a search mechanism modelled on the incremental text search mode of GNU Emacs [19]. When the user presses the search button in the side toolbar, or presses “Control-S” on a keyboard, the document goes into search mode. The page is desaturated, a search window is shown in the upper right-hand corner of the page image, and matching text is highlighted using “pop-out” perceptual cues described in [20]. If text is selected before search mode is entered, that text becomes the search string; this makes it easy to select a word or phrase, then search for other occurrences of it in the document. Otherwise, the user enters text, one character at a time. As the search string is entered, the widget highlights all matches on the page. The “current” match is additionally highlighted with a pop-out color.

A search in progress is shown in figure 4(a). The search string, “thumbnail”, is shown in large letters in the search display. All strings matching that string are drawn with full contrast; the current match, in the abstract, is drawn with a violet pop-out highlight. The search button in the sidebar is overdrawn with a red right arrow to indicate that pressing it again will advance to the next match in the document, as will pressing “Control-S” on the keyboard while the search is active. If the next match is on a following page, the document will automatically be turned to that page.

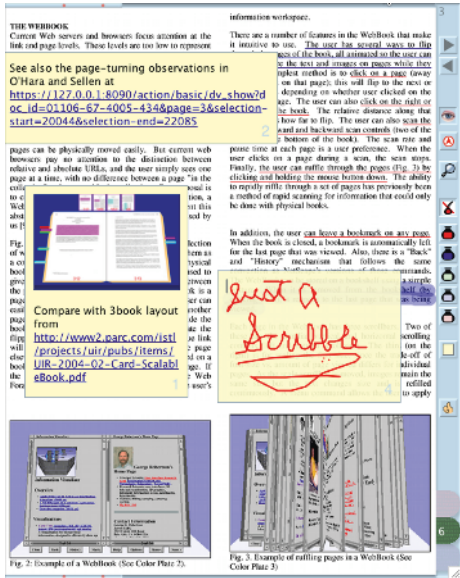
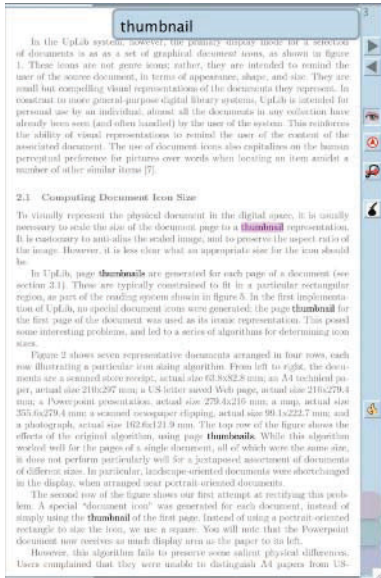


Fig. 4. (a) Text search in progress. All matches on the page are highlighted; current match is overdrawn with a pop-out color. (b) Annotations can be drawn directly on the page surface, or on note sheets, which also support text editing, pictures, and links. One of the links shown is to a part of another document in the reader's UpLib repository, the other to an "external" document on the Web.

The search window displays an indication of how many matches have been found so far, and whether or not the search has wrapped around from the end of the document back to the beginning. If the user advances to the end of the document, the search window turns red. If the current match is behind the search window, the search window turns translucent so that the highlighted match can be seen through it. The search can be terminated either by pressing "Control-G" on the keyboard, or by clicking on a portrait window. When the search is terminated, the current match is turned into a selection.

In addition to this conventional text search, other less common modes of search can be provided by application logic. In the standalone UpLib reader, for example, clicking on the UpLib icon in the side toolbar brings up a window which allows a search for more documents in the UpLib repository. Similarly, other standard searches are available through pie menu entries. One allows the reader to select a region of text, then converts that text passage into a search in the backing UpLib repository for other documents related to that text. The other search provides a similar service on Google, displaying search results in a Web browser window.

Page twiddling. It is common for readers to want to flip back and forth between two locations in a document. For instance, when reading a scholarly paper,

readers often flip to the citations at the back of the paper as they occur in the text, then back to the text that they were reading, as they progress through the paper. Readers also often want to compare items on two different pages of a document, or flip back and forth between two adjacent pages to read a complete thought split across the two pages.

ReadUp provides two different mechanisms to support this. When clicking on a page, the actions of the right and left mouse buttons have reverse effects. This makes it very easy to flip back and forth between two adjacent pages without moving the mouse. Some pen users find it easier to just click the “following page” and “preceding page” buttons in the side toolbar.

The second mechanism is used with non-adjacent pages. Whenever a non-sequential page-turn is made by using a bookmark, or clicking in a page-edge, or changing pages in the overview, a *flipback* button is provided in the side toolbar. It can be seen in figure 1, just below the page-turn buttons on the toolbar. By pressing it, the document is turned back to the previous location. This change is itself a non-sequential page-turn, so the flipback button still appears, and if pressed will reverse the action of the the previous press.

A third somewhat experimental mechanism is also provided. If the reader is using a mouse with a mouse wheel, turning the mouse wheel will turn several pages in succession, the exact number depending on the amount the mouse wheel was turned. This allows the reader to flip back and forth through a number of pages fairly easily.

3.2 Annotation

Annotation mode, seen in figure 4(b), can be turned off and on by clicking on the image of an inkpot and quill pen in the side toolbar. When on, inkpots are visible on the side toolbar, past annotations are visible on the page, and new annotations can be made. When using a pen, the most convenient way to annotate is to “dip” the pen in an inkpot, then write directly on the page. Structured highlighting or underlining of text passages can be accomplished by selecting the text region to be highlighted or underlined, then clicking on one of the annotation buttons in the side toolbar, which replace the inkpots when text is selected, as seen in figure 1. If a separate annotation or note is desired, clicking on the small yellow square button beneath the inkpots on the side toolbar will create a note sheet, which can be arbitrarily resized, moved around, and written on. Pages with annotations are indicated in the page-edges of ReadUp by small orange dots.

If a keyboard is available, the note sheets support a text editor, and text can be entered directly from the keyboard. In addition, note sheets support the inclusion of images and Web links. Links to selected areas of other UpLib documents can also be pasted into note sheets; when they are clicked on, the application can, for example, choose to open the other document in another window, with the selected region highlighted.

3.3 RSVP and ADH Modes

Two techniques for phrase-based reading are built into ReadUp, RSVP and ADH. Both segment the text of the document into short phrases using linguistic techniques. *Rapid serial visual presentation* (RSVP) [18] then presents each phrase in sequence in the center of the page image space at a user-controllable rate. *Animated dynamic highlighting* (ADH) [9], on the other hand, presents each phrase in context, with the rest of the page text heavily desaturated, again at a user-controllable rate.

4 Implementation Issues

Several issues were encountered during implementation of the the widget. Two of these merit separate discussion: management of the widget's graphics layout, and management of memory resources.

4.1 Graphics Layout

Many of our reading situations involve a laptop, either with a trackpad or an external mouse, or a tablet-PC, used with a pen and virtual keyboard of some sort. Laptops have poor vertical resolution, so the toolbar was placed at the side to conserve vertical pixels. Tablet-PCs have few physical buttons to augment the pen (typically only two are usable), so buttons were placed on the toolbar to provide functionality already available to readers with a keyboard. In addition to these issues, the widget had to be embeddable in a larger application, which meant that it could not count on having access to menubars or other input systems outside the bounds of the widget. Conventional Java pop-up menus could be used, but pie menus are more usable by pen readers, so a hierarchical pie menu system derived from [8] was included (see figure 3(b)). Similarly, output modes could not use space outside the bounds of the widget, so pop-up windows like that used for search (figure 4(b)) were allowed to carefully occlude part of the page image when necessary.

4.2 Resource Management

Because ReadUp is a toolkit widget, a single application may have many instances of it active at any one time. For instance, a workspace for writing may have many reference documents open at the same time. There are several memory-intensive data elements associated with the types of books ReadUp is designed for. ReadUp uses page images as its input, which can come from either a network source such as an UpLib, or from local image files. Each page of the book is available as both a large clipped anti-aliased page image, and as a smaller page thumbnail. In addition, information about the text of the page, including bounding boxes, parts of speech, and other metadata, for each word, is usually available. Finally, annotations on the page, which may including drawings, text, and images, can occupy significant amounts of memory.

This is especially true for Java applications, which must work within the somewhat constrained memory model imposed by the standard Java virtual machine design. A large text, such as a reference work, can overwhelm the virtual machine if all of its data is loaded at the same time. To address this situation, ReadUp contains a dynamic caching mechanism.

Each type of data element (large page images, small page images, text meta-data, and annotations) has a separate resource manager. Each resource manager implements the `ResourceLoader` interface, which provides a method called `getResource` to retrieve an instance of the resource, given the document, the page, and a selector which can be used to distinguish between different instances of the resource associated with the page, e.g. different annotations on the same page. Each manager maintains a cache of its resources. When another part of the ReadUp widget invokes `getResource` for a particular resource, the cache is examined, and if it contains the resource, it is returned. Otherwise (a situation known as a *cache miss*), the manager puts that resource on list of desired resources. This list is continually examined by a *resource loader thread* associated with the resource manager, which does nothing but bring desired resources into the cache. Once the request is available in the cache, the requesting part of ReadUp is informed with a callback, which also carries the requested object as a parameter.

The cache uses a *weak reference* [12] to point to the resource object. This is a type of reference which is not considered by the garbage collection algorithm, which means that if no other references to the resource exist, the garbage collector may reclaim the resource if it requires more memory. Resources which are in active use, such as a page image being rendered by the drawing subsystem of the ReadUp widget, have multiple “strong” references active, so they will not be garbage-collected. Unused resources, such as the images for pages not currently visible, may be reclaimed by the garbage collector if necessary. They will be automatically reloaded by the resource manager’s resource loader thread if they are later brought into use. Note that for resources freshly loaded in response to a cache miss, a strong reference to the resource is maintained by its use as a parameter to the callback method, which means that it may not be garbage-collected until the callback returns. This gives the callback code an opportunity to establish its own strong reference to the resource, if needed.

While ReadUp was explicitly developed for use with the UpLib system, it was also designed to be independent of that system. Resources can be loaded from arbitrary sources by providing appropriate resource loaders, which follow an abstract interface. Our standard loaders for use with UpLib load resources via network procedure calls, but other loaders can easily be constructed to load resources in other ways.

5 Current Use of the ReadUp Widget

The Web interface for UpLib has been redesigned to use ReadUp in a Java Plug-In applet. Figure 3(b) shows an example. The page thumbnails of the previous

Web interface have been replaced with document icons for other documents in the repository, allowing the reader to easily flip back and forth between documents, and an UpLib search window has been added. All of the aforementioned deficiencies of the previous Web interface have been remedied.

The ReadUp widget is already being used in other applications such as the corpus browser described in [3], and the ReadUp application, which is a simple search-and-display wrapper around the widget. Because the widget handles all the details of supporting the reader in the reading process, these applications are able to concentrate on collection navigation and management. In addition, the consistency of reading operations across the various applications lowers the learning curve for new users.

Acknowledgements

The UpLib system itself is the result of joint work with Ashok Popat at PARC. Many of our colleagues at PARC have contributed generously to our work on this project, notably Lance Good, Jeff Breidenbach, Eric Bier, and Stuart Card.

References

1. A. Adler, A. Gujar, B. L. Harrison, K. O'Hara, and A. Sellen. A diary study of work-related reading: design implications for digital reading devices. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 241–248. ACM Press/Addison-Wesley Publishing Co., 1998.
2. M. J. Adler and C. V. Doren. *How to Read a Book*. Touchstone Books, revised edition, 1972.
3. E. Bier, L. Good, K. Popat, and A. Newberger. A document corpus browser for in-depth reading. In *JCDL '04: Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries*, pages 87–96, June 2004.
4. S. K. Card, L. Hong, J. D. Mackinlay, and E. H. Chi. 3book: a scalable 3d virtual book. In *Extended abstracts of the 2004 conference on Human factors and computing systems (CHI)*, pages 1095–1098. ACM Press, 2004.
5. D. Chaiken, M. Hayter, J. Kistler, and D. Redell. The virtual book. Technical Report 157, Digital Equipment Corporation Systems Research Center, November 1998.
6. Y.-C. Chu, D. Bainbridge, M. Jones, and I. H. Witten. Realistic books: a bizarre homage to an obsolete medium? In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 78–86. ACM Press, 2004. http://www.nzdl.org/html/open_the_book/.
7. A. Dillon. Reading from paper versus screens: A critical review of the empirical literature. *Ergonomics*, 35(10):1297–1326, October 1992.
8. J. Hong. Java pie menus. World Wide Web, 2002. <http://www.cs.berkeley.edu/~jasonh/download/software/piemenu/>.
9. B. Janssen, O. Gurevich, and L. Karttunen. Animated dynamic highlighting. ACH/ALLC Conference 2005, June 2005. http://mustard.tapor.uvic.ca/cocoon/ach_abstracts/xq/pdf.xq?id=110.

10. W. C. Janssen. Document icons and page thumbnails: Issues in construction of document thumbnails for page-image digital libraries. In *ECDL 2004: Proceedings of the Eighth European Conference on Digital Libraries*, pages 111–121, 2004.
11. W. C. Janssen and K. Popat. UpLib: A universal personal digital library system. In *DocEng 2003: Proceedings of the ACM symposium on Document Engineering*, pages 234–242. ACM Press, November 2003.
12. R. Jones and R. Lins. *Garbage Collection : Algorithms for Automatic Dynamic Memory Management*. John Wiley and Sons, 1996.
13. C. C. Marshall. Annotation: from paper books to the digital library. In *DL '97: Proceedings of the second ACM international conference on Digital libraries*, pages 131–140. ACM Press, 1997.
14. K. O'Hara and A. Sellen. A comparison of reading paper and on-line documents. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 335–342. ACM Press, 1997.
15. K. O'Hara, F. Smith, W. Newman, and A. Sellen. Student readers' use of library documents: implications for library technologies. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 233–240. ACM Press/Addison-Wesley Publishing Co., 1998.
16. T. A. Phelps and R. Wilensky. The Multivalent browser: a platform for new ideas. In *DocEng '01: Proceedings of the ACM Symposium on Document Engineering*, pages 58–67, Atlanta, Georgia, 2001. ACM.
17. B. N. Schilit, G. Golovchinsky, and M. N. Price. Beyond paper: supporting active reading with free form digital ink annotations. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM Press/Addison-Wesley Publishing Co., 1998.
18. K. Sicheritz. Applying the rapid serial presentation technique to personal digital assistants, 2000. Master's Thesis, Department of Linguistics, Uppsala University.
19. R. M. Stallman. *GNU EMACS Manual*. Free Software Foundation, 2000.
20. B. Suh, A. Woodruff, R. Rosenholtz, and A. Glass. Popout prism: adding perceptual principles to overview+detail document interfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 251–258. ACM Press, 2002.
21. Sun Microsystems. *Java 2 Platform, Standard Edition, v 1.4.0: API Specification*, 2002.

The DSpace Open Source Digital Asset Management System: Challenges and Opportunities

Robert Tansley¹, MacKenzie Smith², and Julie Harford Walker²

¹ Hewlett-Packard Laboratories, 1 Cambridge Center, Cambridge, MA 02142
robert.tansley@hp.com

² MIT Libraries, 77 Massachusetts Ave, Cambridge, MA 02139
kenzie, jhwalker@mit.edu

Abstract. Last year at the ECDL 2004 conference, we reported some initial progress and experiences developing DSpace as an open source community-driven project [8], particularly as seen from an institutional manager's viewpoint. We also described some challenges and issues. This paper describes the progress in addressing some of those issues, and developments in the DSpace open source community. We go into detail about the processes and infrastructure we have developed around the DSpace code base, in the hope that this will be useful to other projects and organisations exploring the possibilities of becoming involved in or transitioning to open source development of digital library software. Some new challenges the DSpace community faces, particularly in the area of addressing required system architecture changes, are introduced. We also describe some exciting new possibilities that open source development brings to our community.

1 Introduction

DSpace is a digital asset management system, most commonly used as an institutional repository system by research universities. The system was initially developed by a joint HP and MIT Libraries development team. Since its release in November 2002, it has achieved widespread adoption; at the time of writing there are well over 100 DSpace instances running distributed among all of the continents in the world.

Both HP and MIT Libraries had the same original goal of building DSpace to discover what it takes to build and deploy a repository system to capture, preserve and disseminate an organisation's born-digital assets. It was clear from the start that this goal was not unique to HP and MIT Libraries, and that there are several compelling reasons for pursuing an open source development model:

- It was clearly not sustainable for HP and MIT Libraries to support the entire community of DSpace users. An open source model allows each organisation to customise the platform for their own particular local requirements, and to enable these customisations to be shared as appropriate;

- It is a focus around which a community of researchers and practitioners can form, exploring the areas of managing digital content and long-term preservation of digital material;
- A wider group of stakeholders and developers who understand the system ensures the longevity of the system and content stored within instances;
- It allows researchers, practitioners and developers to work together rather than as islands or silos;
- It provides those groups an opportunity to see the fruits of their work adopted and deployed by end users.

This paper describes how we have made the transition from regular co-located team-based development to a broader open source development model, in which the software source code is maintained and developed by the community around it as a whole, as opposed to being principally worked on by one particular team or organisation. We also describe some of the challenges and opportunities this brave new world presents.

2 The Transition to Open Source

The initial period of DSpace software development before the release of version 1.0 in November 2002 was a fairly typical software project. A co-located HP and MIT team of library staff and developers specified, designed and built a ‘breadth-first’ system, which, although greatly informed and assisted by members of MIT Libraries staff and MIT faculty ‘early adopters’, was a largely closed process.

Over the year following the release of version 1.0, the main change from the initial development period was that a great deal of feedback in terms of bug reports and functionality was received from a widening group of users. However, DSpace software development and maintenance was still a centralised process performed by the HP and MIT Libraries team; hence, although the source code was open and under an open source license, this was not really an open source development model. Although at the time this did attract some criticism, this is understandable and reasonable; one cannot simply remove access control on the source code for allcomers. From a very early stage, many universities libraries’ reputations and patrons were relying on the stability of the DSpace software.

The first signs that a real technical community was forming around DSpace appeared around technical support for installation and configuration problems. In the first days after the 1.0 release of the DSpace software, members of the HP/MIT development team had to answer every technical query in this area, or leave a question unanswered, which would soon lead to frustration. It was only a matter of a few months before many questions were being answered by members of the DSpace community outside of the HP/MIT team. This was very encouraging, as it demonstrated a willingness by members of the community to donate time and effort to help each other out. This ‘self-help’ technical support model is probably where the average DSpace user first encounters the difference between an open source system and a commercial system, and accordingly was the first area where the DSpace community really started to gel.

However, at the same time, the HP/MIT team realised that more needed to be done to support a move to real open source development, where development and maintenance of the code itself is owned by the community as a whole.

One key issue to address was perception and education. Most in the digital library community are used to being just consumers of open source software rather than actively involved; other important open source digital library projects, such as FEDORA[3] and the eprints.org software[2], are moving towards a community development model but are at a different point in their lifecycles as open source software projects. Others are concerned about transferring the copyright over work they produce to HP and MIT. Another issue was to do with finding and making the right use of technical infrastructure to support the community. The following sections describe the way we addressed these problems.

2.1 Copyright and Licensing

In addition to a change in mind set, open source presents a new set of legal issues to the digital library community. Typically, library professionals need to be familiar with copyright law as it relates to published printed material. The issues encountered around software licensing and copyright are quite different.

At the present time, the copyright on DSpace is jointly held by Hewlett-Packard and MIT. We currently require contributors to hand over copyright on their contributions to HP and MIT, so that the whole DSpace source code base has the same copyright holder. This has raised concerns and challenged institutions' policies regarding intellectual property. Some institutions are nervous about transferring copyright of work they have done to a commercial company, particularly to be further distributed via a 'commercial-friendly' license like the BSD license. Other institutions have intellectual property policies which either prohibit such transfers or require that they go through some review process. Many are concerned about losing 'credit' or recognition for their work. These are legitimate concerns which should be addressed.

The BSD license was chosen carefully. We wished to allow for vendors to play a role in the repositories area, and by employing the BSD license we hope to encourage the resources of those vendors to work with and benefit the DSpace community, instead of forcing them to create their own, competing systems. Keeping a heavily-modified version of DSpace up-to-date with the evolving core code base can be time-consuming. Contributing modifications back to the core code base so that they are part of that core means that the community as a whole can take over maintenance. This means that both commercial vendors and universities alike will benefit from contributing to the core code base. Thus, the BSD license encourages commercial vendors to become valuable contributors to the core code base, without removing their ability to develop separate components and services with commercial licenses.

We are already seeing the benefit of this; some commercial services are being built around DSpace. HP India offers a commercial service to set up DSpace instances at universities, and BioMed Central's Open Repository service offers set-up and ongoing hosting services. Both of these activities are benefiting the

DSpace community, by increasing the number of users, stakeholders and developers working with the DSpace platform.

The copyright transfer requirement has raised some concern for potential contributors, particularly since not all open source projects have this requirement. For example, the Eclipse project allows contributors to retain copyright for their contributions. There are two principal reasons for requiring copyright transfer: One is to simplify the legal process surrounding copyright or license infringements; a second is to enable the code base to be re-licensed and the copyright transferred in the future.

In general, sizeable open source projects need a legal entity to represent it in any situation where the project's copyright or license is infringed, or is alleged to have infringed some other license or copyright. More copyright holders on the code could be a real hindrance here, as all such holders may potentially have to be involved in such proceedings.

Additionally, as previously discussed [8,7] we are considering long-term stewardship of the DSpace platform and community outside of HP and MIT; however there are is no concrete plan or timeframe at the time of writing. In order to achieve change either the copyright owner of the DSpace code or the license, the consent of all existing copyright owners would be required. If all incoming contributions had separate copyright holders, this would rapidly become a time-consuming logistical challenge. Thus, in order to remain flexible in this regard, fewer copyright holders is better.

Although the fact that HP and MIT are copyright holders may have initially caused concern for potential contributors, a better understanding of the BSD license reduces this. Also, HP and MIT appear to have successfully communicated that they are merely stewards of the code and are not seeking to 'own' it; this, combined with the fact that HP and MIT fulfil the role of the required legal entity to own and license the code, means that the need to set up or join a third-party organisation or foundation for this purpose is not urgent. However, this is still the intention.

2.2 Development Roles

To address the perception that DSpace was still a centrally-developed, free but immutable 'product' from HP and MIT, in April 2004 we introduced a simple development structure around DSpace based on the Apache Foundation model. We defined a group of *committers*, so-called because they have authorisation to *commit* changes to the DSpace source code repository, which included members from outside of HP and MIT Libraries. HP and MIT Libraries both have just one representative in this group, demonstrating that we consider ourselves peers in this community. The group consisted of five members initially, now seven.

Others who contribute to DSpace are *contributors*; this is a role that requires no 'approval' or special status, anyone who helps out is automatically considered a contributor. It is important to note that people can contribute more than just code; experiences, technical support, bug reports, content, and documentation are all needed and welcome.

As we considered and implemented this change in the way DSpace was managed and developed, for a time, the pace of development slowed. This happened when the HP and MIT teams working on DSpace started to focus less on the core DSpace platform, and more on trying to develop the open source community and goals specific to their own research and deployment objectives. Suddenly, it was no one's full time job to look after the DSpace code base. Although neither HP nor MIT disengaged completely from this core maintenance and development process, the geographically dispersed DSpace committer group had to figure out how to progress things and leverage the wider community, while achieving the other objectives of their organisations, which was a significantly different situation from a focussed, co-located team.

However, no one steps up to fill a gap if no gap is there; this slowing may have been necessary to encourage people to step up and contribute. Previously, if someone had made a contribution when HP and MIT had several full-time developers working on the code, they may have seen their contribution lost in the noise somewhat. Now, contributions could be seen to have a real noticeable effect; this gives people a rewarding feeling of control. Further, it is noticed by others that HP and MIT are no longer the only developers (in fact HP and MIT developers are now a minority!), and this inspires others to consider contributing.

This change, combined with the development of the infrastructure and perception described in other sections, has resulted in the last year seeing a huge increase in the number of external contributions to DSpace, ranging from simple bug fixes to sizeable feature enhancements. The number of people who have worked on the DSpace code has increased from an initial five developers to around twenty-five at the time of writing, from varying organisations, and is constantly increasing. In addition to numerous bug fixes, some features developed outside of HP and MIT are:

- Customisable submission forms – each collection can have a different set of metadata entry fields
- Image thumbnails can be viewed in item display pages, in search results and while browsing title, date and author indices
- Users can add comments to items in DSpace, a little like the discussions that accompany news items on slashdot.org
- Support for LDAP authentication
- Support for internationalised Web user interface

Discussion around many further enhancements continues on the discussion lists, and many groups have indicated that they are engaged in working on those enhancements, so there is no reason to suppose that the number and pace of these enhancements will slow in the near future.

One technical challenge we face is managing those enhancements that involve updating the relational database schema in DSpace. Since applying such changes to an existing DSpace instance requires careful management and taking down that DSpace instance for a time, careful release management is required. To balance this required control with the open development model we have introduced the notion of deadlines for submitting contributions to be included in

a particular release; if a contribution is to be considered for inclusion in the next DSpace version, it must be submitted by a particular date. Then, at that date, we have a fixed set of updates to manage and include in a beta-testing cycle for the next release.

2.3 Infrastructure

It takes more than a Web page and good intentions to support an active open source community. It also needs actively managed collaboration infrastructure which allows a community to form and function cooperatively. It's also important not to have *too many* communication mechanisms available, as they become difficult to keep track of, and the community becomes fragmented. The DSpace community has found the following tools invaluable in this regard.

SourceForge provides many basic functions an open source project requires:

- A publicly accessible CVS repository for managing source code and documentation
- Bug and feature request tracking systems
- A 'patch' tracking system
- Mailing list server and archives

In addition, it is the place where the DSpace software can be downloaded. In general, SourceForge has proved sufficient for all of the above requirements.

Mailing lists are the fundamental means of communication between the geographically diverse community. There should be enough different mailing lists so that not everyone is swamped with traffic not relevant to them, but not so many that the communication is too disparate for a real community to form. It is also important to provide an in-road for those who are interested in the project, but don't necessarily want to sign themselves up to receive dozens of e-mails every day.

For DSpace we have found a good balance with a number of lists:

- A general-purpose, catch-all list for non-technical discussion around the platform and its application and announcements. This list has proved to be fairly low-traffic, which has the benefit that subscribers who are peripherally interested in DSpace are not put off by large volumes of posts. At the time of writing, this list has over 500 subscribers.
- A general technical support and discussion list for those deploying the system and performing local modifications. This tends to be a high-traffic list, and is also the area where the 'community in action' effect can be most felt. It is rare that a request for help on this list goes unanswered, although since support is offered voluntarily by list members as they have time and knowledge to do so, occasionally this does happen. This can leave the poster feeling frustrated or left out; however this is happening less and less, particularly as posters are becoming far more proficient at reporting problems with sufficient information for analysis (as opposed to simply, "I got an error, how do I fix it?") At the time of writing, this list has around 500 subscribers, with over 3,200 messages having been posted over its lifetime.

- A list for developers working on the DSpace platform itself. This is where details around bugs, new features, architectural issues and so forth are discussed. Although there is often overlap with the above technical support list, the discussion does tend to get more involved in this list, so it is useful to have this separate list for those who wish to get that bit more involved with DSpace. At the time of writing, this list has around 120 members and over 1,000 posts.
- A closed list for the DSpace Committer group. This list is primarily used to discuss matters to do with policy and procedures, where members feel a public post is not appropriate. However, in general, discussion happens on the above developer list; as with the source code itself, these processes can benefit from review and input from a wide community.
- Several ‘special interest group’ lists related to particular areas of application of the DSpace platform. The traffic on these lists is low at the time of writing. Although such lists could potentially serve to fragment the community, in fact they are useful for involving people at the periphery of DSpace, perhaps just interested in one aspect, such as digital preservation. These lists provide a useful level for these individuals to become involved with DSpace, without being exposed to a deluge of low-level or orthogonal technical message traffic.

The DSpace Wiki is proving an invaluable tool. Although the archives of the various e-mail lists above contain a lot of information, in general finding relevant information in them is somewhat awkward and time-consuming due to their volume and unstructured nature. The Wiki has allowed information to be collected and disseminated that is far easier for people to access, and in an expedient and collaborative way not possible with a typical Web site with a small group of maintainers.

In particular it has enabled the creation of some valuable resources:

- A list of projects and people working on DSpace. Anyone can (and is encouraged to) add their work to this list. This means those interested in a particular area of DSpace can find other people working on the same area, encouraging collaboration and minimising duplication of effort. It additionally gives an idea of the amount and breadth of work happening on and around DSpace.
- A list of DSpace instances, to which people can add their own. Seeing a large number of organisations actively using DSpace gives people confidence in the platform.
- Guides and FAQs for developing with the DSpace software. As the processes and practices for this evolve quickly, the Wiki provides a useful place for up-to-the-minute information on this, which people can correct and annotate with experience.

Two features of the Wiki have proved essential—access control, and automatic e-mail notification of updates. Although the Wiki is essentially open for anyone to edit, this was abused for a time as ‘spam’ was repeatedly posted on the front page of the Wiki. Fortunately, very minimal application of access control (merely securing the front page) has eliminated this.

Every subscriber to the DSpace developer email list receives notification whenever a page on the Wiki is updated. Although this received some initial resistance due to an increase in email traffic, it serves a very useful purpose by alerting people not only to specific changes in the Wiki, but also to the fact that the Wiki is there and being actively used. This last point has been key to the success of the Wiki.

The dspace.org Web site is a stable reference point for the project, from which the resources described above can be reached. It also provides some background material about the project, and a considerable amount of guidance from the experience of MIT for universities on the non-technical aspects of creating an institutional repository service using DSpace. This site gets about 25,000 visits a month.

These non-technical implementation aspects, while important, appear a far less active and dynamic area of discussion on the DSpace mailing lists. This is probably because things like assembling resources and defining policies take a lot longer to change than source code!

3 Moving Forward: New Challenges and Opportunities

The developer community around DSpace is now starting to function smoothly. We have seen a considerable increase in technical activity around DSpace recently. New patches (code contributions) are received every week. In addition to these feature enhancements and bug fixes that arise from a particular organisational need, various significant research projects around DSpace are underway.

The DSpace/SRB integration project [5] at the San Diego Supercomputer Center and MIT is investigating using Grid storage technologies with DSpace. Specifically, they are using SDSC's Storage Resource Broker technology, although the intention is to make storage 'pluggable' so that a variety of storage mechanisms can be used, from simple file systems to large-scale distributed storage.

SIMILE is a joint MIT and W3C project [6] looking at employing Semantic Web and RDF technologies to support heterogenous metadata storage and indexing in DSpace.

DSpace@Cambridge is looking at various long-term digital preservation issues, such as managing file formats and auditing processes, and also working on support in DSpace for e-learning activities.

CWSpace [1] project to investigate archiving OpenCourseWare course materials in DSpace, as well as the related standards and interoperability protocols to support reuse of those materials in course management systems and collaborative learning environments.

The University of Minho [4] in Portugal are experimenting and developing several add-ons to DSpace, including ontology support for classifying items and tools to visualise relationships between content and researchers.

In addition to projects such as those listed above which explicitly involve DSpace as part of the deliverables, many other projects have adopted DSpace as a platform on which to base prototypes and research, such as the EU-funded Digital Academic Repositories (DARE) project in the Netherlands and the Australian Partnership for Sustainable Repositories (APSR) project of which the Australian National University is the lead institution.

Open source provides these research projects with an exciting new opportunity to see their results directly reflected in a functioning system that is deployed at over a hundred organisations. Their work will be visible and useful to the thousands of end users of these systems.

Along with this new opportunity comes a new challenge: coordinating the results of the various projects. There is a need for these projects to collaborate so that effort is not unduly duplicated, and more importantly, they move the platform as a whole in a consistent and beneficial direction. This is driving some work looking at improving the underlying architecture of the system.

3.1 Architectural Evolution: DSpace 2.0

One of the reasons underlying DSpace's widespread adoption is that it is an end-to-end application. Every basic function required to deploy and operate an institution repository or similar system is present, though not necessarily to a sophisticated degree. This approach was taken because it was not known *a priori* exactly where 'development dollars' would best be spent; such knowledge comes from the experience of building, deploying and operating the system. This, coupled with the unanticipated extent of adoption of DSpace, has resulted in the need for some evolution of the underlying system architecture of DSpace in addition to the development of added functionality, particularly in two areas.

Storage — In the current architecture, all metadata is in a relational database, and all content bitstreams are in the file system on the server. This makes certain preservation-related activities complex, including:

- Backups — coordinating the backups of bitstream storage and relational database. A backup must contain a snapshot of the relational database and the bitstream store in a consistent state; i.e. the contents of the bitstream store backup must be as the data in the relational database expects. To make a reliable back up, one needs to take a snapshot of both the database and the bitstream store at one moment in time, and this involves freezing both for the duration of the backup process.
- Auditing — Although it is simple to audit the bitstreams in the bitstream store, auditing the metadata and the structure, for example checking that all items are present and correct, is not.
- Replication/distribution — Any replication or 'sharing' of content, metadata etc. will require processing at every stage to extract metadata from the relational database and package it with the appropriate content.

Modularity — The diverse community using DSpace needs to be able to deploy different flavours of DSpace, for example one that uses a particular

identifier scheme; researchers and developers need to be able to develop and experiment with DSpace without unduly affecting others or being overly encumbered with the maintenance effort involved in keeping their customisations up to date with the core DSpace code base.

To address these requirements, an evolved DSpace architecture dubbed ‘DSpace 2.0’ was presented at the first DSpace user group meeting held in Cambridge, MA, USA in March 2004 [9]. This updated architecture proposed three major areas of refactoring the DSpace architecture:

Asset Store — a component which stores Archival Information Packages, consisting of the serialised metadata and bitstreams in a DSpace item. This means objects in DSpace are more self-contained in the system, enabling easier backup, auditing and replication. The metadata in these AIPs may be replicated in a relational database performance reasons, however the AIPs are the ‘authoritative’ version of the metadata and content, with the relational database copy being considered a cache.

Module Layer — The various components of the DSpace system are more cleanly modularised; they interact only via defined APIs, and an individual instance can plug in whatever implementation of each API they like.

Web user interface — This needs to be modularised, as most customised functionality must be reflected in the user interface. The Apache Cocoon publishing framework was proposed as a potential candidate for addressing this, as Servlet/JSP has proven to be wanting in this particular area.

This design proposal was largely welcomed by the open source community; however, it wasn’t a full specification. Immediately returning to a ‘closed shop’ style of development would cancel out the benefits of having developed an open source community around DSpace, so we are presented by a number of challenges:

- Establishing consensus around technical issues when there is no single ‘technical lead’ or decision maker.
- Finding resources to do the (considerable) required development work.
- A large existing user base has invested in understanding and modifying the existing DSpace system. Introducing a completely new version of the system could alienate some of these people; at best, it would divide the resources of the DSpace community between essentially two different systems.

For this architectural work, a new development model has started to emerge, based around prototyping. Developers at institutions with a vested interest in a particular aspect of this refactoring produce a small prototype demonstrating a particular approach to one aspect of the architecture.

This breaks down the refactoring task into manageable pieces. Building and sharing these prototypes brings to light the various underlying technical issues, many of which are not apparent during prior design work and discussion. Building such prototypes also represents a relatively small developer commitment;

they certainly do not require the level of consensus, coordination and resources that constructing a complete system does.

Various prototypes have been built thus far:

- Various prototypes for the DSpace 2.0 asset store, built by individuals at HP, MIT and Cambridge University
- A prototype user interface based on Apache Cocoon, built at Australia Nation University
- A prototype ingestion system based on Apache Cocoon, also built at Australia Nation University

These prototypes may mature and be folded into the DSpace code base directly, or they may simply serve as ‘proofs of concept’ serving as input to further development efforts. Where more than one prototype represents a possible route forward, the task of achieving consensus should be facilitated by the availability of functioning code.

This prototyping approach provides a path forward in achieving the DSpace 2.0 architecture; it does not address all of the challenges mentioned above. The problem of how to introduce these architectural changes with a large existing user base still remains. We may need to pursue a gradual, evolutionary approach; it seems unlikely that the current community would be able to support two separate versions of the system.

4 Conclusion

With DSpace we have made considerable progress in moving from a typical, closed team style of development to a wider, open source model where the responsibilities for maintenance and development are held by the community as a whole. This brave new world has presented us with many challenges, and it has not been an easy path; for a time, the pace of development of the DSpace platform slowed considerably. However, this has been a necessary phase. It takes time to establish processes and infrastructure to support open source development; slowing development is also a factor in encouraging members of the community who are used to being only consumers to become active contributors to the effort.

We have made considerable progress in addressing this: The fact that DSpace contributors from the wider community outnumber those inside HP and MIT is a testament to the fact that we are functioning as an open source community. DSpace now has a key attribute of open source projects: *vitality*. These days, open source projects are judged more in terms of the activity around them than the merits of the technology itself; a system which still needs work but has a bustling community around it is likely to be a better long-term bet than a more technically developed system with no visibly active community.

While we have overcome some of the hurdles we reported last year, we face some new challenges. A potential emerging challenge is around research projects looking at significant architectural directions and decisions for DSpace. How do we manage the case when consensus of the community as a whole is to move in a

different, perhaps contradictory direction? How do we still engage the resources of that project, which is obliged to explore its own direction? Will such projects regard the fact that their proposed directions were not adopted as 'failure'?

An additional challenge exists around deeper architectural development: Since DSpace version 1.0 was built as a 'breadth-first' system, it was not perfect in every regard; some work on the underlying architecture to facilitate preservation and modular development are needed. An evolved DSpace architecture to address these requirements has been proposed and accepted; however, making large-scale changes in the system in an open source environment with a large number of existing users is proving a challenge. We are making considerable progress in this regard, as the open source community breaks down the problem and constructs small-scale prototypes of various aspects. This provides us with a grounding for discussion of the technical issues, and a concrete basis around which to build consensus.

As well as new challenges, it is also becoming clear that the open source model of development around DSpace is providing researchers with an exciting new way to see the fruition of their labour. Although publications, presentations and prototype systems are useful, too many good ideas come to a halt at this stage. The DSpace open source platform provides a means for these researchers to see their work deployed and used by a wide audience in a way that was difficult or impossible for this community before.

Although some challenges remain, the open source model of development brings numerous new and exciting opportunities to the digital library community.

References

1. CWSpace. <http://cwspace.mit.edu/>.
2. The eprints.org software. <http://www.eprints.org/>.
3. The flexible extensible digital object and repository architecture project (FEDORA). <http://www.fedora.info/>.
4. DSpace Dev @ University of Minho.
<http://dspace-dev.dsi.uminho.pt:8080/en/welcome.jsp>.
5. Reagan Moore, Richard Marciano, MacKenzie Smith, and Brian E. C. Schottlaender. NARA supplement to the NPACI collaboration: Integrating data management with data grids. <http://libnet.ucsd.edu/nara/>, September 2004.
6. SIMILE: Semantic interoperability of metadata and information in unLike environments. <http://simile.mit.edu/>.
7. MacKenzie Smith. DSpace user group meeting summary and outcomes. <http://www.dspace.org/conference/meetingsummary.html>, March 2004.
8. MacKenzie Smith, Richard Rodgers, Julie Walker, and Robert Tansley. DSpace: A year in the life of an open source digital repository system. In *Proc. 8th ECDL*, pages 38–44, Bath, UK, September 2004.
9. Robert Tansley. DSpace 2.x architecture roadmap.
<http://www.dspace.org/conference/presentations/architecture.ppt>.

File-Based Storage of Digital Objects and Constituent Datastreams: XMLtapes and Internet Archive ARC Files

Xiaoming Liu¹, Lyudmila Balakireva¹,
Patrick Hochstenbach², and Herbert Van de Sompel¹

¹ Research Library, Los Alamos National Laboratory,
Los Alamos, NM, US 87544

{liu_x, ludab, herbertv}@lanl.gov

² University Library, Ghent University,
Rozier 9, B-9000 Ghent, Belgium
Patrick.Hochstenbach@ugent.be

Abstract. This paper introduces the write-once/read-many XMLtape /ARC storage approach for Digital Objects and their constituent datastreams. The approach combines two interconnected file-based storage mechanisms that are made accessible in a protocol-based manner. First, XML-based representations of multiple Digital Objects are concatenated into a single file named an XMLtape. An XMLtape is a valid XML file; its format definition is independent of the choice of the XML-based complex object format by which Digital Objects are represented. The creation of indexes for both the identifier and the creation datetime of the XML-based representation of the Digital Objects facilitates OAI-PMH-based access to Digital Objects stored in an XMLtape. Second, ARC files, as introduced by the Internet Archive, are used to contain the constituent datastreams of the Digital Objects in a concatenated manner. An index for the identifier of the datastream facilitates OpenURL-based access to an ARC file. The interconnection between XMLtapes and ARC files is provided by conveying the identifiers of ARC files associated with an XMLtape as administrative information in the XMLtape, and by including OpenURL references to constituent datastreams of a Digital Object in the XML-based representation of that Digital Object.

1 Introduction and Motivation

Digital Library architectures that are concerned with long-term access to digital materials face interesting challenges regarding the representation and actual storage of Digital Objects and their constituent datastreams. With regard to representation of Digital Objects, a trend can be observed that converges on the use of XML-based complex object formats such as the MPEG-21 Digital Item Declaration Language [1] or METS [2]. In these approaches, the Open Archival Information System [3] Archival Information Package (OAIS AIP) that represents

a Digital Object is an XML-wrapper document that contains a variety of meta-data pertaining to the Digital Object, and that provides the constituent datastreams of the Digital Object either By-Value (base64-encoded datastream inline in the XML-wrapper) or By-Reference (pointer to the datastream inline in the XML-wrapper). This choice for XML is not surprising. Indeed, both its platform-independence nature and the broad industry support provide some guarantees regarding longevity or, eventually, migration paths. Moreover, a broad choice of XML processing tools is available, including tools that facilitate the validation of XML documents against schema definitions that specify compliance with regard to both structure and datatypes.

However, the choice of an XML-based AIP format is only part of the solution. The Digital Objects - represented by means of XML-wrapper documents - and their constituent datastreams still need to be stored. With this respect, less convergence is observed in Digital Library architectures, and the following approaches have been explored or are actively used:

- Storage of the XML-wrapper documents as individual files in a file system: On most operating systems, this approach is penalized by poor performance regarding access, and especially back-up/restore. Also, the OAIS reference model recommends against the storage of Preservation Description Information and Content Information using directory or file-based naming conventions.
- Storage of the XML-wrapper documents in SQL or native XML databases: This approach provides a flexible storage approach, but it raises concerns for long-term storage because, in database systems, the data are crucially dependent on the underlying system.
- Storage of the XML-wrapper documents by concatenating many such documents into a single file such as tar, zip, etc.: This approach is appealing because it builds on the simplest possible storage mechanism - a file - and it alleviates the problems of the “individual file” approach mentioned before. However, off-the-shelf XML tools are not efficient to retrieve individual XML-wrapper documents from such a concatenation file.

The Internet Archive has devised the ARC file [4], a file-based approach to store the datastreams that result from Web crawling. In essence, an ARC file is the concatenation of many datastreams, whereby a separation between datastreams is created by a section that provides - mainly crawling-related - metadata in a text-only format. Indexing tools are available to allow rapid access to datastreams based on their identifiers. While the file-based approach to store a collection of datastreams is attractive, the ARC file format has limited capabilities for expressing metadata. Even the result of the ongoing revision of the ARC file format, in which the authors are involved, will probably not allow expressing the extensive metadata that is typical for Archival Information Packages in Digital Libraries. By all means, it is not clear how various constituent datastreams of a Digital Object could be tied together in the ARC file format, or how their structural relationships could be expressed. Moreover, ARC files do

not provide the validation capabilities that are part of what makes XML-based representation and storage attractive.

In this paper we introduce a representation and storage approach for Digital Objects that was pioneered in the aDORe repository effort of the Research Library of the Los Alamos National Laboratory (LANL). The approach combines the attractive features of the aforementioned techniques by building on two interconnected file-based storage approaches, XMLtapes and ARC files. These file formats are proposed as a long-term storage mechanism for Digital Objects and their constituent datastreams. The proposed storage mechanism is independent of the choice of an XML-based complex object format to represent Digital Objects. It is also independent of the indexing technologies that are used to access embedded Digital Objects or constituent datastreams: as technologies evolve, new indexing mechanisms can be introduced, while the file-based storage mechanism itself remains unchanged.

2 Representing Digital Objects

Over the last 2 years, the Digital Library Research and Prototyping Team of the LANL Research Library has worked on the design of the aDORe repository architecture [5] aimed at ingesting, storing, and making accessible to downstream applications a multi-TB heterogeneous collection of digital scholarly assets.

As is the case in most Digital Libraries, assets stored in aDORe are *complex* in the sense that they consist of multiple individual datastreams that jointly form a single logical unit. That logical unit can, for example, be a scholarly publication that consists of a research paper in PDF format, metadata describing the paper expressed in XML, and auxiliary datastreams such as images and videos in various formats, including TIFF, JPEG and MPEG. For reasons of clarity, this paper will refer to an asset as a *Digital Object*, and to the individual datastreams of which the asset consists as *constituent datastreams*. The complex nature of the assets to be ingested into aDORe led to an interest in representing assets by means of XML wrappers, which itself resulted in the selection of the MPEG-21 DIDL as the sole way to represent the asset by means of XML documents called DIDL documents. The actual use of the MPEG-21 DIDL in aDORe is described in some detail in the slightly outdated [6] and the more recent [7]. Although this paper will illustrate the XMLtape/ARC storage mechanism for the case where MPEG-21 DIDL is used to represent Digital Objects, it will become clear that the approach is independent of the choice of a specific XML-based complex object format. Hence, it could also be used when Digital Objects are represented using METS or IMS/CP [8].

An important, OAIS-inspired, characteristic of the aDORe environment is its write-once/read-many strategy. Indeed, whenever a new version of a previously ingested Digital Object needs to be ingested, a new DIDL document is created; existing DIDL documents are never updated or edited. The distinction between multiple versions of a Digital Object is achieved through the use of 2

types of identifiers that are clearly recognizable, and expressed as URIs in DIDL documents:

Content Identifiers. Content Identifiers corresponds to what the OAIS categorizes as Content Information Identifiers. Content Identifiers are directly related to identifiers that are natively attached to Digital Objects before their ingestion into aDORe. Indeed, in many cases such Digital Objects, or their constituent datastreams, have identifiers that were associated with them when they were created or published, such as Digital Object Identifiers [9] for scholarly papers. Different versions of a Digital Object have the same Content Identifier.

Package Identifiers. A DIDL document that represents a Digital Object functions as an OAIS AIP in aDORe. During the ingestion process, this DIDL document itself is accorded a globally unique identifier, which the OAIS categorizes as an AIP Identifier. Values for Package Identifier are constructed using the UUID algorithm [10]; they are expressed as URIs in a reserved sub-namespaces of the 'info:lanl-repo/' namespace, which the LANL Research Library has registered under the info URI Scheme [11].

A separate component in the aDORe architecture, the Identifier Locator, keeps track of all versions of a Digital Object.

3 Storing and Accessing Digital Objects in XMLtapes and ARC Files

The aDORe environment shares two important characteristics with the Internet Archive:

- Data feeds in aDORe are typically received in batches, each of which can contain anywhere between 1,000 and 1,000,000 Digital Objects.
- Ingestion of a previously ingested Digital Object does not result in editing of that previously ingested version, but rather to a from-scratch ingestion of the new version.

These characteristics suggest that a file-based, write-once/read-many storage approach should be as appealing to aDORe as it is to the Internet Archive. However, Internet Archive ARC files have only limited capabilities to express metadata pertaining to datastreams and to the ingestion process, and they have no obvious way to express structure of a Digital Object with multiple constituent datastreams. Therefore, in aDORe, an approach has been devised that combines two interconnected file-based storage mechanisms: XMLtapes and ARC files.

3.1 XMLtapes: File Storage of XML-Based Representations of Digital Objects

An XMLtape is an XML file that concatenates the XML-based representation of multiple Digital Objects. In the aDORe implementation of the XMLtape, the

XML-based representations of Digital Objects are DIDL documents compliant with the MPEG-21 DIDL standard. In order to keep these DIDL documents small and hence easy to process, they typically contain:

By-Value. The metadata pertaining to the Digital Object, its constituent datastreams, and the ingestion process.

By-Reference. The constituent datastreams of the represented Digital Object. The embedded reference in the DIDL document points to the datastream that is stored in an ARC file that is associated with the XMLtape. The nature of the reference and the access mechanism will be explained in Section 3.3 and Section 3.4, respectively.

The structure of XMLtapes is defined by means of an XML Schema [12]:

- An XMLtape starts off with a section that allows for the inclusion of administrative information pertaining to the XMLtape itself. Typical information includes provenance information of the contained batch of Digital Objects, identification of the processing software, processing time, etc.
- The XMLtape-level administrative section is followed by the concatenation of records, each of which has administrative information attached to it. While allowing for the inclusion of a variety of record-level administrative information, the XMLtape has two strictly defined administrative elements: the identifier and creation datetime of the contained record. This allows for the use of a generic XMLtape processing tool that is independent of the nature of the actual included records. In aDORe, these strictly defined administrative information elements translate to the Package Identifier and the creation datetime of the DIDL document that is a record in the XMLtape.
- The records provided in an XMLtape can be from any XML Namespace. In aDORe, they are DIDL documents compliant with the MPEG-21 DIDL XML Schema [13].
- The XMLtape itself is a valid and well-formed XML file that can be handled by off-the-shelf XML tools for validation or parsing.

In order to interpret an XML file, it is generally necessary to parse and load the complete file. In case of XMLtapes, such an approach would forbid fast retrieval of the embedded XML documents. Therefore, in order to optimize access, two indexes are created. The indexes correspond with the mandatory record-level administrative information, and have identifier and creation datetime of the embedded records as their respective keys. As will be explained in Section 3.5, these indexes facilitate OAI-PMH access to the XML documents contained in the XMLtape. In addition to these identifier and datetime keys, each index stores the byte-offset and byte-count per matching record. When retrieving a record from an XMLtape, first a lookup in an index file is required to fetch a record position, followed by a seek into the XMLtape to return the required record.

3.2 ARC Files: File Storage of Constituent Datastreams of Digital Objects

In some scenarios, it can make sense to physically embed certain constituent datastreams of a Digital Object in the DIDL document that is contained in an XMLtape. For example, embedding descriptive metadata or image thumbnails may improve access speed for downstream applications. However, in other scenarios, such embedding is neither optimal nor realistic. Indeed, the mere size of a constituent datastream, worsened by the required base64 encoding, leads to large DIDL documents that may cause serious XML processing challenges at the time of dissemination.

The ARC file format [4] is used by Internet Archive to store datastreams resulting from large-scale web crawling. The ARC file format is structured as follows:

- An ARC file has a file header that provides administrative information about the ARC file itself.
- The file header is followed by a sequence of document records. Each such record starts with a header line containing some, mainly crawl-related, metadata. The most important fields of the header line are the URI of the crawled document, the timestamp of acquisition of the data, and the size of the data block that follows the header line. The header line is followed by the response to a protocol request such as an HTTP GET.

Tools such as those from `netarchive.dk` [14] are available to generate and consult an index external to the ARC file that facilitates rapid access to contained records, using their URI as the key. As will become clear from Section 3.3, these tools play a core role when connecting XMLtapes and associated ARC files.

3.3 Associating ARC Files with an XMLtape During the Ingestion Process

Both the XMLtape and its associated ARC files are created during the ingestion process. An insight in the ingestion flow is given here:

- When a feed of Digital Objects is obtained from an information provider, the ingestion process creates a DIDL document per obtained Digital Object; each DIDL document receives a globally unique Package Identifier.
- Typically, all DIDL documents for a given batch are stored in a single XMLtape that can easily store over 1,000,000 DIDL documents. An XMLtape itself also receives a globally unique XMLtape Identifier.
- Depending on the size of the constituent datastreams of the Digital Objects in a given feed, one or more ARC files are created during the ingestion process. Each ARC file is given a globally unique ARC file Identifier, and ARC files are associated with the XMLtape by including these ARC file Identifiers in the XMLtape-level administrative section.
- For each DIDL document written to the XMLtape:

- Each constituent datastream of the represented Digital Object is accorded a globally unique Datastream Identifier that has no relation to the aforementioned Package Identifiers or Content Identifiers.
 - The constituent datastream is written to an ARC file; the URI field of the ARC file record header receives the Datastream Identifier as its value.
 - A reference to the constituent datastream is written in the DIDL document. Core elements in this reference are the ARC file Identifier and the Datastream Identifier. As will be explained in the next Section, these references are encoded in a manner compliant with the NISO OpenURL standard [15].
- Indexes are created for both the XMLtape and its associated ARC files:
 - For the XMLtape, two indexes are created, with the Package Identifier and the creation datetime of DIDL documents as their respective keys.
 - Per ARC file, an index is created that has the Datastream Identifier as its key.
 - All globally unique identifiers accorded during the ingestion process are created based on the UUID algorithm [10].

3.4 Adding Protocol-Based Access to XMLtapes and ARC Files

The features described in the previous Sections allow for a persistent standards-based access to both file-based storage mechanisms.

Each XMLtape is exposed as an autonomous OAI-PMH [16] repository with the following characteristics (Protocol elements are shown in **bold**):

- It has a **baseURL(XMLtape)**, which is an HTTP address that contains the XMLtape Identifier to ensure its uniqueness.
- Contained **records** are DIDL documents only.
- The **identifier** used by the OAI-PMH is the Package Identifier.
- The **datestamp** used by the OAI-PMH is the creation datetime of the DIDL document.
- Access based on **identifier** and **datestamp** is enabled via the 2 aforementioned indexes created per XMLtape.
- The only supported metadata format is DIDL, with **metadataPrefix** DIDL, defined by the MPEG-21 DIDL XML Schema.
- The supported OAI-PMH harvesting **granularity** is seconds-level.

Each ARC file is exposed as an OpenURL Resolver:

- The OpenURL Resolver has a **baseURL(ARC file)**, which is an HTTP address that contains the ARC file Identifier to ensure its uniqueness.
- References embedded in the DIDL documents are compliant with the NISO OpenURL standard [15]. As a matter of fact, the reference uses the HTTP-based, Key/Encoded-Value Inline OpenURL. The Referent of this OpenURL is a datastream stored in an ARC file, and this datastream is described on the OpenURL by means of its Datastream Identifier.
- The sole service provided by the OpenURL Resolver is delivery of the datastream referenced on the OpenURL. This service is enabled by the index that is created per ARC file.

3.5 Accessing Digital Objects and Constituent Datastreams

This Section explains how Digital Objects and constituent datastreams are accessed in the aDORe environment in which the XMLtape/ARC approach is used as file-based storage mechanism. Figure 1 is provided to support a better understanding of the flow. In what follows, protocol elements are shown in **bold**, while argument values are shown in *italic*.

- In a typical scenario, an agent requests a Digital Object from aDORe by means of its Content Identifier. The Identifier Locator, not described in this paper, contains information on the locations of all version of a Digital Object with a given Content Identifier. When queried, it returns a list of Package Identifiers of DIDL documents that represent the given Digital Object, and for each returned Package Identifier the OAI-PMH **baseURL**(XMLtape Identifier) of the XMLtape in which the DIDL document resides.
- Next, the requesting agent selects a specific version of a Digital Object, thereby implicitly selecting a specific XMLtape Identifier and the OAI-PMH **baseURL**(XMLtape Identifier) of the XMLtape in which the chosen DIDL document resides. This DIDL document can be obtained using the OAI-PMH request:

baseURL(XMLtape Identifier)? **verb=GetRecord** &
identifier=Package Identifier&
metadataPrefix=didl

- Issuing this OAI-PMH request results in a look-up of the Package Identifier in the identifier-based index that was created for the targeted XMLtape. This look-up reveals the byte-offset and byte-count of the required DIDL

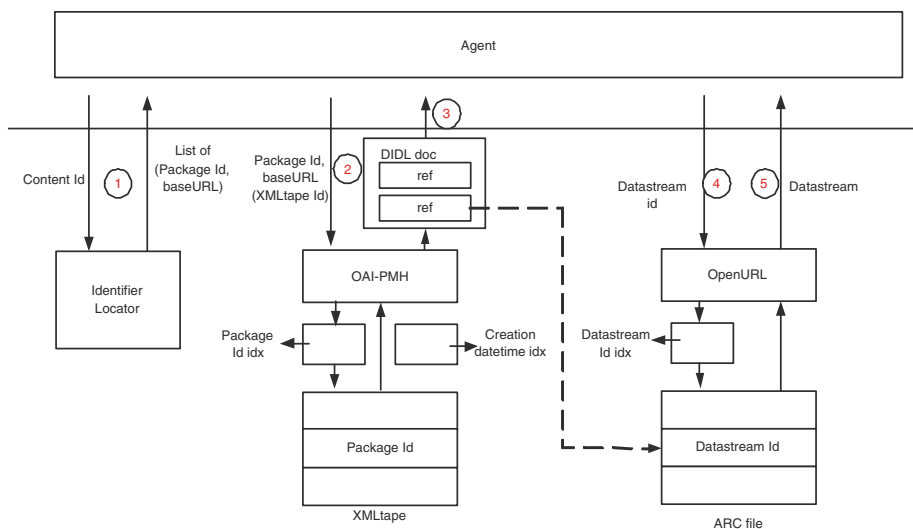


Fig. 1. Accessing a Digital Object stored in XMLtape/ARC

document in the XMLtape. Given this information, a process can access the DIDL document in the XMLtape and return it to the agent.

- Having obtained a representation of the requested Digital Object, the requesting agent may decide to next request a constituent datastream. Because such datastreams are included in the DIDL document By-Reference, selection of a specific datastream is equivalent to selecting the OpenURL that references it. This OpenURL is as follows:

**baseURL(ARC file Identifier)? url_ver=Z39.88-2004 &
rft_id=Datastream Identifier**

- Issuing this OpenURL request results in a look-up of the Datastream Identifier in the URI-based index that was created for the targeted ARC file. This look-up reveals the byte-offset and byte-count of the required datastream in the ARC file. Given this information, a process can access the datastream in the ARC file and return it to the agent.

3.6 Implementation

All XMLtape and ARC file components are implemented in Java. Due to the standards-based approach, several off-the-shelf components have been used. The XMLtape indexes are implemented with Berkeley DB Java Edition [17], while OAI-PMH access is facilitated by OCLC's OAICat software [18] which, in collaboration with OCLC, was extended to support access to multiple OAI-PMH repositories in a single installation. Creation, indexing and access to ARC files are implemented using the netarchiv.dk toolset [14].

The performance and scalability of ARC files are demonstrated by the Internet Archive and its Wayback Machine, which stores more than 400 terabytes of data. The performance of the XMLtape solution depends on the choice of the underlying indexing and retrieval tools. The file-based nature of both XMLtapes and ARC files makes it straightforward to distribute content over multiple disks and servers.

4 Future Work

Two aspects of the reported work will require future updating of the XMLtape/ARC approach:

- First, a problem related to the indexing of XMLtapes must be resolved. Many XML parsers do not support byte-level processing. However, correct byte-level location is essential to yield a waterproof solution for the two indexes that are created for XMLtapes, both of which are based on byte-count and byte-offset. This problem currently limits the choice of XML tools that can be used for the indexing process. A fundamental solution to this problem should come from support for the DOM Level 3 API [19] in XML tools, as this API requires support for byte-level location.
- Second, under the umbrella of the International Internet Preservation Consortium (IIPC) [20], a conglomerate of the Internet Archive and national

libraries, the ARC file format is undergoing a revision. Formal requirements for the revised format have been specified, including OAIS compliance, ability to deal with all Internet protocols, support of metadata, and capability to verify data integrity [21]. The authors are involved in this effort, and have provided input, some of which is aimed at making the revised file format even more suitable for the use case of storing local content, in addition to the typical Web crawling use case [22]. At the time of writing, a draft proposal for a WARC file format is available and awaiting further comments. Once a new format is accepted, existing ARC files in aDORé will be converted, and new tools compliant with the new format will be put in place.

5 Conclusions

This paper has described a storage approach for Digital Objects and its constituent datastreams that has been pioneered in the context of the aDORé repository effort by the LANL Research Library. The approach combines two interconnected file-based storage mechanisms that are made accessible in a protocol-based manner:

- XMLtapes concatenate XML-based representations of multiple Digital Objects, and are made accessible through the OAI-PMH.
- ARC files concatenate constituent datastreams of Digital Objects, and are made accessible through an OpenURL Resolver.
- The interconnection between both is provided by conveying the identifiers of ARC files associated with an XMLtape as administrative information in the XMLtape, and by including OpenURL references to constituent datastreams of a Digital Object in the XML-based representation of that Digital Object.

The approach is appealing for several reasons:

- The file-based approach is inherently simple, and dramatically reduces the dependency on other components as it exists with database-oriented storage.
- The disconnection of the indexes required for access from the file-based storage approach allows retaining the files over time, while the indexes can be created using other techniques as technologies evolve.
- The protocol-based nature of the access further increases the flexibility in light of evolving technologies as it introduces a layer of abstraction between the access method and the technology by which actual access is implemented.
- The XMLtape approach is inspired by the ARC file format, but provides several additional attractive features. It provides a native mechanism to store XML-based representations of Digital Objects that are increasingly being used in Digital Library architectures. This yields the ability to use of off-the-shelf XML processing tools for tasks such as validating and parsing. It also provides the flexibility to easily deal with Digital Objects that have multiple constituent datastreams, and to attach a wide variety of metadata to both

those Digital Objects and their datastreams. Of special interest for preservation purposes is the ability to include XML Signatures for constituent datastreams (stored themselves outside of the XMLtape) as metadata within the XML-based representation of a Digital Object stored in the XMLtape.

- Used in this dual file-based storage approach, ARC files keep the appeal they have in the context of the Internet Archive. For aDORe, they are appealing for additional reasons, including the existence of off-the-shelf processing tools, the proven use in a large-scale environment, and the prospect of the format – or a new version thereof – being used in the international context of the International Internet Preservation Consortium that groups the Internet Archive and national libraries worldwide.

As can be understood, the proposed XMLtape/ARC approach is not tied to aDORe's choice of MPEG-21 DIDL as the complex object format to represent Digital Objects. The approach can also be used when Digital Objects are represented using other formats such as METS or IMS/CP. As a matter of fact, at LANL, the XMLtape approach is even used to store the results from OAI-PMH harvesting of Dublin Core records, in which case the record-level administrative information contains the OAI-PMH identifier and datestamp of the Dublin Core record to which it is attached. While currently untested, the proposed approach could also be used as a mechanism to transport large archives encoded as XMLtape/ARC collections from one system to another.

Acknowledgments

The authors would like to thank Jeff Young from OCLC for his willingness to update the OAICat software to accommodate the multiple repository use case. And many thanks to our LANL Research Library colleagues Jeroen Bekaert, Mariella Di Giacomo, and Thorsten Schwander for their input in devising many facets of the aDORe architecture. Finally thanks to Michael L. Nelson for proofreading a draft of this paper.

The reported work is partially funded by an NDIIP grant from the Library of Congress.

References

1. International Organization for Standardization. ISO/IEC 21000-2:2003. Information technology – Multimedia framework (MPEG-21) – Part 2: Digital Item Declaration (1st ed.)(2003)
2. The Library of Congress: The Network Development and MARC Standards Office. Metadata Encoding and Transmission Standard (METS) (2004, November) Retrieved from <http://www.loc.gov/standards/mets/>
3. International Organization for Standardization ISO 14721:2003: Space data and information transfer systems – Open Archival Information System – Reference model (1st ed.) (2003).

4. Burner, M., Kahle, B.: Arc File format (1996, September 15) Retrieved from <http://www.archive.org/web/researcher/ArcFileFormat.php>
5. Van de Sompel, H., Bekaert, J., Liu, X., Balakireva, L., Schwander, T. (accepted submission): aDORe: a Modular, Standards-based Digital Object Repository. *The Computer Journal* (2005). Preprint at <http://arxiv.org/abs/cs.DL/0502028>
6. Bekaert, J., Hochstenbach, P., Van de Sompel, H.: Using MPEG-21 DIDL to represent complex Digital Objects in the Los Alamos National Laboratory Digital Library. *D-Lib Magazine*, 9(11) (2003, November) Retrieved from <http://dx.doi.org/10.1045/november2003-bekaert>
7. Bekaert, J., Van de Walle R., Van de Sompel, H. (submitted): Representing Digital Objects using MPEG-21 Digital Item Declaration. *International Journal on Digital Libraries* (2005)
8. IMS Global Learning Consortium: IMS content packaging XML binding specification version 1.1.3. (2003, June) Retrieved from <http://www.imsglobal.org/content/packaging/>
9. National Information Standards Organization. ANSI/NISO Z39.84-2000: Syntax for the Digital Object Identifier. Bethesda, MD: NISO Press (2000, May)
10. Leach, P., Mealling, M., Salz, R.: A UUID URN Namespace (3th ed.) (IETF Internet-Draft, expired on July 1, 2004) (2004, January).
11. Van de Sompel, H., Hammond, T., Neylon, E., Weibel, S.: The “info” URI scheme for information assets with identifiers in public namespaces (2nd ed.) (2005, January 12) Retrieved from <http://info-uri.info/registry/docs/drafts/draft-vandesompel-info-uri-03.txt>
12. Van de Sompel, H.: XMLtape XML Schema. <http://purl.lanl.gov/STB-RL/schemas/2005-01/tape.xsd>
13. International Organization for Standardization. DIDL XML Schema. <http://purl.lanl.gov/STB-RL/schemas/2004-11/DIDL.xsd>
14. netarchive.dk. <http://www.netarchive.dk>
15. National Information Standards Organization. (in press). ANSI/NISO Z39.88-2004: The OpenURL Framework for Context-Sensitive Services. Bethesda, MD: NISO Press.
16. Lagoze, C., Van de Sompel, H., Nelson, M. L., Warner, S. (Eds.): The Open Archives Initiative protocol for metadata harvesting (2nd ed.) (2002, June) Retrieved from <http://www.openarchives.org/OAI/openarchivesprotocol.htm>
17. Berkeley DB Java Edition. <http://www.sleepycat.com/products/je.shtml>
18. Online Computer Library Center. OAICat (2004, October) Retrieved from <http://www.oclc.org/research/software/oai/cat.htm>
19. DOM Level 3 API. <http://www.w3.org/DOM/DOMTR>
20. International Internet Preservation Consortium. <http://netpreserve.org/about/index.php>
21. Christensen, S., Stack, M.: ARC file Revision 3.0 Proposal. (2004, September) Retrieved from http://archive-access.sourceforge.net/arc_revision_3/index.pdf
22. Liu, X., Van de Sompel, H.: ARC File Format Revision 3.0 : Feedback from the Los Alamos National Laboratory (2004, November) Retrieved from <http://public.lanl.gov/herbertv/papers/arc3-20041101.pdf>

A No-Compromises Architecture for Digital Document Preservation

Thomas A. Phelps and P.B. Watry

University of Liverpool
Liverpool, Great Britain
phelps@ACM.org, P.B.Watry@liverpool.ac.uk

Abstract. The Multivalent Document Model offers a practical, proven, no-compromises architecture for preserving digital documents of potentially any data format. We have implemented from scratch such complex and currently important formats as PDF and HTML, as well as older formats including scanned paper, UNIX manual pages, TeX DVI, and Apple II AppleWorks word processing. The architecture, stable since its definition in 1997, extends easily to additional document formats, defines a cross-format document tree data structure that fully captures semantics and layout, supports full expression of a format's often idiosyncratic concepts and behavior, enables sharing of functionality across formats thus reducing implementation effort, can introduce new functionality such as hyperlinks and annotation to older formats that cannot express them, and provides a single interface (API) across all formats. Multivalent contrasts sharply with emulation and conversion, and advances Lorie's Universal Virtual Computer with high-level architecture and extensive implementation.

1 Introduction

Of the many issues to digital preservation—capture (reading data from old physical media or harvesting web sites), provenance, metadata, data management, long-term storage, availability, disaster prevention, multiple data types (scientific data, video), protecting intellectual property, and others—this paper focuses on the problem of obsolescence of digital document data formats. It is an important problem: “obsolescence of media formats and data formats is the most demanding problem while preservation of bitstreams can be mastered by using well-known techniques” [17].

Within this focus of digital document formats, it is worthwhile to consider what constitutes successful preservation. For documents on paper, preservation of the physical material implies that the content can be viewed, if sometimes under restricted access. Digital works are unlike paper in that the preservation of the material itself, the data files, is trivially accomplished, once the documents have been initially collected, by successive copying.

However, viewing of digital documents is complex. Beyond a few text-based formats such as ASCII text, document formats are severely if not entirely unreadable without decoding by specialized software. Digital documents often include time-based content such as sounds, video, and animations. Digital documents often contain

active elements such as forms, scripts, and plug-ins. In the context of scientific data and software, which parallel documents with embedded programs, Messerschmitt [12] points out that the distinction between the two is “rapidly blurring” and states that “data and software preservation targets are not separate, but should be assumed from the beginning to be largely inseparable”. Marshall and Golovchinsky [9] consider the additional nuanced dimensions of literary hypertexts, “that arise from the works on-screen appearance, its interactive behavior, and the ways a readers interaction with the work is recorded”. Similar arguments could be made for research systems in general and any other document system with idiosyncratic concepts. Preservation of a viewing capability is especially problematic if the software is proprietary, the software runs only obsolete hardware, and the data formats are not public.

And viewing is not enough. More is expected of digital documents than paper. Users expect to copy and paste text, images, videos, and other content types. Institutions, web sites, and individuals all expect to search the contents of documents (this function is so fundamental it is being built into next-generation operating systems). For some users, text-to-speech and automatic Braille generation are essential. Companies and researchers want to perform text mining and automatic language translation. People want to convert documents to the format du jour, such as for handheld devices with small screens. Researchers want to infer the semantic structure of documents, utilizing all the information the document contains, everything from layout coordinates to style sheets (if any) to explicit semantic structure (if any). Users want to add hyperlinks and annotate, even if the document format does not support those concepts. We can expect the future to be increasingly demanding as new applications are invented that rely upon potentially any aspect of document content.

2 Related Work

2.1 Hardware Preservation

Document software, like software in general, often requires specific supporting software and, directly or indirectly (perhaps through the operating system), specific hardware. The problem is hardware breaks down, and new generation hardware may not be compatible with the software. Hardware preservation to preserve the readability of the original digital document by maintaining the original hardware and software indefinitely. Such a hardware museum is destined to ultimately fail as the hardware breaks down with no other like machine to cannibalize for parts, and parts are too specialized to resume manufacture cost effectively.

2.2 Emulation

Required hardware can be emulated in software on current (more powerful) computers, and therefore emulators can reproduce a document’s exact appearance and behavior. It requires quite a bit of work by experts to emulate a computer, especially a modern computer, but there are many applications for such emulators and several companies sell them. When the current computer grows obsolete, a new emulator for it can run the emulator of the previous generation, and so on, creating an ever-growing stack of emulators, which may or may not be sustainable.

In any case, the document content remains trapped within the emulator. Somewhere within the emulator's memory soup are program data structures that represent "the document". But finding the document and extracting it remains at least as difficult as interpreting the document file's original bitstream. We would like to add the document content to a search engine or send the document to others to read without the overhead of the emulation stack, but cannot.

2.3 Conversion/Migration

Conversion, also called migration, takes material in an older format and recodes it into a newer format. This can have some success for simple data; perhaps the many formats of raster images can all be represented on a two-dimensional grid of color values. But digital documents are more complex and in general semantically incompatible from one another, and conversions from one to another almost always lose information for the fundamental reason that some concepts in one document format cannot be expressed in the other.

As a document format evolves every few years with new releases of the corresponding software, the software can usually read the last couple versions of its own format, but documents older than a mere several years may become unreadable. Thus, the conversion process requires constant attention, constant migration. This chain from format to format can lose information at every step, relentlessly degrading quality. While data loss is almost guaranteed for conversions between document formats, it is likely even within upgrades to the same software application.

Today, although emulation and conversion suffer well-known problems, they are often seen as the only ways. The UK National Archives [3] tries to mitigate the damage done by developing a database of file formats, called PRONOM, that "allows for the automatic generation of migration pathways, by identifying every possible conversion route between a source and target format, with information about how each conversion stage will affect the content". Nevertheless, even if the damage at each step is limited, when multiplied by tens or hundreds of years of conversions, and such a time span is after all the point of preservation in the first place, the data loss is substantial and certain.

CAMiLEON [10] addresses the cumulative data loss problem by always converting from the original bytestream. Documents are read into an intermediate format and various output formats can be developed as needed. The architecture was demonstrated on a selection of vector graphics formats. This is promising, but faces additional issues when applied to more complex documents. Even among vector graphics formats, semantic gaps required elements to be downgraded, and we can expect more of this (even complete data loss in places) with complex document models, which may or may not be an acceptable compromise. It does not address document behavior, such as a JavaScript manipulation of an HTML DOM. The intermediate format seems to be a union set of concepts from all supported formats, and as a practical matter would likely become exceedingly large and unwieldy as the hundreds or thousands of document formats were adopted, many with idiosyncratic concepts and most all with innumerable small but potentially important variations on common structures.

2.4 Universal Format

Some systems convert all sources into a single universal format, which it uses for all further operations. XML seems like an attractive candidate as it captures semantics and structure, is extensible, and is easy to parse. Virtual Paper [2] and UpLib [6] (neither of which claim to be a basis for digital preservation) solve the multiple format problem by capturing image and text representations of all documents, one “projection” to view and the other to search.

The most famous examples of universal formats are PostScript and PDF, which boast the unique advantage that they can already capture any document that can be printed (which is effectively all formats with static content) and increasingly more applications are generating PDF directly and at a higher semantic level than what is sent through a printer driver. In a single format, PDF supports high fidelity viewing as well as text-based operations such as searching, and the PDF file format can bundle the original document bitstream for future editing or more demanding preservation. Adobe promotes PDF for archiving [1], pointing out that PDF is a publicly available (but not open) standard and supports XML metadata records, among other features. A PDF metastandard for archiving called PDF/A [5] identifies “the set of PDF components that may be used and restrictions on the form of their use,” such as disallowing the patented LZW compression filter and requiring that all fonts be embedded.

Somehow the universal format is eternal, and perhaps it becomes so important that society ensures this. Nevertheless, the approach has its limitations. It is simply not practical to completely capture all aspects of all document formats in a union set format. The format would be unwieldy, hostile to full implementation, and would have to be updated constantly as new formats are introduced. So-called universal formats must of practical necessity select certain features and leave others behind, and thus there is a conversion step and corresponding data loss to their use.

2.5 Universal Virtual Computer

Raymond Lorie proposes writing data interpreters “that can extract the data from the bit stream and return it to the caller in an understandable way, so that it may be transferred to a new system” [8]. Programs are written against a Universal Virtual Computer (UVC) so that in the future, all that is needed is an implementation of the UVC on the computer of the day to run the interpreters and thus read the data.

The UVC is extremely cautious about what is certain about the future, and requires little more than the equivalent of a simple microprocessor and memory. Considering this approach from the practical point of view of software engineers charged with building a system that embraces hundreds of document formats of sometimes great complexity, this is not enough.

In practice, software engineers need an architecture outlining the large-scale organization of the software to be built and detailing the interactions among the many components. For preservation of digital documents, this architecture should embrace such domain-specific concepts as “document”, “metadata”, “text”, “behavior”, and “structure”. In practice, software engineers require a high level language, such as Java, and libraries of pre-built functions (all of which can be compiled to the UVC). A level above the UVC must interface with hardware, such as displays, keyboards, and mice.

Lorie's UVC is a solid start, and now it is time for higher-level architecture and implementation.

3 The Multivalent Architecture's Benefits for Preservation

We now examine the Multivalent¹ Document Model to see how its architectural qualities support digital document preservation. The purpose here is not a presentation of the architecture per se (for that see [15] or a concise presentation in [16]), but an elucidation of how important aspects of the digital preservation problem are solved by certain aspects of the architecture, sometimes uniquely so.

The architecture is powerful and versatile, as can be appreciated from the following description of an earlier application of the architecture to a browser. The Multivalent Browser natively displays many document formats (PDF, HTML, scanned paper, UNIX manual pages, TeX DVI, others) and supports *in situ* annotation (highlights, notes, executable copy editor markup, Notemarks) across all formats. Annotations can attach to any point of a document (letter or image), can apply to documents that are read-only (such as the New York Times home page), anchor with Robust Locations so they can reattach correctly even if the source document has been extensively edited, and exploit Robust Hyperlinks to find a document if it moves elsewhere on the Internet.

The architecture has been implemented. The system totals over 100,000 lines and over 4 million characters of source code. The document parsers mentioned above and the browser are freely available online [13]. In the past year, the implementation has been deployed for preservation, first in the San Diego Supercomputer Center's Persistent Archive Testbed project [18].

The architecture is proven over time. Since 1997, as the API has evolved and implementation has advanced, the Multivalent architecture has remained stable. (This predates Lorie's UVM, but it took Lorie to indirectly point out its suitability for digital preservation.)

The architecture has many interlocking concepts, and it can be instructive to first briefly consider the totality. New document formats are supported by *media adaptors*, which are code components that translate concrete document formats into runtime data structures. The primary data structure is the *document tree*, which represents the entire content of a document (as a scroll, or a page at a time), including everything from the text and images, to scripts, to the semantic structure (hierarchy and attributes), to the physical layout. Active (programmable) elements of a specific document or a document genre, such as hyperlinks or outline opening and collapsing, are implemented by *behaviors*, which are program code with complete access to the document contents. The particular behaviors that apply to a document or genre are listed in XML-format *hubs*.

The remainder of this section fleshes out those architectural concepts that address specific aspects of the digital document format preservation problem.

¹ Multivalent was born as a thesis project at UC Berkeley, and the creator has since moved to the University of Liverpool.

3.1 Media Adaptors

New document formats are supported by media adaptors, which are code components that translate concrete document data formats into runtime data structures, primarily the document tree. Currently implemented media adaptors include PDF, HTML, scanned paper of two OCR formats, UNIX manual pages, TeX DVI, ASCII, and Apple II AppleWorks word processing, among others.

Media adaptors encapsulate format-specific parsing knowledge, and are obligated to eliminate any need for further reference to the concrete bitstream. This entails correcting the format wherever needed (coercing HTML to comply to a DTD), and presenting the rest of the system with uniform word units, which may require splitting lines in ASCII or pasting together word fragments in PDF or TeX DVI.

The core system has no media adaptors officially “built in”, although a few popular ones happen to be bundled with the usual distribution. The core system merely associates a MIME type or file type suffix to a hunk of code. The system provides all of the modern access to and control of documents in general. Because there is no distinction between obsolete document formats and those in current use, *obsolete document formats are as vigorous as those in current use*.

Media adaptors directly read original concrete document data formats. This avoids a problem of conversion in which bugs or approximations in one stage cumulatively degrade quality. Bugs, while always undesirable, are more benign in media adaptors, because once they are fixed, all subsequent viewings and other uses are automatically corrected. In the same way, partial implementations of formats, such as ones being painstakingly step-by-step reverse engineered, are *incrementally improvable*. Any progress can be disseminated and exploited immediately, without delaying until perfection is reached, and improvements can be distributed as they are achieved.

The capability to read original document formats is bundled with the system and therefore always *available on demand*. With conversion, perhaps the apparatus employed converted all the known documents in bulk, perhaps by a third party, and now the user encountering a new instance must revive that. With hardware preservation, the museum of hardware and software has a geographic location and even if it is on the network, it may not be amenable to opening its fragile, irreplaceable exhibits to random poking from millions of Web surfers. With a universal format, the fact that the new format may be easy to parse does us no good unless the document has been preprocessed.

Media adaptors serve as *operational definitions* of document formats. When media adaptors are the result of reverse engineering, their operational definitions also serve as the de facto specifications. When media adaptors are based on separate specification definitions, they remain essential as they illuminate the dark corners of real world (ab)use that lie outside the light of the specification. For example, HTML as found on the Web is almost never correct, and it often requires considerable correction, not to the W3C’s HTML specification, but to the operation definition given by Microsoft Internet Explorer. In Multivalent, these operational definitions are part of a live system, so they are always being tuned and kept up to date.

As compared to conversion, media adaptors move the preservation problem from constant massaging of billions of documents to maintaining one media adaptor per format. Preserving individual documents is reduced to *just copying the bits*.

3.2 Document Tree

The primary data structure is the document tree, which represents the entire content of a document (as a single scroll, or a page at a time). The hierarchical structure of a document is directly reflected in the hierarchy of parent-child nodes in the tree, and all nodes may contain attributes. For documents such as SGML, XML and SVG, the tree directly reflects the parse tree of the document. HTML is similarly represented, but after correction to a DTD. Internal nodes of the tree are structural, and leaves hold content (text, bitmapped images). All nodes have layout bounding boxes (coordinates and dimensions), with internal nodes containing the union rectangle of their children. Ordinarily structure and layout coincide, but sometimes a special branch at the root of the tree is required to accommodate divergences such as floating images and multiple columns. Metadata is available as attributes on the root of the tree. Media adaptors can introduce new nodes types when needed, unlike HTML's Document Object Model. Remarkably, all document formats seen so far fit comfortably into a common document tree, from the fixed-format scanned paper and PDF at one end of the continuum, to the flowed HTML and UNIX manual pages at the other.

The document trees of sophisticated document formats are decorated with spans. Hypertext links and font styling are both span types. Spans provide leaf-to-leaf (and within leaf) control over appearance: font family, size, style; foreground and background color; underlining; line width; and more. Spans also control interaction, reporting keys pressed and mouse activity within the span. A handful of spans are reused by media adaptors for many different document formats.

Note that the document tree is not an intermediate format or a universal format. Unlike an intermediate format, the document tree is used directly for document appearance and behavior, preserving full document expression. (The document tree employs concepts common across formats where possible and can be used for conversion.) Whereas universal formats can bloat as the union set of incorporated formats, the document tree is tailored to an individual format at a time, free of overhead.

In support of preservation, the document tree opens *access to all document content*. Conversion, far from opening everything, typically eliminates unusual content types. Emulation hides content in an impenetrable box. In fact, in one way the document tree is superior to the original software editors/viewers for a format, because that software probably did not give access to other applications, at least not to such an extreme comprehensiveness of text, images, structure, styling, and layout.

The document tree *unifies the representation of all document formats* and lifts them to a common set of modern document abstractions. For example, operating system- and application-specific character encodings, including the various ways of dealing with large international character sets, are all normalized to Unicode. Tools and services target the abstractions and automatically work across all formats. A document analysis application has access to content, style, and layout, regardless if source was scanned paper or TeX DVI. That search engine in the previous section that adopted TeX DVI could also collect the hyperlinks on DVI to add to its crawl.

3.3 Behaviors

Active (programmatic) elements of a specific document or a document genre, are implemented by behaviors. A span on the document tree above is an example of a type of behavior, a media adaptor is a behavior, and outline opening and collapsing is implemented with a combination of behaviors.

Behaviors are arbitrary program code with complete access to the document contents, the network, and the disk (subject to security restrictions, but no architectural limitations). Behaviors can be arbitrarily large. Behaviors can arbitrarily edit the document tree. The sole restriction on a behavior is that it adhere to a certain interface for communication with the system and other behaviors.

For preservation, behaviors *fully embrace the active and idiosyncratic* aspects of a document format. There is no limitation to, say, what JavaScript can access of an inherently limited scripting level. For example, PDF defines a set of annotation types, such as ink and stamp, that none of the other implemented document formats do, and with a set of properties that are unlike any other document format. Each annotation type (some but not all of which are presently implemented) becomes a behavior type, and each annotation instance a behavior instance. If a literary hypertext needs a new hyperlink type with special features, it could introduce it as a behavior.

3.4 Hubs

The particular behaviors that apply to a document or genre are listed in XML-format hubs. Hubs use XML attributes to customize general behaviors (passing a URL to a hyperlink, for example) in the same way programming languages employ parameters for functions. Hubs use XML hierarchy to nest more complex data associated with a behavior; for example, when the user authors a note-type annotation, the note behavior is saved under the top-level, and the content of the note, its fonts and colors, and even annotations on that annotation, are nested hierarchically within the note.

Some behaviors apply only to one document (e.g., annotations), some to genres (e.g., the manual page outliner control), and others to multiple formats (e.g., a pop up menu that can send word under cursor to a definition service).

Behaviors developed for one format (or for no particular format at all) can be associated to others formats via hubs and thus bring *new ideas to old formats*. This is not strictly required by preservation (fully expressing the format is sufficient) but is necessary to bring older formats out of the ghetto and into parity with newer ones, and after all, someday today's formats will be considered ancient too. Users want to annotate all of their documents, whether PDF which has annotation types, or ASCII or WordStar (or HTML!), which do not, and hubs associate function out-of-band and therefore are free of limitations of expressiveness in these formats.

4 Practicalities

4.1 Realization

If the Multivalent Document Model defines an architecture well suited to the needs of digital document preservation, it is immediately apparent that it will require a

large-scale implementation effort to fully embrace the 100s or 1000s of file formats. The large number and the fact that they are generally semantically incompatible from one another inherently force individual attention and thus demand considerable effort. But the implementation effort is no more than UVM or CAMiLEON.

Fortunately, the work is highly parallelizable to independent teams implementing media adaptors for different formats. The Multivalent architecture defines the necessary technical points of coordination, but otherwise imposes no bureaucratic overhead, and individual teams can choose formats of local importance or interest. Since all the media adaptors are part of the same architecture, common components can be shared, as is presently done for paragraph formatting of multiple fonts and for hyperlinks.

Our task is considerably lighter than the sum total of all the original document software. Preservation emphasizes appearance and behavior, not the considerable editing component of a system. Devising the document format requires considerable intellectual effort, which we merely read from a specification. We use modern technology and tools, whereas some original systems were written in assembly language to run in 48K bytes of memory.

4.2 Preservation of Preservation

Any preservation strategy will take maintenance to adapt it to future technologies, and our system is preserved in the same way as Lorie's UVC. In the UVC, implementations target a simple core and the maintenance problem is reduced to porting the core. Multivalent is implemented entirely in Java, and our UVC is the Java virtual machine (JVM). The JVM is directly analogous to the UVC as both are virtual machines at more or less the level of assembly language. Java's VM is somewhat more complicated, but the primary consideration is not absolute simplicity but rather a complete, rigorous definition (Java's is given in [7]) coupled with a "reasonable" level of implementation achievability. Perhaps it would not be politically auspicious for IBM to point to a Sun technology as the bedrock of its preservation strategy, but the fact that numerous companies have implemented compatible JVMs proves its viability.

It would be absurd to build a large system in the assembly language of virtual machines. Moving from satisfaction of the key self-preservation requirement to the software engineering considerations of building a large system, we must choose a high-level language. (The use of Java's VM does not imply the use of the Java language itself, as many programming languages can compile to the VM, just as many programs can compile to the different microprocessors. Different groups could choose different languages and effectively cooperate.)

5 Future

The Multivalent architecture is well suited for preservation, but was originally designed for use in a browser. It could benefit from the insights of experts in preservation to ensure that the overall approach fully embraces all essential details before a large-scale implementation effort is launched.

An important subproject will be the collection of document format specifications. These are important for their intrinsic status as the intentional definitions, and considering how time consuming reverse engineering is, it is important for software engineers to have easy access to these specifications. Wheatly [19] catalogs numerous books, web sites (for example, [20]) and projects that collect many types of file formats. Companion subprojects should collect implementations, which are the operational definitions of formats, and sample documents for developers.

Table 1. Comparison of selected systems used for digital preservation

	PDF	UVM	CAMiLEON	Multivalent
Defined	1990	2001	2001 (?)	1997 / applied to preservation in 2004
Demonstrations	everything that can be printed	JPEG and GIF bitmapped images (claimed PDF in fact based on a conversion to HTML)	interconversion among SVG, Draw, WMF vector graphics	PDF, HTML, scanned paper, TeX DVI, UNIX manual pages, Apple II AppleWorks
Method	printer driver captures print stream, or app directly generates. Format externally supported.	read original bitstream by document interpreter	read original bitstream into intermediate representation, convert to another file format	read original bitstream and build runtime data structures
Strengths	captures static aspects of all formats, well developed, well defined	potential to fully express document appearance and behavior	as compared to other conversion, only one level of quality degradation	fully expresses document appearance and behavior
Use by Applications	use Acrobat or third-party library	undefined	App du jour picks up output file format	Live runtime linking (also amenable to conversions)
Document Architecture	emphasis on graphical appearance; structure expressible but not common	undefined	intermediate representation (either unwieldy union set of all formats, or leave out idiosyncratic)	fully developed (media adaptors, document tree with structure and layout, behaviors, spans, hubs, ...; fixed and flowed layouts)
Software Engineering	File format well documented, Acrobat API, many third-party libraries	low-level assembly language UVM (in practice use Java)	(unknown)	well-exercised system API, high-level language (Java)
Maintenance	Upgrade money to Adobe	port UVM to new machines	develop new output formats, "software longevity principles"	port Java VM to new machines
Drawbacks	everything must look like PDF (fixed layout, paginated); lose idiosyncrasies and behavior)	Implementation immature: No document architecture. UVM too low level for development	Conversion's semantic gap between formats downgrades or loses data. Loses behavior.	Intimate linking by apps, or develop own apps. (No compromises for document quality.)

Undoubtedly the present technology will need to be generalized and refined, and already one area in particular area is evident. Documents are often found in wrappers

of various kinds, sometimes for compression (such as .zip files) and sometimes in virtual filesystems (such as the Structured Storage used by Microsoft Office applications). A layer underneath document parsers would need parse these structures and provide access to the documents inside.

This paper has concentrated on digital documents, but there are many other media types (some of which are embedded in documents) in need of preservation, such as scientific data, audio, music scores, video, multimedia such as Macromedia Flash, and DVD menu programs, to name a few. It is unclear whether these all can be accommodated under a common architecture. But Multivalent has already demonstrated its applicability to a variety of documents with text, images, vector graphics, and programmatic manipulation of a document tree. Even if it is limited to this class of a couple hundred formats, billions of document instances make a claim for some social value.

6 Conclusion

Compared to existing approaches to digital document preservation, the Multivalent Document Model offers a step forward. Compared to conversion, the original document remains perfectly preserved. Compared to emulation, the content of the document is easily available. Compared to Lorie's UVM, Multivalent defines the high level architecture necessary for software engineers, and Multivalent's implementation of number of complex and obsolete document formats prove the architecture's power and no-compromises suitability for preservation. Multivalent is a proven plan in the present for the future of preserving the past.

References

1. Adobe Systems, Inc. *PDF as a Standard for Archiving*, Adobe white paper. <http://www.adobe.com/products/acrobat/pdfs/pdfarchiving.pdf>
2. Birrell, A. and McJones, P. Virtual Paper Web site, (1995–1997). <http://www.research.compaq.com/SRC/virtualpaper/>
3. Brown, A. Preserving the Digital Heritage: Building a Digital Archive for UK Government Records, In *Proceedings of Online Information*. (2003) <http://www.nationalarchives.gov.uk/preservation/digitalarchive/>
4. IBM. Digital Asset Preservation Tool Web site, <http://www.alphaworks.ibm.com/tech/uvc?Open&ca=daw-flHts-120204>
5. International Standards Organization. ISO/CD 19005-1, *Document management—Electronic document file format for long-term preservation—Part 1: Use of PDF (PDF/A)*, November 31, (2003). [http://www.aiim.org/documents/standards/ISO_19005-1_\(E\).doc](http://www.aiim.org/documents/standards/ISO_19005-1_(E).doc)
6. Janssen, W.C. and Popat, K. UpLib: a universal personal digital library system, In *Proceedings of the ACM symposium on Document Engineering*, (2003) 234–242
7. Lindholm, T. and Yellin, F. *The Java Virtual Machine Specification, 2nd edition*, Addison-Wesley Longman Publishing Co., Inc. (1999)
8. Lorie, R. Long term preservation of digital information, In *Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries* (2001) 346–352
9. Marshall, C.C. and Golovchinsky, G. Saving Private Hypertext: Requirements and pragmatic dimensions for preservation. In *Proceedings of ACM Hypertext* (2004) 130–138

10. Mellor, P, Wheatley, P, Sergeant, D. "Migration on Request : A Practical Technique for Digital Preservation" ECDL (2002)
11. Meehan, J., Taft, E., Chernicoff, S., Rose, C., Karr, R. *PDF Reference, fifth edition*, (2004)
12. Messerschmitt, D.G. Opportunities for Research Libraries in the NSF Cyberinfrastructure Program, *ARL Bimonthly Report* 229 (2003). <http://www.arl.org/newsltr/229/cyber.html>
13. Multivalent Web site. <http://multivalent.sourceforge.net>
14. The National Archives. PRONOM Web site. <http://www.nationalarchives.gov.uk/pronom/>
15. Phelps, T.A. *Multivalent Documents: Anytime, Anywhere, Any Type, Every Way User-Improvable Digital Documents and Systems*, Ph.D. Dissertation, University of California, Berkeley (1998)
16. Phelps, T.A. and Wilensky, R. The Multivalent Browser: a platform for new ideas, In *Proceedings of Document Engineering*, (2001) 58–67
17. Rodig, P., Borghoff, U.M, Scheffczyk, J., and Schmitz, L. Preservation of digital publications: An OAI extension and implementation, In *Proceedings of the ACM Symposium on Document Engineering*, (2003) 131–139.
18. San Diego Supercomputer Center. Persistent Archive Testbed (PAT). <http://www.sdsc.edu/PAT/>
19. Wheatley, P. Survey and Assessment of Sources of Information on File Formats and Software Documentation (2003) http://www.jisc.ac.uk/uploaded_documents/FileFormatsreport.pdf
20. Wotzits Format? Web site. <http://www.wotsit.org/>

A Study into the Effect of Digitisation Projects on the Management and Stability of Historic Photograph Collections

Veronica Davis-Perkins¹, Richard Butterworth², Paul Curzon³, and Bob Fields¹

¹ Interaction Design Centre, School of Computer Science,
Middlesex University, London. UK. N14 4YZ
{v.davis-perkins, b.fields}@mdx.ac.uk

² Senate House Library, University of London, Malet Street, London. UK. WC1E 7HU
rbutterworth@shl.lon.ac.uk

³ Department of Computer Science, Queen Mary, University of London,
Mile End, London. UK. E1 4NS
pc@dcs.qmul.ac.uk

Abstract. The results of an ongoing interview study with custodians of historic photograph collections are reported. In particular the success or otherwise of recent digitisation projects is addressed, as well as the extent to which these projects have affected the long term management of the collections. We examine the effects of digitisation on the primary sources, their digitised surrogates and the relationship between the two in terms of selection, authenticity and representation. In most cases we have observed that the emphasis placed by the funding bodies on ‘accessibility’ of tangible numbers of resources is detrimental to these three other issues. However, we report in detail on one case study of a local history library where its digitisation work is embedded in core library activity and seen as successful and positive. We conclude by suggesting that their deliberate eschewing of short term project funding is a determining factor in their success.

1 Introduction

New technology has allowed for the unlocking of ‘memory collections’ for public access and has also enabled users to integrate information from many different sources instantaneously. In the UK the past decade has seen a growth in the availability of lottery funding for local and family history collections within the public library and museum world. Often referred to as ‘heritage’ or ‘memory collections’, [6, 13] they have caught the public imagination and custodians have seen a dramatic growth in demand for access to such collections.

In the past such resources were the province of professional specialist researchers, genealogists, and experienced family and local historians. Today, however, anyone with access to a computer can research their own interests and, with the materials now available online, build a unique picture of their own histories and sense of social place. Such online ‘memory collections’ have been argued to have a strong potential to improve citizens’ sense of self and their society’s historical context.

However, there are implications to giving the public access to such a wealth of electronic records of social and cultural memory. Our current research investigating digitisation procedures shows that whilst being beneficial to the general public in terms of access to many previously hidden collections, digitisation is proving to be problematic for both the custodial community and the original resources, especially in the case of photographs. In particular the primary focus of recent digitisation projects has been ‘access’ (See, for example [11]). Our work suggests that putting ‘access’ as the overriding goal is certainly not neutral, and can be detrimental when it comes to other issues in collection management and sustainability. The reasons for these problems are numerous: lack of funding and resources, bad communication between custodians and technologists, bad project management and planning.

However we look in detail at a digitisation programme undertaken by a local history library which has balanced access with other collection management issues, and has succeeded in developing a well used online collection, where other well funded projects have failed. We conclude by discussing whether general lessons can be learnt from this success.

2 Methodology

Over the past two years we have interviewed 21 custodians of historical photograph collections largely selected from the Library and Information Commission’s Directory of Digitisation Projects in UK Local Authority Libraries and Archives [12]. The interviews have taken the form of semi-formal note-taking and pro-forma interview or tape recorded interviews and work practice observations. We have also collected quantitative data from questionnaires which will be analysed and reported in later work – it is the qualitative interview data that we predominantly report here. Our initial analyses of the quantitative interview data support the broad conclusions presented in this paper.

The digitisation projects we have looked at range from those in large, public-funded institutions to those in small local history libraries and specialist subject libraries. The projects often involve the digitisation of mixed media materials from local collections, to which members of the community are sometimes invited to participate by adding their own oral, written, or photographic information and memories to the website.

Our broad aim is to assess the effect that digitisation is having on custodial practice by defining ‘values’ held in primary sources and their digitised surrogates. We can then show how the process of digitisation alters those values, and ultimately suggest well evidenced modifications to existing digitisation processes so that the digitised surrogates better represent the values of the primary sources. However analysis of the interview data has also raised several other issues concerning how digitisation projects are progressing and what effect they are having. It is these issues that we report in this work. The main emphasis of this study is historic photograph collections in the UK, but we would expect the findings reported here to have some generality to other collection domains in other countries.

3 The Impact of Digitisation Projects

The custodians we have interviewed all work on projects which make the assumption that digital access to ‘memory collections’ gives the public accurate information – that what they see and what they read are true representations of past events, places, and lives.

The projects we have looked at typically run a fixed term of eighteen months (ten months is the shortest, two years the longest) and are externally funded. The funding bodies typically work on the assumption that all they are paying for is the digitisation. It is assumed that cataloguing, housing and preservation are either all solved problems which have been dealt with as part of the core library activities before the digitisation project begins, or are small issues that can be dealt with as secondary issues as part of the project. Our interviews with custodians suggest that this is not the case, particularly for the small, specialist collections. It is particularly worrying that the funding criteria does seem to be establishing a ‘Matthew Principle’ (‘He that hath, to him shall be given, but he that hath not, from him shall be taken even that which he hath’) among library collections. Large libraries have whole departments dedicated to securing funding to digitise their already well managed collections. Whereas small libraries are struggling to put together evidence that their collections are well managed enough to warrant digitisation funding, therefore their collections lose public profile, and funding becomes even more difficult to obtain. Only one third of our interviewees used any established standards to guide the project development.

The relationship between the primary source and its digitised surrogate was the main focus of our research. In this paper we draw out the ways in which commonly applied digitisation procedures affect the relationship between the two, broadly categorised into selection, authenticity and representation issues. These issues may be pragmatic: due to failings in project management or lack of resources, or more systemic.

3.1 Selection

The sheer numbers and variation of photographic materials within archives means that very few, even well-funded organisations, have the resources to put entire collections online. Therefore, decisions have to be made about which materials to digitise. These decisions may be systematic or haphazard, but more importantly are typically not made clear to the sites’ audience. The viewers therefore, are unlikely to know that they are not seeing the full picture. Nor are they being made aware of the context from which the images originate.

We asked custodians about the motivations for digitisation in their institutions. This has revealed that the reasons for selection are not always democratic or even custodially sound but may be driven by:

- subject (60% of projects); typically local scenes and events or those images that best reflect the content of the collection.
- use and familiarity (30% of projects); custodians select those items that previous use patterns show to be popular or useful to users.

- vulnerability (15% of projects); custodians select the images needing conservation treatment.
- Other selection reasons given were:
- current commercial bias; what the custodians (or more typically their managers) believe will raise the collection's profile the most, or make them the most money,
 - what is presently considered aesthetically pleasing,
 - ease of digitisation, we observed one project where a photograph collection had been previously rather haphazardly partially catalogued by volunteers. At the beginning of the digitisation project the decision was made to digitise only those photographs that had already been catalogued, as this would give the easiest 'hit' of a large number of catalogued, digitised images,
 - copyright issues; online copyright is still considered to be rather a black art by many custodians, therefore for safety's sake images may be selected for digitisation where the copyright issues are considered uncontroversial.

In approximately one third of cases a short-term project was undertaken by outside teams who were not familiar with the collections, and little consideration was given to the context of the archive from which objects were selected. This almost random selection can remove materials from their context, create fragmentation of collections as an entity, and remove clues and information about the provenance of materials. These findings emphasise the importance of inherent custodial knowledge about a collection, especially when selecting material suitable for digitisation.

Criteria for selection are often made on the perceived needs of the targeted viewer. Hence there is a danger of producing a 'turn-of-the-century view' shaped, as one archivist interviewee put it, by 'today's trends for nostalgia' rather than by online resources that will have sustainability over time.

These issues may not be of particular relevance to the general public as viewer. After all, to find access to details about one's own past is exciting enough, and feedback to the websites in question suggests that the public is more than happy to take what is currently available at face value. The question here, we argue, is one of authenticity and representation of historical material being accessed by the public. Further, the integrity with which selections are being made should be such that the research they undertake gives them as accurate a picture as possible.

Conway [4] suggested that selection is choice, but one could equally argue that selection in itself is an editorial mechanism – a management system used all the time in traditional libraries: what is on open access, what we can see and touch, and what is not available for public access. Users selecting images to view online are only ever selecting a subset of the images that the digitisation team have selected for them. The users' choice therefore only becomes meaningful if the motivations behind selection are trustworthy, explicit, enduring over time, and non-political. This could mean that selection may not be easy. It may be a long process, requiring the input of different professional opinions and it may mean that what is selected does not necessarily meet current populist views. Particularly in the case of 'memory collections', online information sometimes only gives the general public access to an edited view of history, rather than a more balanced view they would have if they were aware of the context from which the information originated.

Although making selection decisions based on use and familiarity may seem to be sensible and ‘democratic’, (because the project is delivering online what the custodians already know to be popular) it could be equally argued that doing so reduces user choice, by limiting the user’s search results to repeated use of the same images.

3.2 Authenticity

The authenticity of a photograph is part of what makes it a primary source – that its provenance can be proven to be genuine adds to its value. Validating the authenticity of historical photographs, however, has always been problematic. Photographs have by tradition been particularly difficult to authenticate because of the reproductive nature of photography, its processes, its history and the sheer mass of production. (See [8, 9, 11] etc.) In the case of digital surrogates, with all the possibilities for easy manipulation it is even more difficult to assess the authenticity of the original unless its cataloguing and identification procedures were carried out at the time of creation. This is rarely the case, particularly with historical collections. Even a photographer as conscientious about recording his work as Ansel Adams, is known to have either mislaid, forgotten, or even deliberately thrown records of dates of photographs he created [1]. How can the authenticity of a digital surrogate be measured if, as Klijn and de Lusenet [10] suggest we can only rely on ‘contextual clues such as the authority of the organisation presenting [the images].’?

Interviewees in our study confirmed that the lack of genuine choice for custodians combined with managerial and government pressure to digitise means that access is in danger of taking precedence over authenticity. Furthermore, according to conservation specialists we have interviewed, the individual resources themselves are often distorted both through manipulation and accidental loss, during the transformation from original to digital, not only because of human intervention but also because technology cannot always reproduce colour accurately, especially when dealing with the complexities of black and white photographic processes. The issue of authenticity is also particularly relevant to digitising photographs because of the amount of physical information carried in the original, in addition to their aesthetic properties. This transfer of information is a difficult task on many levels including the human components of visual analysis, emotion, and technical experience. Just how relevant these losses or distortions are depends very much on the intention of use. But in all cases, digital reproduction raises questions of authenticity.

3.3 Representation

Any representation of reality is always going to be a filter. Primary sources give a filtered view of reality, and digitisation is another filter on top of that. In all likelihood it will never be possible to define an archival or digitisation process which will not be found wanting by future historians. Without the support of the primary resource, historical objects taken out of context, changed during the process of digitisation, inevitably lead to the misrepresentation, or at least a filtering, of historical evidence.

Much has been written about photographic reproduction and its value as a representational medium, (See, [2] etc.) and it is not within the remit of this paper to carry those particular arguments further. However, the ease with which digital technology

can change, substitute, and otherwise manipulate photographic material is highly relevant to examining the impact of digital reproductions being made available to the general public as evidence for building their own histories.

Our interviews confirm that the use of digital editing is widespread amongst technology practitioners but there is not always a record kept of the changes made. Although photographers and photographic printers have always ‘doctored’ photographs to some degree, it is the speed with which it can now be done, and that manipulations may not be immediately obvious to the viewer, that is cause for concern. In many digital projects, we have found that there have often been no guidelines given to digital technicians about the extent to which manipulation is acceptable, or that records should be kept. Further many scanners perform several colour manipulation processes without the user realising. It may be possible to switch these processes off, but it is only the very well informed user who will realise that these processes are occurring in order to know to switch them off or document them. Typically the choice for manipulations to be made is at the discretion of the technician, and it is unlikely that the technician is best qualified to make these decisions. This raises ethical questions about what level of information is represented to the public as historical evidence and their subsequent awareness of any changes or other manipulations made to the surrogate that differ from the original source.

Thorough and informative cataloguing is also key to improving the representation of a historical artefact. It is not only the case that good cataloguing helps users find items, but the better the cataloguing, the richer the contextualising information that surrounds an item and the better able a user is to appreciate it in its historical context. Funders assume that cataloguing is already done to an adequate standard before digitisation funding is allocated. We have seen that this is rarely the case; cataloguing is often done from scratch or improved on as part of a digitisation project. However, as cataloguing is not seen as the main outcome of a project it is often done in a brief, perfunctory way. We saw one project where individual photographs were catalogued by a single term, many photographs were catalogued by the same single term. This made finding both primary photographs and their digitised surrogates problematic for both users and custodians.

3.4 Discussion

There is a simplistic assumption that the digital surrogate is of equal value to that of primary objects, in that users can do everything with a digitised surrogate that they can with the primary source. On that basis the public can rely on the information they access. However, the evidence of this research suggests there are inherent problems with keeping the same level of value in a digital surrogate, in that the processes of reproduction always changes the materiality of the object that has been reproduced.

Digitisation projects badly need what Bellinger [3] calls ‘procedural accountability.’ It is more important that current digitisers document and make the decisions about what they digitise and how they do it explicit, than whether the decisions they make turn out to be right or wrong. However, it should be stated that technology developers and experts entrusted with building digital libraries should be driven by a greater understanding of the issues involved in the selection of historical materials and the complex information they carry.

The speed of change and the rapid growth in access to surrogate information, overlooks the issues of authenticity and loss of information. Often it is not clear what constitutes a good website: it is not always to be found in the superior quality and ingenuity of the site but whether or not the information carried within it is authentic and sustainable. However, funding bodies wanting to see evidence that their money has been well spent will resort to simple objective measures like how many records have been put online.

If projects are not planned and well-structured from the start, we have found that they can create a legacy of problems for custodians left with responsibility for their maintenance, in two cases, as we have discovered, these problems can be so drastic that they can lead to a project's complete abandonment.

Clearly, lack of adequate funding is an obvious candidate for blame when it comes to projects delivering digitisations where the selection decisions, authenticity and representation are questionable. However there is no clear correlation in our work between project success and level of funding. Indeed one of the most successful 'projects' we have seen has no external funding at all.

4 A Local History Library Case Study

So far we have painted a gloomy picture of digitisation projects. The majority of our interviewees have expressed considerable concern that digitisation is not living up to its expectations, and is adding yet more stresses into the management of historic collections. We therefore move on to look in detail at a particular approach to digitisation undertaken by a local history library that is integrating mixed visual collections into a general online bibliographic catalogue for online access. This approach seems to be successful, and although not perfect, a lot of the concerns raised by our other interviewees appear to be being at least addressed, if not solved.

Like many local borough library services, our case study local history library is popular with the local community as testified by the steady growth in readers over the past ten years. The collections now take up an entire floor of a substantial purpose built library services block and it takes three full-time librarians to staff. True to the nature of most local history libraries, the collections consist of mixed media, which are difficult to access and store because of their diversity (sizes and materiality), and physical vulnerability.

4.1 Programme Planning

The library staff are well versed with the difficulties associated with access to photograph collections which had been the first motivation for wanting to digitise them. Having studied other similar digitisation projects and discussed concerns with colleagues from within the library and archive community, the library staff made a plan for digitisation. The objectives of the plan were not technically ambitious, and, as objectives, were similar to those of most other digitisation projects we had looked at. They included carrying out a survey of the photographic holdings, making selection decisions based on the survey, cataloguing the collection using the existing online catalogue system, digitising images and attaching the digitisations to their catalogue entries.

The distinguishing feature of this approach to digitisation is that no attempt has been made to seek external funding to support the project: the plans are to be fulfilled as part of the normal day to day running of the library. In this particular case study, the problems that normally arise from employing outside project staff are eliminated by using existing library and IT staff working as a team. Apart from the cataloguer, who works full time on the photograph collection, each member of the team has their own area of responsibility to the digitisation programme which they fit in with their everyday duties. The cataloguer is the only full time member of staff employed on the programme. Having worked for the library service for over 20 years, he has developed a deep knowledge of the collections, and is experienced in dealing with reader's inquiries. The cataloguer's background plays an invaluable role in identifying material and structuring the descriptive catalogue which is the strength behind this project. The input manager (who is also the technical leader responsible for producing library's website), and the scanning technician both integrate their digitisation tasks into their daily schedules.

4.2 The Digitisation Programme

An initial survey of the photograph collections was made in order to identify vulnerable or damaged photographs and negatives. This first trawl of the collections enabled the cataloguer to make a preliminary selection for a test trial of procedures. Delicate articles (such as glass plate negatives) were assessed by colleagues from the archive service and sent out to a recommended museum photographer for photographic copying and digitisation.

It was decided that the cataloguer should try to create as many fields as possible (within reason), that a user may need when searching the database. Once catalogued the photographs are passed on to the scanning technician in small groups at a time. When scanning has been completed, the photographs are returned to the cataloguer who box files and shelves them according to their reference numbers. To help promote the collections, the cataloguer also writes a weekly column for the local newspaper in which a picture is discussed and readers are invited to feed back information about it to the library.

For the first scanning trials they experimented with JPEG format at 600dpi's but they found this took up too much memory for their system. They now scan most photographs at 200dpi's using PDF. If a user makes a special request, for a higher resolution or a different format then they are happy to respond but will make a small charge for doing so.

The input manager is at the end of the digital chain in that he receives electronic files of the images from the scanning technician, and descriptive data from the cataloguer. In interview the input manager stressed the importance of pre-planning and discussion between the different members of the team, as well as making an effort to keep up with other people's experiences in the domain. He also emphasised the importance of keeping things as simple as possible and the need to work steadily to achieve what is possible within specific circumstances. The philosophy behind the programme is that it is better to do a little well, than a lot and mess it up. 'You can't have all your cakes at once. You have enough cakes on there... so you're up and running... then you can look at other things and add bits and pieces to it. We know

that what we can offer now is selective and limited but we are building all the time and we will finally reach the stage where we practically have everything.'

Other useful information, such as an Ordinance Survey grid reference is added to some of the illustrations so that the user can see where a specific place or building is located. The website also has a feedback page for adding or correcting information but most of the feedback about the images comes from the cataloguer's weekly column in the local newspaper.

The intention is to continue updating the website until the entire photograph collection has been catalogued and digitised, along with the odd painting, drawing, or map as needed by researchers. They also intend to improve the site, adding further facilities as work pressure allows, for example by adding and giving information to cover more towns and villages within the area and perhaps adding an audio facility.

4.3 The Outcomes of the Programme

Over two years the programme has so far digitised two thousand (of ten thousand) images with full and thorough cataloguing and has also been very successful in generating user feedback and comment about the images. Only one serious problem was identified by the staff during these interviews; they believed there was a weakness in the quality of their scanned images. It is felt the images cannot be scanned to best advantage because of the size of files required to do so, and the time they can presently allow for scanning procedures. However given that the photographs are carefully catalogued and housed, it is not problematic to find individual primary sources and digitise them at high resolutions should users request them.

The aim of the programme is to eventually digitise all the photograph collections, and in that sense selection decisions are not important. However given that the programme will continue over several years selection decisions are being made as to which items get priority. These decisions were initially made based on the original collection survey, but can also be modified based on user requests.

The determination to place both user needs and the cataloguing of the photograph collection firmly at the forefront of the programme's objectives, is the reason that in terms of financial budget and time, the catalogue was given priority. As discussed in the previous section in many projects, we have found that cataloguing is an area which is often hastily pulled together, (if at all) where the most important objective is seen as the number of images that can be put online in the shortest time. Detailed, descriptive cataloguing is a time consuming task but is also essential if a project is to have any stability or sustainability.

Just as an analogue library is constantly growing and changing, so too is this digital library as more ideas are implemented and new images added. This is a dynamic library that is being steadily built within the current capabilities of funds and staff, for the purpose of meeting the needs of a specific community.

5 Conclusions

It is an easy assumption to make that the problems with digitisation projects we identified at the start of this paper are caused by lack of funding; none of the custodians

we interviewed complained of having too many resources. However this assumption is too simplistic; the evidence of our interviews strongly suggests that it is the artificiality of deadlines and outcomes imposed by short term projects that causes the problems more than lack of resources.

It is interesting to note that when we have discussed these issues with individuals from the funding bodies, or who have managed the large flagship digitisation projects in the UK, they have candidly asserted that most, if not all, of the projects funded so far have been exploratory techno-centric projects primarily aimed at discovering what opportunities new technology can offer, rather than looking at how new technology impacts on collection management and end users. ‘Sustainability’ of digitisation projects has been seen to be problematic for several years, but the solutions suggested by the funding bodies (advertising, affiliation, subscription, etc) are untested. Our evidence suggests that sustainability is a problem because of the monolithic product centred focus of project work. A much better model for sustainable, well managed digital resources is shown by the case study described in the second half of this paper: digitisation is considered as a *process* to be undertaken as part of the normal core activity of the library, rather than as a *product* which at some arbitrary date is ‘finished’. It is the pressure to finish a digitisation project that causes corners to be cut with cataloguing and other long term collection management issues, rather than lack of funding.

An evolutionary approach provides other benefits. A product centred approach is typically based around a bureaucratic ‘waterfall’ design approach, where requirements are established at the beginning of the project (hopefully, but not always, in liaison with end users), then a product is designed and built for the main duration of the project, and then tested against these requirements towards the end of the project. It is well documented [5] that this approach does not sit at all well with user centred design (eg. [7]). Even if the requirements are established with end users in mind, then the resulting product does not get tested with users until near the end of the project, by which time it is usually too late and too expensive to make any serious changes in the light of user feedback. Even this unsatisfactory approach is not reflected in practice: not one of the projects we looked had conducted *any* evaluation of their deliverables.

A more evolutionary approach means that small amounts of content can be digitised, put online and subjected to public scrutiny (recall how the cataloguer wrote a weekly column in a local newspaper, inviting online comment from local users). Both the content and the delivery system can then be incrementally, therefore easily and cheaply, improved based on user feedback.

We need to be careful making very general claims about the case study being a model for all digitisation projects as it may be simply a happy concatenation of surface events that causes their approach to be a success, rather than there being some more profound difference. The case study is unusual amongst those we have examined for several reasons, some of which may seem purely random: the placement of staff in a particular place, at a particular time of technology’s development; a photograph collection that is in numbers, manageable, and in format, suitable for digitisation; a Borough Executive that encourages innovation and welcomes flexibility in its

employees, and a Library, Archives and Museums Service that communicates across disciplines. A failure in any one of these elements would probably mean that their approach to digitisation would become problematic, whether they were using an evolutionary or product centred approach. We would argue, however that the case study's approach is likely to cope much better with serious problems, such as knowledgeable and experienced staff leaving, whereas a failure in staff retention would almost certainly cause a short term project to fail completely.

The other important factor is communication: from the beginning there had been a constructive dialogue between the cataloguer and the input manager out of which emerged the initial digitisation plan and the will to drive it forward. A failure point we have seen in other projects has been the librarians' inability to express what they actually need from the technologists in terms that the technologists can fully understand. Having a librarian (the cataloguer) and a technologist (the input manager) who spoke each others' language undoubtedly helped considerably.

Digitisation of historic resources is costly and complex. Requirements and expectations can legitimately differ greatly between institutions. However the technology underlying digitisation is now becoming stable and cheap enough that libraries can begin to consider digitisation as being a core activity alongside their more traditional activities such as cataloguing, indexing, conservation, etc. rather than being a separable, externally funded 'bolt-on' activity. Our research suggests that such a move would be beneficial in several ways.

To successfully merge public access and commercial expediency with promoting heritage collections, there needs to be less techno-centricity (less worrying about what current technology can do, because that changes rapidly), and more worrying about what stakeholders actually need, and how technology can be made to meet those needs.

References

1. Adams, A., *Examples: The Making of forty Photographs*, Bulfinch Press, 1984.
2. Barthes, R., *Camera Lucida: Reflections on Photography*, Vintage, 1993.
3. Bellinger, M., *Information Representation: The case for surrogates*. Parallel Lives: digital and analogue options for access and preservation. Joint conference of the National Preservation Office and Kings College London. Digital Consultancy Service. See <http://www.kcl.ac.uk/humanities/cch/kdcs/content/conf2003summary.htm>. 2003.
4. Conway, P. *The Relevance of Preservation in a Digital World*, North Eastern Document Conservation Centre, Andover, USA. <http://www.nedcc.org/plam3/tleaf55.htm>. 1999.
5. Dix, A., Finlay, J., Abowd, G., Beale., R. *Human computer interaction*, Pearson, 2004.
6. Dempsey, L. *Scientific, industrial, and cultural heritage: a shared approach: a research framework for digital libraries, museums and archives*. Ariadne issue 22, <http://www.ariadne.ac.uk/issue22/dempsey/intro.html>
7. Faulkner, X. *Usability engineering*, Macmillan, 2000.
8. Frizot, Michel, *A New History of Photography*, Könemann. 1998.
9. Jeffrey, I., *Photography: A Concise History*, Thames and Hudson, 1996.

10. Klijn E. and de Lusenet, Y., In the Picture: Preservation and digitization of European photographic Collections, European Commission on Preservation and Access. <http://www.knaw.nl/ecpa/PUBL/pdf/885.pdf>. 2000.
11. Martin, E. and Ride, P. (eds) Collecting and Preserving Old Photographs. Collins. 1998.
12. The Museums, Libraries and Archives Council, Listening to the Past, Speaking to the Future, Report of the Archives Task Force, 2004.
See http://www.mla.gov.uk/documents/atf_report.pdf
13. Parry, D. Directory of completed, current and planned projects in libraries and archives. In Virtually New: Creating the Digital Collection. Final report to the Library and Information Commission. pp56 – 105.
See <http://www.ukoln.ac.uk/services/lic/digitisation/dig.doc>. 1998.
14. UNESCO Memory of the world. Available from UNESCO website: <http://www.unesco.org/>

Strategies for Reprocessing Aggregated Metadata

Muriel Foulonneau and Timothy W. Cole

¹ Grainger Library, University of Illinois at Urbana-Champaign
1301 W. Springfield Avenue
Urbana, IL 61801
+1 - 217 - 244 -7809
{mfoulonn, t-cole3}@uiuc.edu

Abstract. The OAI protocol facilitates the aggregation of large numbers of heterogeneous metadata records. In order to make harvested records useable in the context of an OAI service provider, the records typically must be filtered, analyzed and transformed. The CIC metadata portal harvests 450,000 records from 18 repositories at 9 U.S. Midwestern universities. The process implemented for transforming metadata records for this project supports multiple workflows and end-user interfaces. The design of the metadata transformation process required trade-offs between aggregation homogeneity and utility for purpose and pragmatic constraints such as feasibility, human resources, and processing time.

1 Aggregating Metadata Describing Scholarly Resources

In recent years, large aggregations of metadata describing heterogeneous resources have been created using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). OAI service providers who build applications on top of such aggregations must amalgamate large amounts of metadata harvested in a range of formats. By reprocessing harvested metadata, service providers can adapt metadata for their specific use and present those metadata to end users in an integrated fashion.

The process of adapting metadata for another application than originally envisioned when the metadata records were created, i.e., repurposing metadata, requires analyzing the metadata harvested, identifying processes to apply to the metadata, and then building the reprocessing system to select, transform and organize the metadata. The present paper discusses issues related to metadata analysis and the implementation of a metadata reprocessing system. It suggests a range of strategies for metadata reprocessing and adaptation and identifies issues needing further study.

1.1 The CIC Portal: An Aggregation of 450,000 Metadata Records

The CIC metadata portal is a major metadata aggregation encompassing digital resources from 9 Midwestern universities in the U.S., mostly holdings of academic research libraries. It provides access to 450,000 descriptive metadata records from 152 defined collections. The CIC metadata portal has three main interfaces. A primary search and retrieval interface provides classic digital library access points for

scholars: author, title, subject, type, and a number of filtering and grouping functionalities based on dates and collections,. A clickable geographic map allows users to browse by spatial coverage attributes of the resources indexed. Finally a second search and retrieval interface is provided that takes advantage of both collection-level and item-level descriptions in concert[3]. The interfaces were designed to improve information retrieval in aggregated collections, improve usability of heterogeneous information, and demonstrate the wealth of U.S. Midwestern digital library resources. To enable these three interfaces, metadata is reprocessed and then ingested by two distinct systems: the DLXS software developed by the University of Michigan on top of the OpenText XPat search engine; and a Microsoft SQL database. We describe below the implementation of our initial metadata reprocessing system, detailing workflow from harvesting to data publishing.

1.2 Metadata Reprocessing: Challenges and Objectives

"A metadata record is created in the objective of a specific use." [2] The challenge of repurposing metadata records is to reuse records created to fit one context in a different context with different constraints and objectives. Any attempt to reuse descriptive metadata runs the risk of misusing (misunderstanding) those records. The problem is exacerbated since even assuming that metadata records are generally well-adapted to the original context for which they were created, this original context typically remains partially or totally unknown to service providers. Moreover, OAI-PMH is designed to facilitate the harvesting of the same metadata records by multiple service providers, each likely to have their own unique purpose and context. However, since metadata records are expensive and resource consuming to generate, their reusability has great potential for benefit and is at the core of a number of current initiatives, notably the National Science Digital Library and the Digital Library Federation working group on best practices for OAI and shareable metadata.

These considerations dictate a thorough analysis of harvested metadata evaluated in terms of the service provider's context and typically applying different measures of metadata quality and utility [9] than were applied when the metadata records were originally created. Metadata records adequate in original local context may not be adequate in an aggregated context. In reprocessing harvested metadata, the service provider's challenge is to implement strategies to transform harvested metadata in a way that enhances usefulness, while avoiding misuse or misunderstanding.

1.3 Technical Implementation of Metadata Reprocessing

For the CIC portal, harvest of the OAI metadata provider repositories (18 in all) is customized according to a number of configurable parameters, including the strictness of XML validation, the specific sets to harvest, and the metadata format to harvest.

Once harvested, records are first processed through a program that selects relevant records (for the purposes of the aggregation). Selected records are then sent through an XSL pipeline (chain of XSLT files) which implements a series of transformations.

Each pipeline is customized by repository and composed of a number of XSLT stylesheets (five to eight) that are named in a configuration file. Only two XSLT stylesheets per repository are repository-specific. Repository specific stylesheets mostly implement a subset of element-specific normalization and augmentation functions. To facilitate normalization, the stylesheets import a generic dictionary of XSLT templates and a number of data dictionaries encoded in XML: e.g., ISO639 language codes, Dublin Core Metadata Initiative and CIC type vocabularies, a date data dictionary, ISO3166 and ISO3166-2US geospatial codes, and a subset of Internet MIME Type (IMT) format string values.

The selected, normalized, and enriched metadata records are stored in a distinct location separate from where the as-harvested metadata records are maintained. Periodically a procedure is run to upload the enriched metadata records into the Microsoft SQL database. Another program applies an additional stylesheet and concatenates transformed metadata records as required for use in the DLXS-based service mentioned above. The concatenated files are then transferred to another server where a shell script routine rebuilds DLXS indexes. Specifics of the metadata record filtering, normalization, and enrichment tasks implemented for the CIC portal are described below in sections 3 & 4.

2 Metadata Analysis

Each new collection to be added to the aggregation is analyzed to define specific reprocessing needed. We also identify new mappings or transformation keys (e.g., to convert coded values to human-readable strings) for the data dictionaries used in reprocessing. We look both at a statistical characterization of the entire population of records and at the range of value strings encountered considered by metadata field.

2.1 Statistical Analysis of the Records Population

Analysis of the whole body of records harvested from a given provider is difficult to do manually due to the large numbers of records typically provided. A statistical analysis is therefore essential to obtain key indicators on the record population.

Earlier work here at the University of Illinois led to the development and refinement of a methodology for analyzing the use of DC metadata elements. [4, 7, 10] We have extended this methodology and applied it successfully to simple and qualified DC records of the CIC aggregation, and more recently to alternative formats offered by CIC metadata providers (e.g., ETDMS and MODS). Metadata records were analyzed either on a repository by repository basis, or where necessary on a collection by collection basis.

The metadata records are entered into a database and the summary information listed in Table 1 about the population of metadata records is generated. This analysis of the metadata population allows a service provider to identify reprocessing needs and anticipate impact new collection might have on the overall metadata aggregation.

Table 1. Statistics generated for metadata populations

<p>General metadata population</p> <ul style="list-style-type: none"> <input type="checkbox"/> The size of collection or set (i.e., number of records)
<p>Structure of the records</p> <ul style="list-style-type: none"> <input type="checkbox"/> The average number of metadata elements per record <input type="checkbox"/> The list and frequency of attributes used to specify encoding schemes or any other information <input type="checkbox"/> The use of the <i>About</i> section of the record to specify rights statement, provenance, or any other relevant information about the metadata
<p>Elements used</p> <ul style="list-style-type: none"> <input type="checkbox"/> The list and frequency of metadata elements of the specified metadata format that are actually used <input type="checkbox"/> Whether there is present in each record exactly one URL understood to link (directly or indirectly) to the individual resource described <input type="checkbox"/> Whether there is present in each record at least one of the fields displayed in results lists (i.e., title, subject, or description) <input type="checkbox"/> Whether all the fields used as browse or limit categories for portal interfaces are present in all records (eg. <i>Type</i>)
<p>Value length</p> <ul style="list-style-type: none"> <input type="checkbox"/> The number and percentage of empty metadata elements

2.2 Analyzing Metadata Values

All distinct text values are then extracted from the database for each metadata field present in the record set being analyzed. Also extracted are counts of how often each distinct text value occurs for the field considered. Distinct text values for each field are ordered by frequency of occurrence. This provides an indication of set consistency and the use of controlled vocabularies. Generally the characteristics of metadata elements are quickly identified using this approach, even in large metadata populations. The frequency of recurring values is also easy to identify since each distinct value is displayed with its number of occurrences in the collection.

This analysis of metadata values on a field-by-field basis also makes it easier to identify the location in the records of the concepts used in the CIC metadata portal, either as access points or for display purpose or as categories. For instance, the CIC metadata portal makes special use of resource type, format, language, URL, and rights information when present. Analysis facilitates identification of parameter values needed for XSL templates (e.g., what character to use for splitting concatenated terms) and any needed modifications or extensions of the data dictionaries used during reprocessing. When new unrecognized values appear (e.g., “technical reports” as a *Type*), they can be added. However, in several cases, the meaning may not apply across all collections (e.g., The *Type* “other” means *software* in one collection, *text* in another). The data dictionaries are built to accommodate such variability.

For several fields, it is interesting to identify redundant values found in multiple records – e.g., in *Identifier*, in *Description*, and in *Title* – as the presence of such redundancy may indicate the presence of duplicate or overlapping metadata records. For other fields, redundancy is a good sign; it can suggest consistency and imply that the records in the set being analyzed will be easy to normalize. In one collection, duplicate records, having different URLs pointing to the same resource could be identified because their description and title fields were similar. It should be possible in the future to automatically identify records where the only difference is the page number of an online book. Page by page granularity does not make sense for the CIC metadata portal. Such records, while not duplicates in original local context, are essentially duplicative in the context of our CIC service provider implementation.

Improperly used (in the context of the service provider portal) and imprecise concepts should be normalized or renamed (i.e., put in a different field). When processing simple DC records, it may be necessary during processing to separate IMT format values from extent information. This is a qualified DC distinction. Spatial and temporal coverage data, which appear in the same element in simple DC records, are also considered for further processing. Information contained in records is assessed according to its function in the CIC portal. For instance, special effort is made to identify a single main URL per resource. We also verify the presence of elements needed for display in search results lists. Finally encoding consistency for any fields commonly retrieved across the whole metadata collection is assessed. This is an issue for *Author/Creator* values which may be encoded differently in different collections.

2.3 Analyzing Language Used in Metadata Fields

Metadata records harvested for the CIC aggregation include both "guide" metadata, expressed in natural language and intended for human consumption, and "control" metadata, intended for use in context of a database or other computer system application. [1] For the purposes of a search and discovery service, the fields used as access points should be clearly identified and controlled vocabularies used should be clearly labeled in all metadata records. This also implies that the terms by which the records are queried are the same as the ones used in the metadata records or at least retrievable through additional language processing functions commonly used in search engines (see for example [6]). Consider the following examples of Dublin Core *Description* field content: "First ed. Cf. BM.", "D_North_American_1983_HARN", "Added t.-p., engr.", "Co. C". These strings are not really useful for direct retrieval since most users do not tend to query using abbreviations. Neither do they tend to query by codes. In order to search on such content, codes and abbreviations must be expanded to more human-readable forms. While this can be difficult for abbreviations or codes that are idiosyncratic to a particular local context, the service provider can expand recognized standard abbreviations and codes contained in harvested metadata.

OAI service providers also must display retrieved metadata records for end-users. Results listings of metadata records allow an end user to select resources of interest after gauging their likely usefulness for meeting his or her information need. All metadata fields are not equally efficient for this purpose. It depends on the information they contain and the form of language used in the values (e.g., natural language versus codes). Each metadata field that will be displayed for the end user

should be human interpretable. For example, “wln” is a code, created for machines, it is not human interpretable if the user does not know the ISO639 codes for languages. Some values may be both human and machine amenable. The value, “Text.Correspondence.Letter” in a Dublin Core *Type* field is easily understood by an end-user, although its stilted form suggests it is also intended to be machine parseable. However, even in such cases, it may be desirable to explicitly adapt controlled encodings to more human-readable forms. Thus, while the value, “197-“ is likely to be understood by most users, the value “created between 1970 and 1979” decreases the risk of end-user misunderstanding.

While the transformation of metadata field value strings to more human-readable form is generally desirable, a service provider displaying a value that was not originally created by the data provider risks betraying the original record, either intellectually or legally. This issue can be of major concern. In the museum community, a metadata record that describes an original object may be the result of an in-depth scientific analysis. Altering such a record before presenting it to end-users might risk providing less precise, duplicate, or less accurate information. This can have negative consequences for both service and content provider. The natural inclination of the service provider should be to maintain the original record values, unless he can safely and with a high degree of certainty decode an encoded value. Metadata normalization that focuses on adding machine readable values, expanding standard codes and abbreviations, and adding explicit labels for such information as type, language, URL, collection and rights, as we have done, is a fairly safe first step.

3 Metadata Reprocessing for the CIC Metadata Portal

After analyzing harvested metadata, we take the following steps to reprocess records for use in the CIC metadata aggregation.

3.1 Records Selection

The primary criteria in the collection development policy for the CIC metadata portal¹ are that all resources described by metadata in the aggregation should originate from a CIC institution and that the metadata records should be descriptive. Some repositories harvested include metadata describing both CIC and non-CIC resources. For those repositories the CIC portal collection development policy is implemented in the selection process using scripts to identify relevant metadata records. De-duplication of records, however, is not currently implemented at initial record filtering.

3.2 Metadata Cleaning

This process consists of deleting erroneous characters at the beginning and end of strings, most commonly extra occurrences of characters used to delineate concatenated values (typically, certain data providers concatenate multiple values in a single metadata element, e.g., <description>16 History; 17 Geography;</description>, </description>

¹ CIC collection development policy <<http://cicharvest.grainger.uiuc.edu/collection.asp>>

which can result in an unnecessary semi-colon at the end of the value), removing empty metadata (in the first analysis of the raw data collected that was done at the beginning of the project, 17.5% of the records contained at least one empty metadata element such as <description />), removing meaningless values (<date>--</date>), and splitting data when a metadata element contains multiple values (such as in the *History* and *Geography* example mentioned above).

3.3 Metadata Normalization

This step consists in disambiguating concepts and metadata semantics across the collections. It maps the elements used by data providers to a number of normalized fields of a specific metadata format used internally in the CIC portal (and derived from qualified DC). The *Format* field is renamed as *Extent* in the following case: “<format>163 pages</format>”. Values also are reprocessed at this stage for machine interpretability. This is the case for the resource type which is mapped to a controlled vocabulary used in a drop-down list on the search interface and for a filtering option in the search results. The format and the language are also normalized. One URL suitable for linking to each resource also is identified at this stage if possible.

3.4 Metadata Augmentation

Metadata augmentation consists in adding information to the records from external sources. While normalizing *Type* information, the values are translated through the DCMI-Type standard in addition to the local terminology used for the CIC portal. For several collections, a default *Type* is applied. A collection name is also added to the record to help provide additional context for the end user for when the record might be displayed in a results list. A provenance element is created in order to trace the record source.

3.5 Customizing Records for Use in Portal Interfaces

Even after filtering, cleaning, normalization, and enrichment, records may need to be modified further for use in a specific interface. The records to be ingested by the DLXS application are transformed from our internal, qualified DC-based metadata format to the Bibclass record format used by DLXS. Similar elements also are concatenated so that when the record is displayed *Subject* fields, for instance, are not displayed on multiple lines. For our SQL-based interface, searchable metadata elements are concatenated into an additional field to facilitate use of built-in Microsoft SQL Server full-text search functions and features.

3.6 Performance Issues

Ingestion of new collections has been streamlined. As an example, once initial data analysis was performed for a new collection ingested in February 2005, the actual adaptation of reprocessing filters took less than one hour since there was no specific new processing to write (not always the case). However, any new metadata format to be ingested by the system can require significant new adaptation of the XSLT chain.

The execution of metadata reprocessing should not take too long, otherwise data would not be updated on the portal in a timely manner and the service credibility would be consequently jeopardized. Selection, clean up, normalization and augmentation can currently be performed on the full aggregation within 30 hours using parallel batch programs processing at a speed of about 2 records per second for each process. Processing collection by collection entails additional ongoing maintenance as noted by D. Hillmann and N. Dushay of Cornell University. [5] Metadata provider practices can change over time, meaning that the original, precise analysis of a certain data provider practice does not remain valid forever. This may happen with a new version of a turnkey system which might considerably change the nature of records exported when upgrading to the new version. In a specific repository harvested to build the CIC metadata portal, collection to record associations were originally recognized automatically through processing the URL of each resource as recorded in each item-level metadata record. A number of rules had been defined to identify a collection code in resource URLs. One day, the CIC portal appeared to have lost several of its collections, this altered considerably the value of some of the services offered. The data provider had changed the form of its URLs. Collections to item associations were no longer being recognized for this repository.

4 Specific Metadata Properties Featured in CIC Portal Interfaces

The overall metadata repository is composed of 445210 records (as of February 2005). The metadata reprocessing described above facilitates automatic recognition of specific concepts (properties) present in harvested metadata records, notably the resource URL, associated collection, format, type, and language. Normalization insures encoding of properties when present in a standard manner to facilitate use in portal interfaces. These properties are used in portal interfaces either as search access points, for filtering results, or for display of search results. Table 2 shows

Table 2. Presence of information in the original records

Property	Field potentially containing the property	# of records having field before processing	% of records in the repository
Type	Type field	344816 (297756)	77%
Format	Format field	319157 (42501)	72%
Language	Language field	268994	60%
Collection	Relation field	167990	38%
Resource URL	Identifier field	430848	97%

Pre-processing count of metadata records including at least one occurrence of a field potentially containing a property of interest. (For brevity DC field names are shown -- records in Qualified Dublin Core were included using appropriate field names) Presence of a field does not guarantee property of interest is present and recognizable. An *Identifier* field may contain something other than a resource URL. The value of a

Type field may not correspond to the type vocabulary used in portal interfaces. For *Type* and *Format* we include in parenthesis a count of how many as-harvested records contained a string value from controlled vocabularies used for these two concepts in CIC portal interfaces.

4.1 Acceptable Threshold of Normalization for Implementing a Search Interface

Not all properties used for search access or result filtering in portal interfaces apply to full range of resource descriptions included in the metadata aggregation. Language does not apply to images and URLs are not available for analog-only resources. The share of the records in the aggregation for which a concept can be identified and consistently included as a property in the augmented record will often be less than 100%. This may or may not be significant according to the nature of the property, how the property is exploited in the interface, and why it is not present in all records. For instance, 23% of the as-harvested records do not contain a *Type* field. Since *Type* is used as a way to filter results this means that 23% of the records in the aggregation are excluded when an end user chooses to limit search by *Type*. This is a serious problem. On the other hand, if a user limits his or her search by *Language*, we can reasonably assume he or she is not interested in still images. Excluding records that have no language property because they describe pictures is acceptable.

4.2 Applying Default Values

While record-by-record reprocessing facilitates the recognition of properties used as search access points, for filtering results, and/or for display of results, this technique alone may not insure presence of an essential property in a sufficient percentage of records in the aggregation. You cannot normalize the *Type* property value of a record if no *Type* field is present. Often explicit information is left out of item-level metadata records because in the local context for which the records were created this information was understood implicitly. In order to insure the presence of a property in all or almost all records, default property values are sometimes applied to (i.e., added to) all records in a given collection if the appropriate property value can be inferred with confidence from collection-level information.

The identification of the collection a record belongs to is typically not based on metadata values. A *Relation* field only appears in 38% of the as-harvested records. Even assuming this field is generally being used to express collection association, collection information would be present in at most 38% of the records in the aggregation. This percentage is clearly insufficient. Collection associations are determined primarily by harvest repository and/or OAI set membership.

Assignment of default values by collections is also used to identify restricted access resources. This method may be extended in the future, through the use of collection-level descriptions and property inheritability in order to complete the information contained in the item-level metadata record (information completeness) and to increase confidence in information added (information accuracy). Table 3 shows final post-processing counts of records containing *recognized* property values for the concepts listed in Table 2.

Table 3. Presence of information in records after processing

Property	# of records having property after processing	% of records in the repository
Type	441788	99%
Format	295803	66%
Language	268989	60%
Collection	445209	100%
Resource URL	320005	72%

4.3 Trade-Off Between Accuracy and Share of Aggregation

Not all automated interpretations of concepts and meanings are precise and exact. In some cases, the nature of the resource (digital / analog) is not absolutely certain. After processing, a slight difference (3,322 records) exists between the number of resources without a resource URL in the augmented records and the number of resources to which a *Type* “analog resource” is applied. Even though those records do not contain a resource URL, they are still displayed when filtering “digital resources only.” The probability of inaccuracy can be increased by the application of default values. For example a service provider might apply a default *Type* of “analog resource” to all members of a specific collection (where *Type* was not given in metadata records provided) based on initial inspection of a sample of the collection. If in fact the collection contains a few online resources mixed in with mostly analog resources these resources will be mislabeled as analog only.

The content of the language property of the original record is accurately recognized in almost 100% of cases -- only 5 records contain unrecognized language values (much less than 1%). Unrecognized values either come from values impossible to identify such as “other” or from failure in the data analysis to correctly define the condition for splitting elements (a slash in the following case: "English/Japanese"). The character used as a separator to split metadata elements is identified when analyzing the records in a new collection. It is possible that when the collection was analyzed, the case did not occur. Records being dynamic (updated from uncontrolled sources), new phenomena will appear from time to time.

5 The Future: Targeting Transformations for User Benefits

Metadata reprocessing can be a resource-intensive (i.e., expensive) process for a service provider. There also are potential hazards to the fidelity and integrity of the harvested metadata. The processing performed for the CIC metadata portal may improve the level of completeness of information, but it represents a risk of altering several features of the metadata, notably by making it less accurate. Further work is needed to measure and quantify the magnitude of this risk, especially when the metadata processing includes not only direct interpretation and normalization of string values, but also adds default properties to records based on association with a collection or OAI set or repository. The same metadata reprocessing workflow will be applied in the future to more complex data, such as dates. Complementary

methodologies might also be used better adapt human-readable and machine readable data to the different functions of the portal.

Further work is also needed to quantify the impact of metadata quality on service to end-users. This should include, a comparison of user queries found in transaction logs to the content of specific metadata fields and an assessment of the impact of uncertain metadata created during service provider metadata reprocessing on recall and precision of information retrieval. User expectations and tolerance to inaccuracy of information displayed or retrieved (or not retrieved) may be different depending on whether the service is provided by libraries or in context of Web search engines [8].

Guidelines and best practices² for metadata creation and transformation by data providers contribute to improve the efficiency and accuracy of normalized information. However, service providers usually cannot impose common rules to all data providers and data providers cannot apply a single rule that is valid for all service providers. Ultimately a better understanding is needed of what metadata cleanup, normalization, and enrichment can reasonably and safely be done by service providers (i.e., harvesting agents) versus what processing really should be done by metadata providers prior to making records available for harvesting. While metadata records will always be authored with an immediate and specific local implementation in mind, in an environment that increasingly encourages sharing, reuse, and repurposing of digital metadata and content, authoring “shareable” metadata will benefit development of more robust and full-featured Web services.

Acknowledgements

This work was supported by a grant from the Committee of Institutional Cooperation’s Center for Library Initiatives. We acknowledge the libraries of the following participating CIC member institutions for providing metadata and collection descriptions: University of Chicago, University of Illinois at Chicago, University of Illinois at Urbana-Champaign, Indiana University, University of Iowa, University of Michigan, Michigan State University, Northwestern University, Pennsylvania State University, and the University of Wisconsin-Madison.

References

1. Bretherton, F.P., Singley P.T.: Metadata: a User’s View. In: Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management. IEEE Computer Society, Washington, DC (1994) 166-174.
2. Coyle, K.: Data with a purpose. Talk at the California Library Association Meeting, November 2004. <http://www.kcoyle.net/meta_purpose.html>
3. Foulonneau, M., Cole, T.W., Habing, T.G., Shreeves, S.L.: Using collection descriptions to enhance an aggregation of harvested item-level metadata. In: Proceedings of the Fifth ACM / IEEE-CS Joint Conference on Digital Libraries, Denver, CO (2005, in press).

² E.g., DLF/NSDL best practices for OAI and shareable metadata <<http://oai-best.comm.nsdsl.org/cgi-bin/wiki.pl>> and CIC-OAI project recommendations for Dublin Core metadata providers <<http://cic harvest.grainger.uiuc.edu/dcguidelines.asp>>

4. Halbert M., Kaczmarek J., Hagedorn K.: Findings from the Mellon Metadata Harvesting Initiative. In Koch, T., Sølvsberg, I.T. (eds.), *Research and Advanced Technology for Digital Libraries 7th European Conference, ECDL 2003*, Trondheim, Norway. Proceedings, Lecture Notes in Computer Science vol. 2769, Springer-Verlag GmbH, (2004) 58-69.
5. Hillmann, D.I., Dushay, N., Phipps, J.: Improving Metadata Quality: Augmentation and Recombination. In: *DC-2004: Metadata Across Languages and Cultures*, Shanghai, China. <http://purl.oclc.org/metadataresearch/dcconf2004/papers/Paper_21.pdf>
6. Ross, S., Donnelly, M., Dobрева, M., Abbott, D., McHugh, A., Rusbridge, A.: Natural Language processing. In: *Core technologies for the cultural and scientific heritage sector. Digicult technology watch report 3* (2005) 67-103. <<http://www.digicult.info/downloads/TWR3-lowres.pdf>>
7. Shreeves, S.L., Kaczmarek, J., Cole, T.W.: Harvesting cultural heritage metadata using the OAI protocol. *Library Hi-Tech*. 21 (2003) 159-169.
8. Shreeves, S.L., Kirkham, C.M.: Experiences of educators using a portal of aggregated metadata. *Journal of Digital Information* 5 (2004) Article No. 290, 2004-09-09. <<http://jodi.ecs.soton.ac.uk/Articles/v05/i03/Shreeves/>>
9. Shreeves, S.L., Knutson, E.M., Stvilia, B., Palmer, C.L., Twidale, M.B., Cole, T.W.: Is quality metadata shareable metadata? The implications of local metadata practice on federated collections. In Thompson, H.A. (ed.): *Proceedings of the Twelfth National Conference of the Association of College and Research Libraries*, Minneapolis, MN. Association of College and Research Libraries, Chicago, IL (2005, in press).
10. Stvilia, B., Gasser, L., Twidale, M., Shreeves, S.L., Cole, T.W.: Metadata quality for federated collections. In *Proceedings of ICIQ04 - 9th International Conference on Information Quality*. Cambridge, MA (2004) 111-125.

A Hybrid Declarative/Procedural Metadata Mapping Language Based on Python

Greg Janée¹ and James Frew²

¹Alexandria Digital Library Project
Institute for Computational Earth System Science
University of California, Santa Barbara
Santa Barbara, CA 93106-3060
gjane@alexandria.ucsb.edu

²Donald Bren School of Environmental Science and Management
University of California, Santa Barbara
Santa Barbara, CA 93106-5131
frew@bren.ucsb.edu

Abstract. The Alexandria Digital Library (ADL) project has been working on automating the processes of building ADL collections and gathering the collection statistics on which ADL's discovery system is based. As part of this effort, we have created a language and supporting programmatic framework for expressing mappings from XML metadata schemas to the required ADL metadata views. This language, based on the Python scripting language, is largely declarative in nature, corresponding to the fact that mappings can be largely—though not entirely—specified by crosswalk-type specifications. At the same time, the language allows mappings to be specified procedurally, which we argue is necessary to deal effectively with the realities of poor quality, highly variable, and incomplete metadata. An additional key feature of the language is the ability to derive new mappings from existing mappings, thereby making it easy to adapt generic mappings to the idiosyncrasies of particular metadata providers. We evaluate this language on three metadata standards (ADN, FGDC, and MARC) and three corresponding collections of metadata. We also note limitations, future research directions, and generalizations of this work.

1 Introduction

The Alexandria Digital Library (ADL) project¹ has been working to develop a lightweight, federated digital library architecture for heterogeneous georeferenced information. *Federated* means that distributed, autonomous libraries can work cooperatively to provide global discovery of and access to the union of their content. *Heterogeneous* means that a library can contain multiple types of information, including remotely-sensed imagery, textual documents, executable models, and multimedia instructional materials. *Georeferenced* means that, whenever possible, each item in a library is associated with one or more regions on the Earth's surface; this information is used by ADL to provide a spatial search capability. *Lightweight* means that the

¹ <http://www.alexandria.ucsb.edu/>

burden of setting up and running ADL is small enough that groups and systems that would not ordinarily be thought of as spatial data providers (traditional library catalogs, for example), nor capable of being spatial data providers (small digital library implementations lacking spatial engines, for example), can participate in a spatial system. ADL is not itself a source of spatial data; rather, it is a system that provides a spatial orientation to heterogeneous data sources.

ADL has historically focused on creating libraries and collections via mediation techniques that map library provider services to ADL's services, and that map native collection- and item-level metadata to ADL's common metadata model. The most developed of these techniques uses a schema-mapping language and configurable query translation software to enable virtually any relational database containing metadata to be viewed as an ADL collection. Less developed techniques include prototype gateways to the Z39.50 [1] and SDLIP [22] protocols, and mediators for ad hoc situations.

This service-level mediation approach has proven effective at bringing heterogeneous information sources into the ADL fold, but we have found that it can place an intimidating burden on library providers. To fully support ADL, library providers must run multiple services (both online services that provide search functionality and offline services that gather the statistics for ADL's federated collection-level discovery service), and these services can require complex configuration and substantial operational support. To offer a lighter-weight means of ADL participation, and to take advantage of the widespread availability of metadata via OAI-PMH [18], ADL has become interested in forming collections by directly ingesting both harvested and manually-submitted metadata. Such collection-building hinges on the ability to automatically map item-level metadata to the ADL metadata views (defined in the next section), a task that forms the subject of this paper.

Many other projects have taken a similar approach to collection-building, typically by mapping metadata to Dublin Core² and providing services over that common representation. The work we describe in this paper is necessarily specific to the ADL metadata views, but as will be seen, the ADL metadata views are more challenging to map to than Dublin Core, and thus the solution presented here has broad applicability.

2 Problem Statement

The ADL architecture [14] defines three item-level metadata formats that provide to clients a uniform view of ADL content and form the basis for the ADL services. Because ADL continues to store and make available the original source (or "native") metadata, and because the lineage of the mappings from native metadata to the ADL formats is explicitly represented, we refer to the ADL formats as *views*. To give some context to ADL's metadata mapping problem we briefly describe these views.

The *bucket view*³ [13] aggregates and maps native item metadata into a few strongly-typed, high-level search indices, or "buckets." Bucket types, which govern the allowable syntactic representations and search operators, are extensible, as are the

² <http://www.dublincore.org/documents/dcmi-terms/>

³ View definitions can be found at <http://www.alexandria.ucsb.edu/middleware/dtds/>.

bucket definitions themselves, allowing collections to define collection-specific buckets. To support federation-wide interoperability, ADL defines 9 standard buckets that roughly correspond to Dublin Core elements, the key distinction being that ADL buckets are much more structured [7]. For example, the Dublin Core *coverage.spatial* qualified element suggests some encodings but ultimately may be populated by any free text, whereas the ADL spatial bucket type requires specific encodings of a handful of accepted geographic shapes.

The *access view* [16] describes the item's accessibility via zero or more "access points." Each access point corresponds to a single, independent representation of the item; different types of access points reflect fundamentally different modes of accessing content and capture the different kinds of information needed by clients to successfully use the access point. For example, a *download access point* simply returns a representation of the item, while a *service access point* allows the item to be accessed via a programmatic web service. Access points can be grouped into hierarchies to capture both alternative representations and decompositions of the item.

The *browse view* describes available browse-level representations of the item (image thumbnails, for example).

Our goal is to be able to programmatically map native metadata to these metadata views. Because most metadata mappings are declarative (for the most part, mappings are as trivial as "metadata field *A* maps to bucket *B*"), we want to be able to express mapping rules using a correspondingly concise and declarative language. However, not *all* mappings are declarative. In our experience, some mappings require recourse to a procedural language in order to express syntactic transformations (geographic coordinate and date conversions, for example) and to deal with the pervasive problems of metadata incorrectness, inconsistency, nonuniformity, and incompleteness. Fortunately, our experience has shown that such metadata problems are often relatively uniform within the context of a single provider, an observation echoed by Stvilia, et al [27], which calls for a language that can easily accommodate provider idiosyncrasies. Finally, validation of mappings is critical.

3 Related Work

Many other digital library projects have encountered the problems of assimilating distributed, heterogeneous metadata into a common framework; recent examples include NSDL [2] and OAIster [11]. Hillmann, et al, have been investigating more flexible ways of aggregating metadata from multiple sources [12]. These efforts and others have created many project-specific, ad hoc software solutions, typically implemented in Perl or XSLT, as opposed to a generic, reusable infrastructure.

Crosswalks—tables that map relationships and equivalencies between metadata schemes [29]—have been created between all the major metadata standards and are being gathered into repositories [9]. Sathish, et al, have created a visual tool for creating crosswalks [24]. From our perspective, there are three principal limitations of crosswalks. First, the representation of metadata relationships, usually being limited to equivalence, is unrealistically simple. Second, crosswalks provide no *formal* means of describing complex transformations or handling idiosyncrasies. And third, crosswalks are not directly executable. Per Godby [9], "the record structure for the

crosswalk object is a separate issue from the implementation details of metadata translation.” Our work unifies these two aspects of metadata mapping.

Given that we are mapping metadata from one XML-encoded form to another, XSLT [3] suggests itself, and indeed, it is the tool most commonly employed for this purpose. However, we find XSLT deficient for two reasons:

1. The language is too low-level and verbose; it forces the mapping developer to work at the level of XML elements as opposed to the level of metadata fields.
2. The language is computationally hamstrung. Although it is Turing-complete in theory [17], it is notoriously difficult to express computation beyond simple template matching of XML structures. Coding the kinds of simple transformations that ADL must perform (e.g., the conversion of a geographic coordinate in degrees-minutes-seconds notation to decimal degrees) ranges from extremely difficult to practically impossible.

Several groups have attempted to overcome XSLT’s deficiencies, both for general reasons and for the specific purpose of metadata mapping [10], by integrating it into a procedural language such as Java. However, this approach creates a large division between the declarative and procedural aspects of the mappings. Also, Java is generally too rigid and low-level a language for the purposes of metadata transformation; a higher-level scripting language is more appropriate.

Other groups have looked at creating new languages that operate over XML documents. The work of Manghi, et al, in developing a hybrid declarative/procedural language over XML is intriguing [19]. But operating on XML elements is fundamentally the wrong level of abstraction for our problem.

4 A Language for Mapping Metadata

We have developed a high-level language and supporting programmatic framework that allows metadata to be mapped at the level of source metadata fields and destination buckets and access points. Using the “piggyback” design pattern described by Spinellis [26], in which one language is implemented on top and in terms of another, our language is based on Python⁴, a popular, open-source scripting language. A complete description of the mapping language is outside the scope of this paper⁵. In the remainder of this section we highlight the language’s key features and characteristics; in the following section, where we evaluate the language, we provide a couple illustrative examples.

A mapping from a native metadata format to the ADL metadata views is described by an executable Python script having the form

```
from ADL_mapper import *
input()
...declarations...
output()
```

⁴ <http://www.python.org/>

⁵ A language tutorial can be found at <http://www.alexandria.ucsb.edu/mm/tutorial.html>.

The `import` statement loads the necessary infrastructure into the current Python workspace; the `input` statement reads and parses the native metadata; and the `output` statement performs all processing and outputs the ADL metadata views.

Between the `input` and `output` statements may be placed 24 different kinds of declarations that govern mappings and processing. Since the mappings are executable Python, these “declarations” are Python procedure calls. However, these procedure calls perform no immediate action, but only record the actions to be taken when processing finally occurs at the end of the script, and the calls can generally occur in any order. Thus the calls have all the essential characteristics of declarations, resulting in a language that is simultaneously both declarative and procedural in nature. Declarations can be emplaced in a procedural context (e.g., inside conditional constructs), and can embed and call on procedural code.

The bucket types, buckets, and associated vocabularies and thesauri are not pre-defined by the language, but can be declared within the language. (Each bucket type also requires a Python module adhering to a simple API to define validation and encoding of the type.) Using Python’s module mechanism, these kinds of background declarations can be packaged into standard modules that a mapping simply imports.

There are several ways to specify mappings, but they all share the same general processing model. A query (described below) is executed to form tuples of metadata values which are then passed through a series of mapping-specified filter and converter functions. Filters perform arbitrary processing and may reject a mapping, while converters serve as pattern recognizers (for example, to process dates, one would create a converter function for each supported date format). Tuples that survive any filtering and conversion are passed to a validator. If valid, they are appropriately encoded.

To identify metadata fields in the source metadata document and to form the tuples used by the mapping framework, we have defined a simple query language built on XPath⁶ that allows Cartesian products and “outer joins” [28] to be formed from relative and absolute XPath expressions and constant values. This query language is not as powerful as XQuery⁷ or DSQL [25], but it has proven sufficient for our purposes and has the twin virtues of being compact and making simple queries simple to express.

Additional language features include the ability to express mapping-specified requirements (which generate errors) and expectations (which generate warnings) to be checked by the mapping framework, which are useful as sanity checks when processing large quantities of metadata. The language also supports “opportunistic” mappings, i.e., mappings that are performed when validation is satisfied but silently ignored when not.

As described thus far, the language provides an easy and flexible means of describing mappings declaratively, and inserting procedural code as necessary. But perhaps the most significant language feature is an inheritance-like feature that allows map-

⁶ <http://www.w3.org/TR/xpath>

⁷ <http://www.w3.org/TR/xquery/>

pings to be derived from other mappings. Derived mappings use Python’s import mechanism and take the form

```
from ADL_mapper import *
input()
import parent mapping
...additional declarations...
output()
```

Derived mappings can thus add to parent mappings. In addition, the language provides declarations that allow the derived mapping to augment, undo, or override any parent mapping, requirement, or expectation. This makes it possible to succinctly express small refinements to comprehensive, generic mappings.

5 Evaluation

The mapping language was evaluated by creating mappings for several well-known metadata standards. These mappings were then exercised on collections of real-world metadata. The results of the mappings were visually inspected and, in the case of large collections, analyzed by software written for the purpose.

It should be noted that the mere *ability* to perform a mapping is not a criterion of our evaluation. Mappings can be created by any full-featured programming language, and in simple cases, by less powerful languages such as XSLT. Rather, our criteria are the qualitative considerations that apply to programming language design [23, 8]: writability and readability; brevity; expressiveness; appropriateness of the level of abstraction to the domain; and orthogonality.

In the following subsections we describe three evaluation tests⁸.

5.1 ADN/DLESE

In a first test, we created a mapping for the ADN (ADEPT/DLESE/NASA) metadata standard⁹, which has been used primarily by DLESE¹⁰, and we exercised the mapping on 5,061 ADN records representing DLESE’s entire corpus at the time of the experiment. DLESE exercises considerable review and quality control over its records, resulting in relatively complete and homogeneous metadata, and we had already performed a preliminary analysis of mapping ADN to the ADL buckets [15], so the mapping was easy to create. The mapping specifies 17 simple declarations to map ADN fields to 8 ADL buckets and 2 DLESE-specific buckets. The language fragment shown below maps ADN’s notion of temporal coverage to the ADL `dates` bucket. For each ADN `<timeAD>` element, a (begin, end) time range tuple formed from the `<begin>` and `<end>` sub-elements’ date attributes is mapped to the bucket. Some procedural code (a filter function) converts ADN’s “Present” time keyword to ADL’s numeric representation of the same concept.

⁸ The complete mappings can be viewed at <http://www.alexandria.ucsb.edu/mm/>.

⁹ <http://www.dlese.org/Metadata/adn-item/>

¹⁰ <http://www.dlese.org/>

```

def convertPresent (v):
    if v[1].lower() == "present":
        return (v[0], "9999")
    else:
        return v
map("adl:dates",
    ["/itemRecord/temporalCoverages/timeInfo/timeAD",
     "begin@date", "end@date"],
    prefilters=convertPresent)

```

For comparison, XSLT code that performs a subset of the same mapping is shown below. The XSLT version performs only a subset because it does not include the conversions, standardizations, duplicate eliminations and other consolidations, and validation that the mapping language implicitly performs.

```

<xsl:template
  match="itemRecord/temporalCoverages/timeInfo/timeAD">
  <temporal-value>
    <range>
      <begin>
        <xsl:value-of select="begin/@date"/>
      </begin>
      <end>
        <xsl:choose>
          <xsl:when
            test="translate(end/@date,
              'ABCDEFGHJKLMNOPQRSTUVWXYZ',
              'abcdefghijklmnopqrstuvwxy')='present'">
            <xsl:text/>9999<xsl:text/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:value-of select="end/@date"/>
          </xsl:otherwise>
        </xsl:choose>
      </end>
    </range>
  </temporal-value>
</xsl:template>

```

5.2 FGDC/Maya

In a second test, we created a mapping for the MesoAmerican Research Center's¹¹ digital geographic database for the Maya forest region. This is a collection of approximately 60 images, maps, and models having metadata conforming to the ESRI profile [5] of the FGDC Content Standard for Digital Geospatial Metadata [6]. Because this collection uses a well-known metadata standard but also exhibits several problems endemic to real-world metadata, we exploited the mapping language's inheritance feature to develop both a generic mapping for the FGDC standard and a derived Maya-specific mapping. The generic FGDC mapping uses 23 declarations to map FGDC fields to the 9 standard ADL buckets. In addition, 10 converter and filter functions are used to convert different date formats and perform other minor transformations.

¹¹ <http://www.marc.ucsb.edu/>

An interesting aspect of the FGDC mapping is how thesaurus terms are mapped. Mapping thesauri has been explored at length by the NKOS¹² and digital library communities. Our mapping language easily supports mapping terms from a source thesaurus to a destination thesaurus, using boolean combinations and partial equivalences as suggested by Doerr [4] and formalized by SWAD [21]. But in the case of ADL's `types` and `formats` buckets, there is often no comparable source thesaurus to map *from*; instead, a mapping must be inferred from various other pieces of information. For example, the vocabulary associated with the ADL `formats` bucket distinguishes whether an item is available online or offline (or both); if it is offline, whether it is non-digital or digital, and if digital, what type of media it is available on; and if it is online, in what formats it is available. There is no single FGDC vocabulary that captures all this information, so our FGDC mapping is really 5 mappings that look at different aspects of the item's distribution-related metadata. An example is shown below, where the presence of an online linkage is used as an indication that the item is available online. Other mappings look for more specific information.

```
if present("/metadata/idinfo/citation/citeinfo/onlink"):
    mapConstant("adl:formats",
               ("ADL Object Formats", "Online"))
```

In a post-processing step, the mapping framework gathers all such mappings and consolidates the information into a minimal set of terms. For example, if terms “Online” and “Image” both get mapped to the `formats` bucket, the “Online” mapping will be elided as a broader term of “Image.” More sophisticated post-processing is possible via programmatic hooks.

As we noted previously, like many real-world collections, the Maya collection appears to adhere to a metadata standard but in fact exhibits a number of deficiencies—omissions, idiosyncrasies, obvious mistakes—that require correction:

1. Item titles are simple, extension-less filenames, e.g., “mex250kr.” While possibly appropriate within the context of the collection (perhaps not even then), such cryptic titles do not serve users well in the larger context of a distributed digital library.
2. For some of the items, the FGDC element that describes how an offline, non-digital form of the item can be accessed erroneously contains a licensing agreement. Since the generic FGDC mapping uses this element as evidence that the item is available offline, this error causes the item to appear to be available offline when in fact it is not.
3. The name of the collection's creator, being an implicit piece of collection-level metadata, is not listed among the item-level originators.

Using the mapping language, we were able to correct these flaws as follows:

1. Item titles were improved by appending to the FGDC-specified title mapping a filter function that constructs a new title from the given title and several other pieces of information within the metadata.
2. The mapping of the erroneously-populated element was “unmapped” (disabled).
3. An additional mapping to the originators bucket of the collection creator's name (as a collection-wide constant) was added.

¹² Networked Knowledge Organization Systems/Services; <http://nkos.slis.kent.edu/>.

Thus by starting with a generic mapping for the FGDC standard, the mapping language's inheritance feature allowed us to create a mapping for the Maya collection with just a few simple declarations.

5.3 MARC/Maps

In a third test, we created a mapping for the MARC 21¹³ bibliographic metadata standard (specifically, the MARCXML¹⁴ encoding) and applied the mapping to 47,280 MARC records describing map holdings within UCSB's Davidson Library.

Our mapping is only a prototype; due to MARC's long history, all-inclusive nature (consider that it includes the entire FGDC standard as a subset) and wide range of applications, a complete mapping will take substantial effort to complete. The difficulty here is due simply to the large number of mappings required. With respect to evaluating the mapping language, though, we note that all mappings were particularly trivial to declare given MARCXML's flat structure.

5.4 Performance

The performance of the system is not stellar, but is adequate for our purposes. The time required to perform a mapping is dependent on a number of variables, chief among them the size and complexity of the input XML document and the amount of processing specified by the mapping. To give a rough order of magnitude, mapping a 12KB, 220-element ADN metadata record requires approximately one second on a 1.8GHz dual-processor PowerMac G5. Instrumentation revealed that from two-thirds to three-quarters of the processing time is taken up by Python's XML parser¹⁵ reading and parsing the input document and building a complete DOM tree, and by writing the output XML document. Given that our declarative approach is intrinsically dependent on performing multiple XML queries over this DOM tree, there is little opportunity to significantly increase the performance of the system as presently implemented.

6 Future Work

Our evaluation of the mapping language demonstrated that the language is highly suitable for mapping descriptive metadata to ADL's bucket view. However, prototype mappings to the access and browse views were less successful.

Access-related information is difficult to handle for several reasons. To begin with, such information is poorly supported by metadata standards, if at all. ADN, MARC, and Dublin Core are capable of describing item access by simple URLs only, which is insufficient for the rich kinds of access mechanisms associated with geospatial and scientific data [16]. FGDC provides better support, but in our experience, its access-related fields are too unreliably populated to be of any practical use. METS¹⁶ addresses some of these limitations, but we find there are subtle differences between describing the structure of an item and access to that item.

¹³ <http://www.loc.gov/marc/>

¹⁴ <http://www.loc.gov/standards/marcxml/>

¹⁵ We use the PyXML package, available at <http://pyxml.sourceforge.net/>.

¹⁶ <http://www.loc.gov/standards/mets/>

The underlying issue here, regardless of the native metadata format, is that access-related information is intimately related to both the intrinsic structure of the item and what we might call the item's *instantiation* or *emplacement* in the library, i.e., the item's relationship to the library services that provide access to the item. To effectively map access-related metadata, we believe the mapping language requires additional inputs that capture the context of library servers and services. In addition, validation (link-checking, etc.) of access-related mappings is critical.

7 Conclusion

In automating ADL's collection building and statistics gathering functionality, a need arose for automated mapping of item-level, XML-encoded metadata to the ADL metadata views. This metadata mapping problem can be characterized as a complex specification problem that is largely—but not entirely—declarative in nature, and for which it must be possible to perform arbitrary procedural computations over metadata values. In addition, to operate on real-world metadata any mapping framework must support validation of mappings and the ability to easily adapt mappings to the idiosyncrasies of particular metadata providers.

To answer this need we have developed a general framework for mapping metadata. In this framework, a mapping is implemented as a Python script, and individual mappings are expressed as procedure calls within the script. By delaying all processing to the end of the script, the procedure calls effectively become declarations and the procedures defined by the mapping framework effectively form a language. This framework allows mappings to be specified at a higher level than, say, XSLT, and provides many other features such as implicit value conversions and validation. The ability to derive mappings from more generic mappings facilitates idiosyncratic adaptations.

Our experience with this approach has shown that it is a good match for the problem. Mapping access-related metadata remains a challenge, however, and requires that the mapping framework understand additional concepts such as servers and services, and that it be integrated into a library's ingest workflow.

Our approach has its limitations, of course. Although mappings are easy to express, artifacts of Python syntax intrude significantly; essentially, a mapping developer must also be a Python programmer. This is not a significant concern, for as we have argued in this paper, mappings of real-world metadata often require recourse to a procedural language to handle the various problems encountered, and Python is a better language than most for that purpose. Still, a custom mapping language could make mappings less syntactically stilted [20]. But then, such a language would require a custom parser, a substantial development task. By contrast, our Python-based language is implemented in fewer than 1,000 lines of Python code.

The metadata mapping language and framework software are currently specific to the ADL metadata views, but with only minor modifications both could easily be adapted to another mapping target, especially one requiring translation and validation of relatively structured metadata fields. More broadly, we believe that the hybrid declarative/procedural approach we have taken has applications in many other domains. Any specification problem that is largely—but not entirely—declarative in nature would be a candidate for this approach.

References

1. ANSI Z39.50-1995. *Information Retrieval (Z39.50) Application Service Definition and Protocol Specification*. <http://www.loc.gov/z3950/agency/markup/markup.html>.
2. William Y. Arms, Naomi Dushay, Dave Fulker, and Carl Lagoze. "A Case Study in Metadata Harvesting: the NSDL." *Library Hi Tech* **21**(2) (June 2003): 228-237. <http://dx.doi.org/10.1108/07378830310479866>.
3. James Clark (ed.). *XSL Transformations (XSLT)*. Version 1.0. <http://www.w3.org/TR/xslt>.
4. Martin Doerr. "Semantic Problems of Thesaurus Mapping." *Journal of Digital Information* **1**(8) (March 2001). <http://jodi.ecs.soton.ac.uk/Articles/v01/i08/Doerr/>.
5. Environmental Systems Research Institute (ESRI), Inc. *ESRI Profile of the Content Standard for Digital Geospatial Metadata*. March 2003. <http://www.esri.com/metadata/esriprof80.html>.
6. Federal Geographic Data Committee. FGDC-STD-001-1998. *Content Standard for Digital Geospatial Metadata*. June 1998. <http://www.fgdc.gov/metadata/contstan.html>.
7. James Frew and Greg Janée. *A Comparison of the Dublin Core Metadata Element Set and the Alexandria Digital Library Bucket Framework*. 2003. <http://www.alexandria.ucsb.edu/~gjanee/archive/2003/dc-adl.pdf>.
8. Carlo Ghezzi and Mehdi Jazayeri. *Programming Language Concepts*. 2nd ed. New York: John Wiley & Sons, 1987.
9. Carol Jean Godby, Jeffrey A. Young, and Eric Childress. "A Repository of Metadata Crosswalks." *D-Lib Magazine* **10**(12) (December 2004).
10. Damien Guillaume and Raymond Plante. "Declarative Metadata Processing with XML and Java." *Astronomical Data Analysis Software and Systems X. ASP Conference Series* **238** (2001). <http://www.adass.org/adass/proceedings/adass00/O6-03/>.
11. Martin Halbert, Joanne Kaczmarek, and Kat Hagedorn. "Findings from the Mellon Metadata Harvesting Initiative." *Proceedings of the Seventh European Conference on Research and Advanced Technology for Digital Libraries (ECDL)* (Trondheim, Norway; August 2003): 58-69.
12. Diane Hillmann, Naomi Dushay, and Jon Phipps. "Improving Metadata Quality: Augmentation and Recombination." *DC-2004: International Conference on Dublin Core and Metadata Applications* (Shanghai, China; October 2004). http://purl.org/metadata-research/dcconf2004/papers/Paper_21.pdf.
13. Greg Janée, James Frew, Linda L. Hill, and Terence R. Smith. "The ADL Bucket Framework." *Third DELOS Workshop on Interoperability and Mediation in Heterogeneous Digital Libraries* (Darmstadt, Germany; September 2001). <http://www.ercim.org/publication/ws-proceedings/DelNoe03/13.pdf>.
14. Greg Janée and James Frew. "The ADEPT Digital Library Architecture." *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)* (Portland, Oregon; July 2002): 342-35. <http://doi.acm.org/10.1145/544220.544306>.
15. Greg Janée. *ADN Metadata Mapping*. October 2003. <http://www.alexandria.ucsb.edu/~gjanee/archive/2003/adn-mapping.html>.
16. Greg Janée, James Frew, and David Valentine. "Content Access Characterization in Digital Libraries." *Proceedings of the 2003 Joint Conference on Digital Libraries (JCDL)* (Houston, Texas; May 2003): 261-262. <http://doi.acm.org/10.1145/827140.827185>.

17. Stephan Kepser. "A Simple Proof for the Turing-Completeness of XSLT and XQuery." *Extreme Markup Languages 2004*.
<http://www.mulberrytech.com/Extreme/Proceedings/html/2004/Kepser01/EML2004Kepser01.html>.
18. Carl Lagoze and Herbert Van de Sompel (eds.). *The Open Archives Initiative Protocol for Metadata Harvesting*. Version 2.0 (June 14, 2002).
<http://www.openarchives.org/OAI/openarchivesprotocol.html>.
19. Paolo Manghi, Fabio Simeoni, David Lievens, and Richard Connor. "Hybrid Applications over XML: Integrating the Procedural and Declarative Approaches." *Fourth ACM CIKM International Workshop on Web Information and Data Management (WIDM)* (McLean, Virginia; November 2002). <http://doi.acm.org/10.1145/584931.584935>.
20. David Mertz. "Create declarative mini-languages: Programming as assertion rather than instruction." *Charming Python* (February 27, 2003).
<http://www.ibm.com/developerworks/library/l-cpdec.html>.
21. Alistair Miles and Brian Matthews. *Inter-Thesaurus Mapping*. Retrieved February 22, 2005. <http://www.w3.org/2001/sw/Europe/reports/thes/8.4/>.
22. Andreas Paepcke, Robert Brandriff, Greg Janée, Ray Larson, Bertram Ludäscher, Sergey Melnik, and Sriram Raghavan. "Search Middleware and the Simple Digital Library Interoperability Protocol." *D-Lib Magazine* 6(3) (March 2000).
23. Eric Steven Raymond. *The Art of Unix Programming*. Boston: Addison-Wesley, 2004.
24. K. Sathish, K. Maly, M. Zubair, and X. Liu. "RVOT: A Tool For Making Collections OAI-PMH Compliant." *Proceedings, 5th Russian Conference on Digital Libraries (RCDL)* (St. Petersburg, Russia; October 2003).
<http://RCDL2003.spbu.ru/proceedings/A5.pdf>.
25. Arijit Sengupta and M. Dalkilic. "DSQL - an SQL for structured documents." *Proceedings, 14th International Conference, CAiSE 2002* (Toronto, Canada; May 2002): 757-760.
26. Diomidis Spinellis. "Notable Design Patterns for Domain-Specific Languages." *Journal of Systems and Software* 56(1) (February 2001): 91-99.
<http://www.dmst.aueb.gr/dds/pubs/jrnl/2000-JSS-DSLPatterns/html/dslpat.html>.
27. Besiki Stvilia, Les Gasser, Michael B. Twidale, Sarah L. Shreeves, and Tim W. Cole. "Metadata Quality for Federated Collections." *Proceedings of the 9th International Conference on Information Quality (ICIQ)* (Boston, Massachusetts; November 2004): 111-125.
28. Jeffrey D. Ullman and Jennifer Widom. *A First Course in Database Systems*. 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, 2002.
29. Mary S. Woodley, et al. *DCMI Glossary*. September 15, 2003.
<http://dublincore.org/documents/usageguide/glossary.shtml>.

Using a Metadata Schema Registry in the National Digital Data Archive of Hungary

Csaba Fülöp, Gergő Kiss, László Kovács, and András Micsik

MTA SZTAKI,
Computer and Automation Research Institute,
of the Hungarian Academy of Sciences,
Department of Distributed Systems,
H-1111 Budapest XI. Lágymányosi u. 11. Hungary
{csaba.fulop, gergo.kiss, laszlo.kovacs, micsik}@sztaki.hu

Abstract. The National Digital Data Archive (NDDA) is an ongoing initiative of the Hungarian government that makes Hungary's national cultural assets available in digital form. The NDDA features a decentralized OAI-based network of archives and service providers facilitating discovery and access to digitized objects. Authors' participation in the project is described including the implementation of an NDDA service provider. This service provider is connected with an RDF-based metadata schema registry enabling the service to automatically adapt to the metadata schemas defined within the NDDA.

1 Introduction

The National Digital Data Archive (NDDA) is an ongoing initiative of the Hungarian government that makes Hungary's national cultural assets available in digital form [1]. Despite its name, NDDA is not a digital archive itself, rather a program that will enable a higher degree of co-operation at the archive levels, and a better integration at client levels. The motto is: „The NDDA is the nation's digital data store.”

The NDDA places emphasis on self-organisation and co-operation among communities. The goal is to establish the technical infrastructure for the knowledge-based society where discovery and access to data and information in digital format becomes an everyday need. Actions are supported on different levels:

- Definition of key concepts and structures such as namespaces, metadata schemas, identifiers.
- Definition of the building blocks of the infrastructure and example implementations of these components.
- Digitization of cultural assets: museums, radios, music archives, libraries, etc. are supported in the process of converting their valuable material into digital format.
- Establishing digital archives and connecting them to the NDDA.
- Creation of new services for the easy and unified access to digitized assets.

MTA SZTAKI Department of Distributed Systems participates actively in the development of the NDDA at various levels: in the work of the advisory committee, in

the design of metadata schemas, connecting archives to the network and implementing unified services. In this paper we present our view of the NDDA and our contributions to the initiative.

2 Architecture of the NDDA

The basic concept of the NDDA is a decentralized OAI-based network of archives and service providers [2]. In a broad sense, there are three types of information providers in the network:

- Data providers make metadata available for harvesting. They may also make their data downloadable either freely or with some restrictions.
- Service providers offer services for the public or for other NDDA components.
- Protocol providers serve core schemas, protocols needed for interoperability in the network.

Data providers, usually operated by the maintainers of the archives, implement the OAI Protocol for Metadata Harvesting (OAI-PMH) using NDDA metadata schemas instead of the plain old `oai_dc` schema. OAI-PMH does not contain any mechanism for the retrieval of the data. A usual solution was selected for this problem: the Identifier metadata element contains the URI for the downloadable resource. In this way archives provide metadata and data as well for the NDDA, thus facilitating both discovery and access to stored content.

Some service providers are planned to provide basic services usable by the whole network, not only by the clients. Descriptions for locations and persons are provided as separate services, as well as an ontology of Hungarian subject terms. These controlled databases can be harvested through OAI-PMH by service providers and archives. Archives may use this information for the improvement of their metadata definition process. Persons, locations or terms may be referred from these services in the subject, coverage, creator, contributor and other metadata elements. Service providers may use this information for the improvement of their query and browse facilities.

Protocol providers are primarily registries and repositories of internal standards for NDDA. These internal standards may include for example metadata schemas and protocol extensions.

Up to the date of paper submission, two regular service providers exist in the network, a generic service provider for authority records (service providers for geographical locations and Hungarian thesaurus are under development), and a protocol provider, which stores metadata schema definitions. 20 archives are available within the NDDA, with more than 500,000 metadata records. The number of resources available digitally is almost ten percent of all metadata records. Data types represented in our service provider are texts (cca. 60%), images (cca. 25%), video (cca. 2%), audio (cca. 1%) and statistical data (nearly 10% of the records contain no type information).

The initiative decided to create its own national metadata schemas. These schemas are based on DCMI Metadata Terms using element refinements and encoding schemas. This ensures more precise description of metadata than the standard, „unqualified” `oai_dc` schema. Furthermore, a specialized metadata schema is defined for each

genre of stored items in NDDA. Currently, there are metadata schemas for textual documents, images, audio and video programmes.

3 Implementation of a Service Provider for the NDDA

Our department has implemented the first public OAI-based interconnection of libraries in Hungary [12]. Using the results of this pilot project, we decided to create a service provider for the NDDA. Metadata schemas did not stabilize until the development started which gave us the idea to automate metadata schema management in our prototype. Metadata schemas are thus dynamically loaded from a schema registry into the NDDA service provider (Fig. 2).

3.1 Schema Registry for Metadata

The authors previously participated in the CORES project, which provided a solution for automatic schema dissemination [3]. As part of the European Community funded IST Semantic Web Technologies programme, the CORES project has promoted the use of metadata schema registries to support the disclosure, discovery and navigation of information about metadata element sets stored as schemas distributed on the Web. Such a “schema navigation service” provides users (both human and software) with information about existing metadata element sets and the terms used within them. In particular, it assists implementers in locating and re-using existing schemas.

The CORES project implemented a registry infrastructure with the following components:

- A graphical schema creation tool: this tool facilitates the discovery and re-use of existing metadata schema elements and definitions by drag-and-drop and graphical editing. With this tool users are able to create proper schema definitions without learning schema definition languages such as RDF Schema.
- An API for the remote manipulation of schemas in the registry. The schema creation tool also uses this API to upload schemas to the registry.
- A schema registry, which stores schemas in an RDF database, generates various browsable and searchable representations of schemas and their relations.

Metadata schemas are constructed as RDF schemas [4] following the rules for creation of metadata application profiles [5]. Briefly, schemas are built using elements from existing element sets (e.g. Dublin Core), where each element can be refined in multiple ways:

- Permitted encoding schemes and values may be specified (e.g. W3CDTF for dates)
- Element definitions can be semantically narrowed
- The obligation and maximum occurrence of elements can be changed

The model used for storing metadata schemas is shown on Figure 1. Element sets and encoding schemes are the basis for schema construction. Application profiles select and refine these, thus creating new, specialised metadata schemas. Agencies play the role of publisher and maintainer of schemas, versioning and maintenance information of these schemas are stored as administrative data. It is possible to annotate all

schema elements, which provides a way of sharing knowledge and practice among users and developers of schemas. The CORES toolkit is publicly available as a demonstration service on our department web server¹.

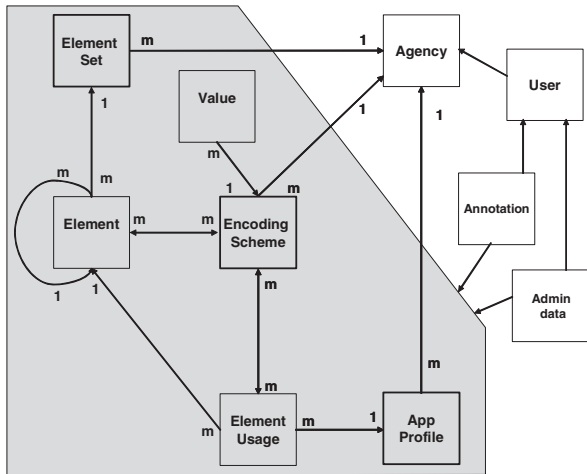


Fig. 1. The CORES model for metadata schema registry

3.2 The Harvester Module

The harvester periodically visits NDDA archives and collects metadata records. The metadata schema used for the available resources can be retrieved using OAI Protocol for Metadata Harvesting. The harvester prefers to fetch the record in an NDDA-defined schema, but plain Dublin Core records can be harvested as well. As the schema of the metadata record is known, the harvester can validate the record against the schema definition retrieved from the registry. During this validation we can identify the following problems with metadata records:

- Use of not allowed element or element refinement,
- Use of not allowed encoding scheme,
- Use of not allowed data types,
- Restrictions not fulfilled: for example a mandatory element is missing, a non-repeatable element is repeated.

When an improper element refinement is used, it can be replaced with its parent element. Inappropriate values can sometimes be automatically converted to the correct type or encoding scheme. Other types of the problems listed above cannot be corrected automatically in a safe way.

As our metadata storage and query engine are flexible enough to deal with such data inconsistencies, non-conforming elements are usually kept in their original format. Information loss is considered more disadvantageous than loose application of

¹ <http://cores.dsd.sztaki.hu>

schemas. In case of any problem with the metadata its publisher is notified about the non-conforming records.

Considering the small number of archives in NDDA so far and the close cooperation with these archives the use of metadata schemas is satisfactory. Generally, the problem is more about the incompleteness of metadata than about the conformance to metadata schemas. However, we make good use of automatic correction of values for Type, Language and Date elements. The most typical errors are:

- the use of Hungarian, Magyar, etc. as the value for Language element instead of ISO639-2,
- not using DCMIType vocabulary in the Type element,
- various non-standard date formats in Date elements.

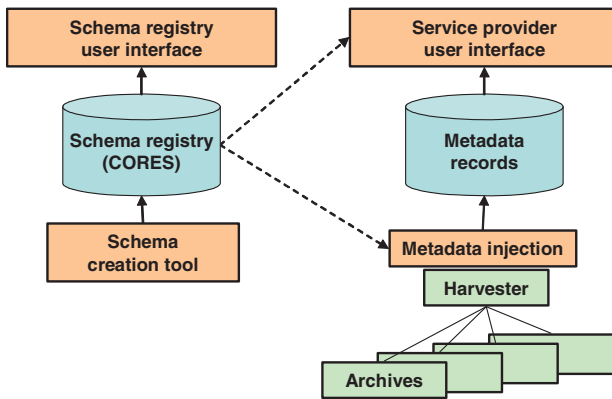


Fig. 2. Architecture of NDA@SZTAKI system

After validation metadata values are stored in a database with element refinements, encoding schemes and language identifiers all converted to an internal naming scheme. This schema-independent internal representation makes it possible to handle some of the schema modifications in the future.

Parallel to the introduction of new metadata schemas existing schemas are also bound to change. Such changes may break the schema-conformance of thousands of records. Besides automatic notification about schema changes to publishers, a system can also adapt to some of these changes. Renaming is considered as bad practice in this world, but with our extra mapping of names onto internal identifiers such changes can be handled transparently. A change in the allowed encoding scheme requires new data conversion algorithms, which normally needs human intervention. An interesting research question not elaborated here is the handling of semantic changes in a metadata schema, i.e. the shifting of the desired meaning of an element. In general, additions to a schema can be handled automatically in this system, while semantic changes may easily remain invisible. Reducing a schema presents the problems of information loss and backward incompatibility.

3.3 Query Interface

The user interface of the NDDA service provider² is accessible using Hungarian and English languages and offers various browsing and searching facilities for the nearly 500,000 metadata records. The service has been available since May 2004 and currently has an average of 5000 hits per day.

The screenshot shows a query construction window with the following structure:

	element . qualifier	relation	value	encoding scheme	
O R	type . [any]	is	Sound	DCMIType	- +
	type . [any]	is	Text	DCMIType	- +
AND					+
	publisher . place of publication	contains	Budapest	---	- +
AND					+
	date . available	later than	2005 / 01 / 01 . . .	W3CDTF	- +

Fig. 3. An example query in the query construction window

Connecting the schema registry to the query interface enables us to provide the latest schemas and schema elements for query composition. This affects the list of elements in the simple search facility, which is a minimalist Google-like solution with one text input field and simple switches for query mode variants.

We also wanted to demonstrate the full potential of searching among metadata records of multiple schemas. The so-called advanced query interface enables the user to construct arbitrary Boolean query expressions in a simple and transparent way (Fig. 3). Queries are produced in conjunctive or disjunctive normal form, that is, queries are either in the form $(A \text{ or } B) \text{ and } (C \text{ or } D)$...or in the form $(A \text{ and } B) \text{ or } (C \text{ and } D)$...where A, B, C, D are simple (atomic) query expressions like *creator contains John*. Atomic expressions are composed of the selected schema element and the value sought. Elements may have a refinement selected and the value may be given using an encoding scheme. The type of match is a context sensitive selection box offering relations such as ‘later than’, ‘earlier than’ for dates, ‘contains’, ‘begins with’ or ‘does not contain’ for texts, etc. Negation is also possible as matching mode within atomic expressions.

A query construction process using all available features of the user interface contains the following steps:

- Restricting the query to run against selected archives
- Select metadata schemas for use during query construction
- Building the query expression
- Assign sorting order for the result
- Execute the query, browse results

² <http://nda.sztaki.hu>

- Switch back to query expression editor, refine query and execute it.
- Save the constructed query in personal profile for later reuse

The query expression editor provides a graphically emphasized view of the edited query where new atomic expressions can be easily added or removed using the plus and minus icons. Users may also include their saved queries into a newly built query: the selected query will be an atomic expression of the new query.

A functionality for users more familiar with Boolean logic is the ability to switch between conjunctive and disjunctive normal forms. This can be done in two ways: either the AND and OR operators are simply exchanged (syntactical) or the expression is transformed into an equivalent expression in the other format (semantic).

During 4 months of test operation, we found that most user activities are for browsing archives and sets (20%) and viewing metadata records (12%). Search results were retrieved in 5% for simple search and in 0.3% for advanced search of all accesses. 29% of accesses were initiated from Google hit lists as we allowed Google to index the contents of our service. 23% of user actions followed a link to the content or to another related webpage.

4 Related Work

The Stanford InfoBus [7] is a well-known architecture where archives with different metadata schemas (and query semantics) are unified as a single resource for the users. In this case user queries are automatically translated for each archive and their responses are merged into a query result. With the launch of the Open Archives Initiative the approach was shifted and the focus moved on standardized metadata export for unified query services. However, the use of a single metadata schema is not feasible in many scenarios. Instead of returning to the old world of multiple independent schemas, schema refinement and specialisation appeared in the evolution of Dublin Core [5]. A formal model for the description of schema reuse and schema refinement based on RDFS help to regulate this process and provide a mapping between schema elements [6]. A software framework supporting this model has also been developed [3]. Current paper describes the next step in this direction: connecting the metadata repository and the metadata registry in order to ensure metadata correctness and proper discovery with respect to registered metadata schemas.

Although best-match, ranked-output retrieval techniques are considered superior to exact-match systems based on Boolean queries in terms of recall and precision [11], Boolean queries are usually much preferred among professional searchers such as librarians [10]. Our system provides both solutions: ranked results using simple search and Boolean query construction within the advanced search interface. Precise query formulation and exact matches, for which the implemented interface is essential, are often requirements in libraries and archives. We can mention Query By Templates (QBT) [9] or VQuery [8] to show how many different methods are available to help Boolean query construction. With QBT users can attach search terms to visually separated parts of a document (e.g. title, author and subheading) using a document template. VQuery also produces query expressions with the help of Venn diagrams: each search term is entered in a separate circle and the positions of the circles establish the Boolean operators between the terms. The main advantage of the query expression

editor described in this paper compared to the above mentioned intuitive and experimental methods is its simple and minimalist design. It basically works with any web browser and does not require special graphical capabilities, yet it provides a view of the query, which is easy to understand and modify.

5 Conclusion

The NDDA has an essential role in the dissemination of Hungarian cultural objects and is a unique initiative both in its scope and in its architecture. It experiments with the principles of OAI and self-organisation at national level. Its distributed architecture stands as an example for new projects in tourism support, healthcare and administration.

In an initiative with such a broad focus the evolution of metadata schemas is a natural phenomenon. The presented service provides a working example for the connection of metadata registries and metadata repositories in real world settings. Furthermore, an easy-to-use interface for the construction of query expressions using multiple metadata schemas is also presented. This solution couples the results of the CORES project with our previous pilot experiments for library interconnections using OAI in Hungary. The connection provides the benefits of automatic checking of harvested metadata records and automatic adaptation of the query interface to the latest metadata schemas.

References

1. National Digital Data Archive (NDDA). <http://www.nda.hu>
2. A. Micsik: Open Archives Initiative: applications in Hungary and worldwide. Netties 2004 Conference, Budapest, 27-29 October 2004
3. R. Heery, P. Johnston, Cs. Fülöp, A. Micsik: Metadata schema registries in the partially Semantic Web: the CORES experience. 2003 Dublin Core Conference, DC-2003, 28 Sept - 2 Oct 2003, Seattle, Washington USA
4. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004, <http://www.w3.org/TR/rdf-schema/>
5. R. Heery, M. Patel: Application Profiles: mixing and matching metadata schemas. *Ariadne* 25 (2000 September). <http://www.ariadne.ac.uk/issue25/app-profiles>
6. R. Heery, P. Johnston, D. Beckett, D. Steer: The MEG Registry and SCART: complementary tools for creation, discovery and re-use of metadata schemas. Proceedings of the International Conference on Dublin Core and Metadata for e-Communities, 2002.
7. M. Baldonado, C. K. Chang, L. Gravano, A. Paepcke: The Stanford Digital Library metadata architecture. *International Journal on Digital Libraries*, Volume 1, Issue 2, Sep 1997.
8. S. Jones, S. McInnes, M. S. Staveley: A graphical user interface for Boolean query specification. *Int. Journal on Digital Libraries* 2(2), Springer-Verlag, 1999, pp. 207-223
9. A. Sengupta, A. Dillon: Query by Templates: A Generalized Approach for Visual Query Formulation for Text Dominated Databases. 4th International Forum on Research and Technology Advances in Digital Libraries (ADL '97), May 7-9, 1997, Washington, DC

10. D. Byrd and R. Podorozhny: Adding Boolean-quality control to best-match searching via an improved user interface. Technical Report IR-210, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, Massachusetts, 2000.
11. Nicholas J. Belkin and W. Bruce Croft: Retrieval techniques. In Martha E. Williams, editor, ARIST chapter 4, pages 109-145. Elsevier, 1987.
12. HEKTÁR project homepage. <http://hektar.sztaki.hu>

Finding Appropriate Learning Objects: An Empirical Evaluation

Jehad Najjar, Joris Klerkx, Riina Vuorikari, and Erik Duval

Computer Science Department, K.U.Leuven,
B-3001 Leuven, Belgium
{najjar, jorisk, Riina, erikd}@cs.kuleuven.ac.be

Abstract. The challenge of finding appropriate learning objects is one of the bottlenecks for end users in Learning Object Repositories (LORs). This paper investigates usability problems of search tools for learning objects. We present findings and recommendations of an iterative usability study conducted to examine the usability of a search tool used to find learning objects in ARIADNE Knowledge Pool System [1]. Findings and recommendations of this study are generalized to other similar search tools.

1 Introduction

Much of current research on content reuse for learning focuses on the notion of learning objects [9]. Learning objects are stored in Learning Object Repositories (LORs) (such as ARIADNE [1], EdNa [10], SMETE [24], or Merlot [16]) to be used for different kinds of educational purposes and by different end users. One of the bottlenecks for end users is the challenge of finding appropriate learning objects in the current LORs [9]. There are several reasons why users cannot easily find their appropriate objects, among which that the current search tools are not user friendly and require users to provide too much information in order to locate relevant objects.

In Ariadne [1], over the last decade we have been researching concepts, techniques and tools to facilitate finding appropriate learning objects. More and more, we are carrying out empirical studies to evaluate tools used to search or index learning objects.

Empirical evaluation has been used to evaluate and improve the use and usability of different tools in different contexts, such as internet browsing [5][21] and digital libraries [11]. The context of learning objects lacks such important studies [9]. In our earlier works [17][18] we investigated the behavior of learning object indexes and searchers by analyzing the usage logs. We further applied a usability evaluation [19] to determine and improve the usability of indexation tools used by Ariadne users.

In this paper, we introduce a user study conducted to evaluate the usability of search tools used in LORs. More specifically, we want to determine the usability of a search tool used by Ariadne users to locate relevant objects. This evaluation will help us to determine to what extent search tools enable users to reach their goals effectively and efficiently [2] [15]. In addition to this, it will help us to determine users' satisfaction on the overall use of the evaluated tools. Findings of this study are generalized for similar tools used in other LORs.

The paper is structured as follows: in section 2, a background is provided. In section 3, we discuss the research objectives, methods and materials. In sections 4, we provide findings and recommendations of the study. Discussion is illustrated in section 5. Conclusions and future directions are provided in section 6.

2 Background

The basic mission of ARIADNE [1] is to enable better quality learning through the development of learning objects, tools and methodologies that enable a “share and reuse” approach for education and training. A search and indexation tool (SILO) was developed to facilitate store new objects and search for relevant objects in the Knowledge Pool System (KPS). This KPS is a Learning Object Repository (LOR), like EdNa [10], SMETE [2324], and Merlot [16]. Ariadne users use SILO tool (see figure 1) to introduce new learning objects or search for relevant objects in the Ariadne KPS and other collaborated repositories as well. The SILO indexation functionalities are not relevant for this paper as they are discussed in [20][19].

Fig. 1. Screenshot of SILO Indexation client

Each repository aims at providing their users (teachers, students, content developers) with the learning objects they need to reuse for their different courses. This requires repositories to:

- Bear a vast collection of learning objects. As manual indexing (uploading the objects and adding descriptive metadata) of new objects is labor-intensive, searching other repositories using federated search (query objects in a set of collaborated repositories from within the same search tool) may lead to gathering a great set of appropriate learning objects to users. Nowadays, the five repositories Ariadne, Merlot, EdNa, and SMETE allow their users to access learning object in

the different repositories. Furthermore, semi-automatic or automatic indexation of learning objects can significantly increase the number of learning objects.

- Provide tools and different functions that allow finding appropriate learning objects. Different repositories use different functions (simple/advanced search and browse) and criterions to locate learning objects.

Repositories may rely on different metadata sets (profiles) to facilitate searching relevant learning objects, for example, Ariadne and Merlot uses a LOM [13] application profile while EdNa uses Dublin Core [8] profile. In this study, we want to evaluate the usability of the search tools used in those repositories.

3 Description of Research

In this study, we aim to evaluate and improve the usability of a search tool (SILO) used by Ariadne users to search learning objects in the KPS. We want empirically to answer the following questions:

- How do users use the search tool?
- How effectively and efficiently does the search tool help users perform their search tasks?
- What are the factors that may increase the performance of finding learning objects?
- Are users satisfied with the overall use of the tool?

In order to collect primary data on the usability of SILO and double check and validate our findings we conducted two iterative usability phases:

- **First Phase:**

We collected primary data from extended sessions with 16 participants to determine the initial usability of the tool.

- **Second Phase:**

Here, we evaluated the tool after solving the usability problems and integrating the recommendations that appeared in the first phase. In this phase, we collected data from extended sessions with 10 new participants who have no prior experienced with the tool.

The twenty six participants volunteered to be representative of the intended user community cover a range of disciplines including Arts, Computer Science, Electrical Engineering, Chemistry, Physics, Archeology, Architecture and Medicine.

We started each test session by introducing participants to the process. We also explained to them how much their feedback is important for us to improve the usability of the search tool. No instructions were given for SILO, this is because many users who may search learning objects probably are inexperienced with such tools.

During each test session, a participant was asked to use SILO to perform a number of pre-selected task scenarios (an example is provided in table 1) that are related to participants work context. Participants were asked to think aloud while carrying them out. At the end of each session of both phases, participants were asked to fill in a feedback questionnaire on the overall use of SILO. An experimenter was present in the test room throughout the session to provide any assistance solicited and to observe

participants behavior. In addition, screen recording was applied to observe user interaction during test sessions.

Table 1. Excerpt from the tasks assigned to participants in the usability test

Task1	Search for learning material on the topic “Business Process”.
Task2	Try to locate learning materials of Dutch language on the topic “Process Re-engineering”.
Task3	Find a questionnaire on the topic process re-engineering.
Task4	Find all learning objects that published by Aebi Marqaux.
Task

Screen recordings and paper notes were compiled, analyzed and findings were sent to the development teams in both phases for improvements.

In the next section, we discuss findings and recommendations of both usability phases.

4 Usability Evaluation

In section 4.1, we present the first usability phase conducted with sixteen participants to collect primary data on SILO usability. In section 4.2, we present the second usability phase applied to measure the usability of SILO after solving usability problems obtained from the first phase.

4.1 First Phase Findings and Recommendations

1) *Efficient simple search function is important to elevate users’ motivation and trust*

The simple search function is provided as the default service at the user interface. It works by matching terms provided by users and the stored metadata of data elements. All participants started their sessions (first task) of searching appropriate learning objects using the simple search function. Surprising, most participants were not able to find their appropriate objects through the simple search, although, they were asked to search for objects that are available in the KPS. We believe that participants were not able to find their appropriate objects because of the following reasons:

- Matching terms provided in the simple search box with stored metadata of the two elements “title” and “author” (elements used in the simple search function) is narrowing the possibility of finding appropriate objects.
- Simple search uses an exact match approach, which is also not an efficient function in the context of finding objects; users rarely provide exact keywords for their searches.

The simple search function of the evaluated search tool and other tools used in other repositories can be more efficient and smarter if we address the following issues:

- Simple search functions should search in all metadata elements that contain rich information about learning objects. The current search tools (in Ariadne and other repositories) search three to six metadata elements. That might be related to some

technical issues, like string comparison load. This issue can be enhanced using the power of database indexing techniques [7], which allow instantly search of many more metadata elements. We may also search elements that are mostly used by users when index or search for material [18]. For example, all simple search functions should search in values provided for elements like “main concept”, “main discipline”, “document format” (diagram, slides, simulation or video) and “description”, which are mostly used by users in different context like LORs [18], digital libraries [11].

- Simple search should not only be based on exact match between values provided for metadata elements and the terms entered by users. Unfortunately, the metadata information stored for each LO and terms provided for user queries often have many morphological variants. That leads to considering for example the two keywords “ontology” and “ontologies” as non equivalent terms. While, in the context of information retrieval both terms are equivalent. However, using some form of natural language processing (NLP), like stemming algorithms [23] may increase the performance of the simple search function.

The techniques mentioned above can noticeably enhance the usefulness of the simple search service.

2) Organization and structure of information should be adapted to users' needs

Advanced search function uses a set of metadata elements to enable the search of learning objects. Those elements are grouped into five relevant panels (see figure 1), this approach of relevance grouping (semantic, technical, pedagogical, etc.) was proposed to be usable by the end users [20]. For users to perform a successful search query, they need to fill-in or select values for elements that form their search.

Most participants (88%) found that the current advanced search is overwhelming and they preferred to have the advanced search elements at one panel.

One participant replied, “Why not to have the ‘main concept’ element in the first panel instead of the second, it should be some where else”.

Moreover, two third of the participants reported that the advanced search interface has too many metadata elements, they wanted to have very few fields to be presented in the advanced search. For example, they think that technical data elements like file name, operating system, indexation identifier are technical and should be hidden from users. That is a trade-off issue; other set of users (technical users) may prefer to be presented with the technical elements.

In addition, about two third of the participants, found that the current interface of advanced search makes some options unreachable. For example, one of the tasks was to locate a “UML diagram” that explains the concept “inheritance”. Most participants were not able to finish that task successfully, because participants were not able to select the value “diagram” from the “document format” element. Selecting that value means that searchers want only to locate diagrams not a video neither a narrative text.

Based on the observations and feedback questionnaire (see table 2), we found that the organization of the current advanced search does not match users' needs. The metadata elements form on the advanced search should be decreased and re-organized in one panel. In addition, it should be adapted to the users' needs and not to the used metadata standard (LOM). That can be achieved by presenting users with 2 to 4

elements that have rich metadata and mostly used for searching. Those most used elements can be different from one community of users to another.

3) Used terminology should be understandable by users

More than 60% of participants found that the terminology used for metadata elements and vocabularies at the advanced search function are confusing. That is due to the fact that the labels of elements as well as vocabularies at advanced search are adapted to the labels of the metadata standard and not to users' context and understanding. For example, the labels of the vocabularies (active and expositive) of the element "document type" were un-understandable for most users (the term active means learning objects that includes interaction between the end user and the object, like questioners or problem statements. While the term expositive means objects like an audio file or a diagram, which have no interaction with the user). Also, labels for the elements "semantic density" (the amount of information conveyed by the learning object), "header author" (author of metadata and not the learning objects itself) were not understood by most participants.

Questionnaire feedback (see table 2) and participants observation shows that the use of terminology in the search interface should be improved. This can be achieved by presenting those elements and vocabularies in a terminology that is understandable by the users, not based on the labels provided by the standard. Giving those elements and vocabularies labels that are understandable by users and different from how they are represented in the metadata standard, does not mean that the LOR metadata are not relying on the standard. It would be recommended that labels of the data-elements and its associated vocabularies be adapted to better align with a more familiar terminology used by users and then mapped to the appropriate elements in the standard at the technical level.

4) Help and feedback should be improved to improve users' performance

About half of the participants found that SILO should provide users with more purposely help and feedback whenever needed. Participants asked for help hints for the meaning of some terminologies. They found that more feedback messages should be provided to users when perform a wrong action, when the object is not found or restricted (e.g., copyright, author permission, etc.).

Based on participants observation and feedback questionnaires (see table 2) we believe that users should be provided with help messages and hints that guide them to improve their search performance. That can be achieved by providing users with some simple hints for the meaning of data elements or vocabularies wherever needed. In addition to that, guidance help messages that may guide users find their objects should be provided. For example, we may present users with the following message when the learning object is not for free of use: "in order to use this learning object you should contact the author of the object", also, we should provide that user with the contact information of that author in question. That kind of guidance messages can greatly enhance the overall user satisfaction.

5) The ability to refine search terms enhances search performance

The majority of participants disliked the fact that SILO does not provide the facility to reformulate previous queries. The search interface should enable the users to

reformulate their previous queries. Especially, in cases when users want to change or remove one or two letters to submit the new query or to narrow down the search within the given results (like in Google).

Overall Satisfaction

Table 2 presents the participants response to questions concerning the overall use of SILO search tool. The popular attitude scale with seven points (ranging from 1- poor to 7- good) was used to measure the response of participants on the overall use of SILO.

Table 2. User satisfaction on the overall evaluation

	1. Ease of use	2. nformation organization	3. Use of terminology	4. Quality of feedback	5. Navigation	6. Search and download	7. Results List	Overall Mean
Mean	3.6	3.7	3.3	3.8	5	4.8	5.1	4.2
St. Div.	1.3	1.5	1.4	1.9	1.5	1.9	1.4	

The mean for the level of ease-of-use was less than 4, which means that the participants found SILO rather difficult to use. The level of information organization, use of terminology and quality of feedback was perceived as low (mean 3.7, 3.3 and 3.8 respectively). We believe that this is related to the fact that the organization and naming of metadata elements and vocabularies are more adapted to the used metadata standard (LOM) and not to the user needs. Navigation between the panels and the readability of the results list were seen as acceptable (means 5 and 5.1).

As can be seen from the table above, the level for the overall use of SILO was perceived as moderate (mean 4.2). We think that SILO usability should be improved, in order to increase the performance of finding appropriate learning objects.

In the next section, we present findings of the second phase of the usability test aimed to evaluate SILO after integrating usability recommendation of the first phase.

4.2 Second Phase Findings

The aim of this phase is to determine the usability of SILO after removing the sophistications and usability problems that decreased participants' performance in the first phase. Also, to discover new findings and usability problems that we could not find in the first phase.

Ten new participants (inexperienced with the tool) were asked to test the tool after fixing the usability problems that appeared in the first phase. The aim here is to measure SILO usability after solving some of the identified problems. The following improvements were integrated in the new SILO:

- The current simple search now queries data in more metadata elements (author, title, and concept) to increase the number of retrieved learning objects. The main concept element mostly contains the main key words on the learning object.
- The advanced search function presents users with only four fields of metadata elements (Title, Author, Usage rights and Concept). Users can also extend their queries with more fields if needed, by clicking on a button captioned with “more” (see figure 2).
- Users can refine their search terms provided at the advanced search functions.
- A federated search function was added to allow users search for appropriate objects in more LORs like Merlot, SMETE and EdNa.
- More purposely help notes and feedback messages were provided to users whenever needed.

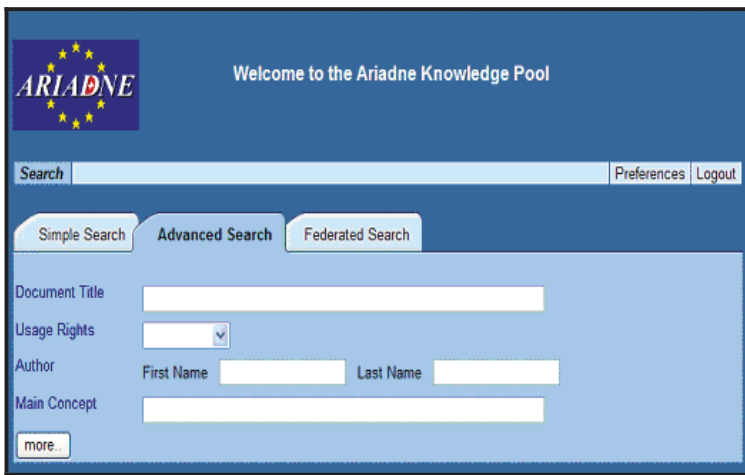


Fig. 2. Screenshot of the new SILO

Table 3 presents the participants’ response to questionnaire questions on the overall use of SILO after solving the usability problems obtained from the first evaluation phase.

Table 3. User satisfaction on SILO in the second phase

	1. Ease of use	2. Information organization	3. Use of terminology	4. Quality of feedback	5. Navigation	6. Search and download	7. Results List	Overall Mean
Mean (second phase)	5.5	5	5	5.1	6.3	6.3	6.1	5.6
Mean (first phase)	3.6	3.7	3.3	3.8	5	4.8	5.1	4.2

The mean for level of ease-of-use was noticeably increased to a good level (from 3.6 to 5.5). That means participants found SILO rather easier to use after solving the major usability problem obtained from the first phase. The organization of information, use of terminology, quality of feedback were also noticeably increased to a good level. Navigation, readability of the results list and download of objects were perceived as high. In general, the level of the overall use of SILO was perceived as good.

We drew a comparison between the overall use of SILO before and after identifying the usability problems. Five participants (group 2) who participated in phase two were asked to use the old interface of SILO. Moreover, we asked another five participants (group 1) who participated in the first phase to evaluate SILO interface of the second phase (SILO 2). That is to revalidate the recommendations and usability problems obtained from the first evaluation phase and to draw some comparisons between the two interfaces. Based on participants' feedback, we found that SILO 2 was much easier to use and much less overwhelming than SILO for both of the groups (see figure 3).

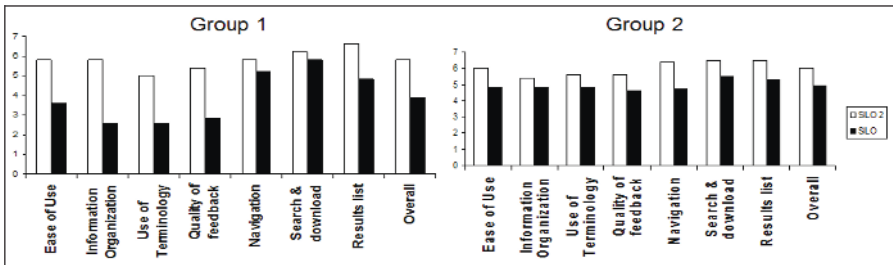


Fig. 3. Comparison between responses of participants who evaluated SILO 2 then SILO (group 2), and participants who evaluated SILO then SILO 2 (group 1)

As shown in figure 3, in both groups 1 and 2 the overall usability of SILO 2 is higher than SILO. In addition, the level of user satisfaction on all the studied factors (ease of use, use of terminology, information organization, etc.) of SILO 2 is higher in both groups.

5 Discussion

As discussed in the previous section, finding appropriate learning objects is still not an easy task. The usability of the search interface may noticeably decrease the performance of users searching for relevant material. The use of terminology and structure of information in the old SILO was more adapted to the metadata standard than to the user needs. That practice of metadata use can be noticed in the search interface of other existing repositories such as Merlot and SMETE (see figures 4 and 5). Duval, E. [9] called that practice a metadata myth that should be killed. The labels used for metadata elements and the way they are structured in the standard are intended to provide guidance to tool developers and need to be replaced by words

meaningful to the end users and should be structured/presented according to the needs of the local community.

The main goal of a search tool is to facilitate the finding of appropriate learning objects. We believe that this goal is a great challenge facing the different LORs. As illustrated in the previous sections form-based search tools are not easy to use and most of their components (metadata elements and the associated vocabularies) are not familiar and may not be used by most users. Users tend to use few terms to look for appropriate materials. In order to provide the user with a more usable and interactive methods for finding appropriate objects, the following issues should be addressed:

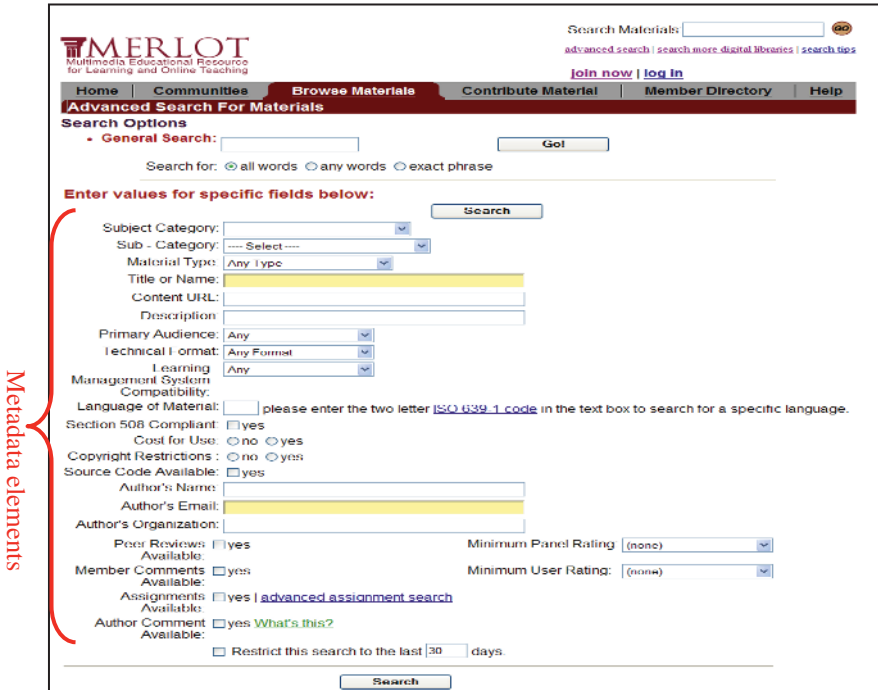


Fig. 4. Screenshot of the search interface of Merlot

- Simple search functions should be smart enough, and should compare the keywords provided in the text box with as many metadata elements as possible. In particular, elements that are often used by indexers when introducing learning objects to such repositories. The existing simple search functions of all mentioned LORs does not match the terms provided by the user in the search box with the values provided for the “document format” element. Therefore, when users provide the keywords “inheritance diagram”, they will not be able to locate the diagram object that were indexed with the title “inheritance”.
- Advanced search interfaces should not contain too many metadata elements. Merlot advanced search presents users with 16 elements and SMETE presents 8 elements (see figures 4 and 5). Users should only be presented with two or

maximum four most used elements. Advanced search interfaces that require the user to navigate between different menus and provide values for elements placed in long lists, decrease users' search performance and do not guarantee finding the appropriate objects. Metadata and vocabulary labels should be understandable by users, some help notes should be provided for unobvious metadata elements. As shown in figure 5, SMETE advanced search provide help notes for all data elements. We recommend to provide help notes for unobvious elements only.

- Ranking mechanism: results ranking has a major impact on users' satisfaction with search engines and their success in retrieving relevant documents [6]. The major existing repositories sort the retrieved results based on the author, title or metadata creation date of the objects. They do not rank retrieved objects according to their relevance to users search queries. Retrieved results list should place objects that best match the search query at the top ten of the results list. Such ranking mechanisms for LORs search tools can enhance users' performance in retrieving relevant objects.
- More novel access techniques such as information visualization [14] and recommender systems should be considered to improve finding appropriate objects.

The screenshot displays the SMETE search interface. At the top, there is a navigation bar with links for FIND, BROWSE, WORKSPACE, ABOUT SMETE, and HELP. Below this is a search bar with a GO button. The main content area is titled "Find Learning Resources" and includes instructions on how to use the search fields. A sidebar on the left, titled "Find More Resources", contains links for "Find SMETE Learning Resources", "Find Partner Collection Learning Resources", and "Related Terms Not Available". A red bracket on the left side of the search form groups the fields under the label "Metadata elements". The search form includes the following fields: Keyword(s), Type of Learning Resource, Grade, Title, Author / Creator, Collection, Publication Year (with After and Before input boxes), and Peer Reviewed. There are Search and Reset buttons at the bottom of the form.

Fig. 5. Screenshot of the search interface of SMETE

6 Conclusions

In 1986, Borgman discovered that Online Public Access Catalogs (OPAC's) are difficult to use, because their design does not match searching behavior [3]. In 1996, in a follow-up study, Borgman concluded that OPAC's were still difficult to use [4]. In this paper, we conclude that, at least for Learning Object Repositories, the situation has not improved substantially. (We would argue that the same observation holds for many digital libraries, but that was not the focus of the study reported here.)

More specifically, we conclude that the Ariadne search tool is hard to use, because its interface reflects the metadata standard rather than the characteristics, aims and requirements of the end user. In the paper, we make specific recommendations to improve the usability of the Ariadne search tool and we generalize our recommendations for other learning object repositories.

More generally, we once more want to emphasize (as others have done before us) that the outcomes of such user studies are vital to improve the design of search tools that can better serve the needs of end users.

References

1. ARIADNE. <http://www.ariadne-eu.org>
2. Blandford, A., Suzette, K., Connell, I., Edwards, H., Analytical usability evaluation for digital libraries: a case study, ACM/IEEE Joint conference on Digital libraries, 2004, pp. 27-36.
3. Borgman, C. L. Why are online catalogs hard to use? Lessons learned from information retrieval studies. *Journal of the American Society for information Science*, 37(6). 387-400, 1986.
4. Borgman, C. L.. Why are online catalogs still hard to use? *Journal of the American Society for information Science*, 47(7), 493-503, 1996.
5. Cockburn, A., McKenzie, B., What do web users do? An empirical Analysis of Web Use, *Int'l. Journal of Human-Computer Studies*, 54(6):903-922. 2001.
6. Courtois, M. P., & Berry, M. W. Results-ranking in Web search engines. Online, 1999, 23(3), 39-40.
7. dtSearch, How to index databases with the dtSearch Engine. <http://support.dtsearch.com/faq/dts0111.htm>.
8. Dublin Core, Dublin Core Metadata Element Set v1.1. <http://www.dublincore.org>
9. Duval, E., Hodgins, W., A LOM research agenda. International conference on World Wide Web, 2004.
10. EdNa. <http://www.edna.edu.au/>
11. France, R. K., Nowell, T. L., Fox, A. E., Saad, A. R., Zhao, J., Use and usability in a digital library search system, CoRR cs.DL/9902013, 1999.
12. Jones, S., Cunningham, S. J., McNab, R., An Analysis of Usage of a Digital Library, ECDL 1998, pp 261-277.
13. IEEE Standard for Learning Object Metadata. <http://ltsc.ieee.org/doc/wg12/>.
14. Klerkx, J., Duval, E., Meire, M., Using information visualization for accessing learning object repositories, Information Visualization, 2004 (IV04), pp. 465-470.

15. Marchionini, G., Plaisant, C., Komlodi, A., The people in digital libraries: Multifaceted approaches to assessing needs and impact. In: A. Bishop, B. Bittenfield, & N. VanHouse (Eds.) *Digital Library Use: Social Practice in Design and Evaluation*. MIT Press. November 2003. pp. 119-160.
16. MERLOT. <http://www.merlot.org/>.
17. Najjar, J., Ternier, S., Duval, E., the Actual Use of Metadata in ARIADNE: An Empirical Analysis, ARIADNE 3rd Conférence, 2003, pp. 1-6.
18. Najjar, J., Ternier, S., Duval, E. User Behavior in Learning Objects Repositories: An Empirical Analysis. *EdMedia*, 2004, pp. 4373-4378.
19. Najjar, J., Klerkx, J., Ternier, S., Verbert, K., Meire, M., Duval, E., Usability Evaluation of Learning Object Indexation: the ARIADNE Experience, *European Conference on e-Learning*, 2004, pp. 281-290.
20. Neven, F., Duval, E., Ternier, S., Cardinaels, K., Vandepitte, P. An Open and Flexible Indexation and Query tool for ARIADNE, *EdMedia 2003*, pp. 107-114.
21. Nielsen, J., *When Search Engines Become Answer Engines*, *usit.com*, 2004.
22. O'Neill, C., Paice, C. D., the lancaster stemming algorithm. <http://www.comp.lancs.ac.uk/computing/research/stemming/>.
23. Porter, M., *The Porter stemming algorithm*. <http://www.hackdiary.com/>.
24. SMETE. <http://www.smete.org/>.

Managing Geography Learning Objects Using Personalized Project Spaces in G-Portal

Dion Hoe-Lian Goh², Aixin Sun^{3,*}, Wenbo Zong¹, Dan Wu¹, Ee-Peng Lim¹,
Yin-Leng Theng², John Hedberg³, and Chew Hung Chang³

¹ Center for Advanced Information Systems, School of Computer Engineering,
Nanyang Technological University, Nanyang Avenue, Singapore 639798
{zong0001, 147667409, aseplim}@ntu.edu.sg

² School of Communication & Information, Nanyang Technological University,
31 Nanyang Link, Singapore 637718
{ashlgoh, tyltheng}@ntu.edu.sg

³ Center for Research in Pedagogy and Practice, National Institute of Education,
1 Nanyang Walk, Singapore 637616
{jhedberg, chchang}@nie.edu.sg

Abstract. The personalized project space is an important feature in G-Portal that supports individual and group learning activities. Within such a space, its owner can create, delete, and organize metadata referencing learning objects on the Web. Browsing and querying are among the functions provided to access the metadata. In addition, new schemas can be added to accommodate metadata of diverse attribute sets. Users can also easily share metadata across different projects using a “copy-and-paste” approach. Finally, a viewer to support offline viewing of personalized project content is also provided.

1 Introduction

Digital libraries (DLs) are no longer static repositories of information in which access is limited to searching and browsing. As the amount of digital content grows, there is increasing recognition that DLs will play important roles in education, research and work. Correspondingly, access mechanisms in DLs have also become richer, providing a greater array of services for users to interact and manipulate content including annotations, user contributions and workspaces [3, 6].

Such services are especially needed in educational digital libraries that are typically designed to support classroom learning. Here, the mode of learning no longer adopts a traditional approach in which the teacher fills students with deposits of information deemed to be knowledge and the students store these pieces of information intact until needed [8]. Instead, a constructivist approach is usually taken in learning activities that are characterized by active engagement, problem-solving, inquiry, and collaboration with others so that each individual constructs meaning and hence knowledge of the information gained.

* Aixin Sun is currently with School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia.

Consider a class of students attending a course in a university or high school. The students are required to complete a project that involves the acquisition of course content from the teacher, gathering of other reference materials from the library or other sources, making sense of all the available information, synthesizing content, writing the project report and submitting the completed project for grading. Throughout these activities, the role of traditional DLs is usually confined to the provision of referenced materials. While access to library collections has been made much simpler by Web-based interfaces and digitized content, there is still a significant gap to bridge when DLs are to be included as part of learning process. Crucial to the design of such DLs is the need for an integrated work environment where users may store personal or group information objects relevant to a particular task at hand [7].

In this paper, we describe G-Portal [4], a digital library of geospatial and georeferenced resources. We focus on the system's support for personalized learning activities. Known as the personalized project space, it is a key feature that allows its owner, the learner, to gather and organize metadata about geography-related learning objects relevant to his/her learning goals. Our previous work on G-Portal has dealt with the design and implementation of a public portal space and features to manage geography learning objects on the Web. Querying, classification, annotation and map browsing are among the functions provided to manipulate these objects. These functions are now made available in the personalized project space. In addition, functions to support the acquisition of new metadata content, synthesizing knowledge relevant to a topic, exporting of personalized project content and creating (and referencing) project contexts have been implemented to make G-Portal a full-featured integrated learning environment for geography and earth system science.

2 An Overview of G-Portal

G-Portal is a Web-based digital library of geospatial and georeferenced resources and provides a variety of services to access and manage them [4]. The resources maintained by G-Portal comprise mainly metadata records that describe the actual resources, such as Web pages, images and other objects that are accessible on the Web. Other types of information managed by G-Portal include semi-structured data records and annotations.

Each resource contains among other attributes, a location attribute (if available) storing its geospatial shape and position, and a link to the corresponding actual resource. G-Portal provides a map-based interface that visualizes resources with location attributes on a map (see Figure 1). This interface makes resources with known geographical locations easily and intuitively accessible and helps users discover the spatial relationships between resources. For resources without a location attribute, G-Portal provides a classification-based interface that organizes resources based on a customizable taxonomy. A query interface that supports searches for resources based on keywords and spatial operators is also available.

G-Portal organizes resources into projects in which each project contains a collection of resources that are relevant to a specific topic or learning activity. Within each project, resources are further grouped into layers for finer grained organization. Each

layer serves as a category to store logically related resources. For example, a project studying flora and fauna in nature trails may include rivers, lakes and hills in a map layer, flora and fauna information in another, and annotations in a separate layer.

G-Portal is developed as a Java applet with all projects, layers and resources stored within a database server that supports XML and spatial operations. G-Portal can therefore be accessed from any Java-enabled Web browser, making it possible for users to easily access and manipulate personalized project space anywhere, anytime.

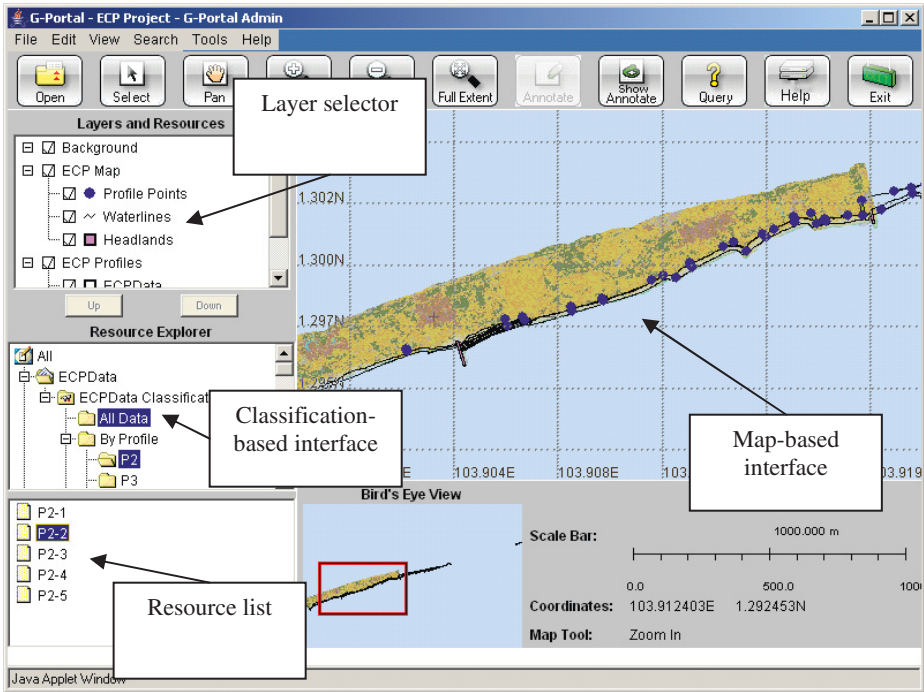


Fig. 1. The G-Portal interface

3 Personalized Project Spaces

While G-Portal's existing project feature appears to be a good place to manage objects relevant to a learning activity, there are a few limitations. Firstly, these are public projects and are not user customizable, but are created by administrators to serve the general needs of all G-Portal users. As such, they do not cater towards supporting individual or group learning. In addition, no provision is made for carving out projects that only a certain user or group can access. The personalized project space was therefore developed to address these issues so as to better support learning activities with G-Portal.

3.1 Project Configuration

A personalized project is provided to each user (or group of users) to allow the creation of personal collections of resources and annotations relevant to a particular learning activity or need. A personalized project has the same basic features as any project in G-Portal but only its owner can modify the contents of the project. When a personalized project is first created, new layers have to be created to organize metadata records. G-Portal therefore provides features to create/delete layers.

Once layers are created, metadata are organized according to the needs of the user by assigning them to appropriate layers. In addition, metadata can be assigned spatial locations and be displayed in the map-based user interface under different layers. For example, metadata of buildings and roads can be displayed in one layer while that of parks and lakes can form another layer. Metadata can also be organized under one or more category hierarchies and made viewable via the classification-based interface.

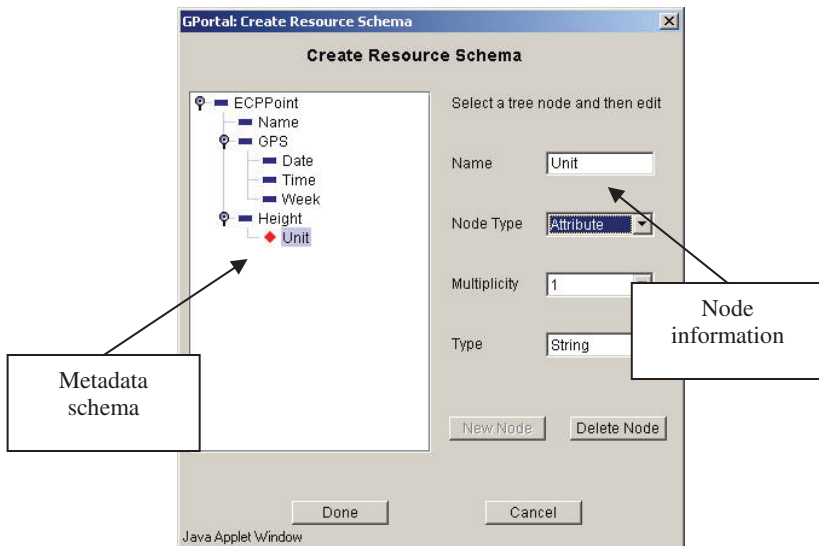


Fig. 2. Metadata schema creation

3.2 Metadata Creation

Every resource in G-Portal is created using a resource schema that serves as a template for describing resources. In a personalized project, schemas can be either predefined or user-defined to meet the needs of a learning activity for a user (or group of users). As shown in Figure 2, G-Portal provides a facility to create new metadata schemas if the required schemas are not defined yet. Each schema is represented as a tree structure with nodes representing metadata elements and multiplicity constraints. New metadata schemas are required when users wish to add metadata of new formats into their personalized projects. Due to its personal usage nature, these schemas are not subject to a formal review and registration process.

Once schemas are defined, users can create new metadata records (resources) using G-Portal's built-in metadata editor (see Figure 3). Like schemas, metadata records are represented as a tree structure. Creating a new metadata record involves specifying the metadata schema to use causing the editor to ensure that only valid metadata can be created. Users then complete the metadata fields with relevant content.

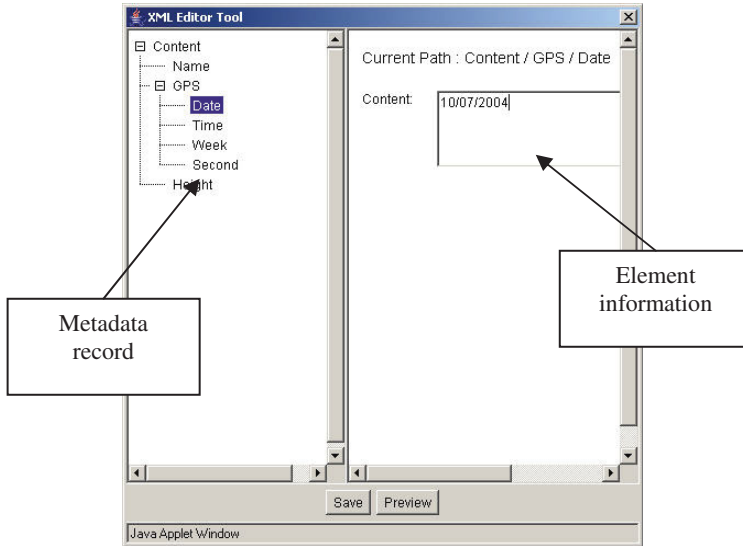


Fig. 3. Metadata creation

3.3 Sharing Metadata Records

Resources in public projects are often useful for reference within a personalized project. For example, a teacher might maintain a public project containing information for an entire semester's course in geography. A student working on a particular learning activity may need to include existing metadata records in the public project into his/her own personalized project. To facilitate easy transfer of these metadata records, a copy-and-paste function is supported. With this facility, users can simply select metadata records of interest from one or more public or group accessible projects, copy and then paste them into their personalized projects. No records are duplicated in this process and G-Portal simply provides links to them. These operations are similar to existing copy-and-paste functions in desktop applications. Through this copy-and-paste process, metadata records can therefore be shared across projects.

4 The G-Portal Project Viewer

There may be occasions when Internet connections are not available but access to a project's resources is necessary. Such a situation may occur when, for example, a student is on a field trip to a remote area, or when the project resources are to be referenced by someone who has no access to the G-Portal server. G-Portal therefore

supports offline viewing of project content through a desktop viewer application. An export function is also developed to export projects to local storage to complement the offline viewer.

4.1 Exporting Projects

Prior to offline viewing, the project, be it public, group or individually accessible, has to be exported to local storage. This is accomplished through a feature in the application version of G-Portal, namely, the G-Portal Viewer. Here, the exporting operation requires the G-Portal Viewer to be connected to Internet. The entire project including all resources and resource organization information (layers and schemas) are packaged into a project file and saved to the user’s computer. When invoked, the G-Portal Viewer then accesses the package and displays the personalized project offline. The Viewer is very much similar to the G-Portal applet except that functions for modifying project content are disabled to protect the integrity of project data stored in the G-Portal server.

4.2 Context Creation and Referencing

G-Portal provides a host of navigation facilities for exploring content in the map-based and classification-based interfaces. In large projects, extensive navigation might lead to users being lost in “project-space”, a phenomenon not unlike that found in hypertext navigation.

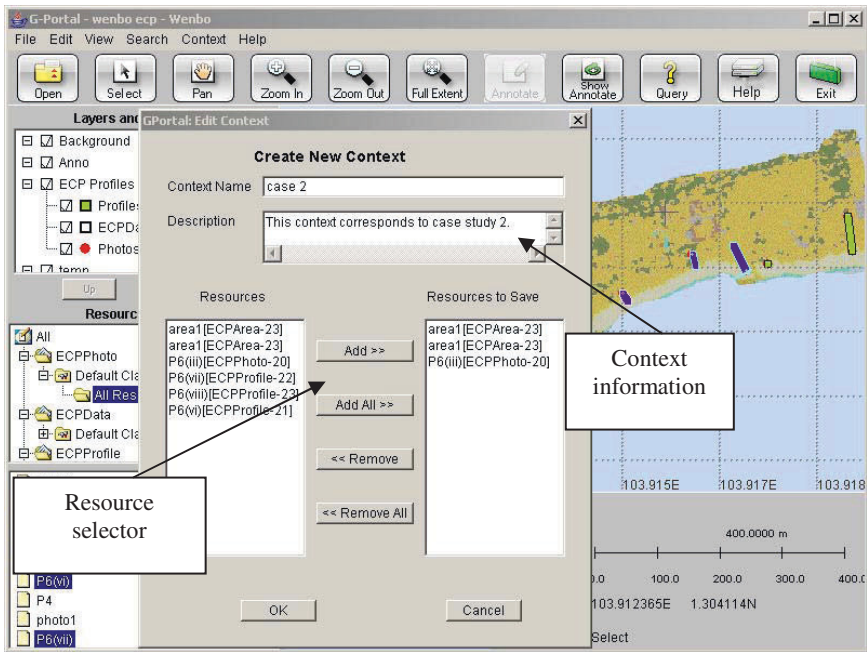


Fig. 4. Saving a context

The context is thus another important feature in the G-Portal Viewer. A context refers to the registration of a map extent (a specific map area) and its associated meta-data records for future reference and is similar to the idea of bookmarking. Figure 4 shows the dialog for saving a context. Users specify the context's unique name and description as well as the resources to be saved.

Referencing contexts can be done in two ways. The first is through the G-Portal Viewer in which a user can select from a list of previously registered contexts to visit. The second approach is to launch the G-Portal viewer with a specified context by embedding a link to the context in a document, such as a Microsoft Word file or HTML file. This approach is particularly useful in that it provides seamless access to personalized projects from other applications. An example of its use would be for a student to complete a project using a certain application (e.g. Microsoft Word) but make references to information found in a G-Portal project. Teachers, graders and other students can then view the report as well as the referenced resources using the G-Portal Viewer.

5 Related Work

G-Portal shares similar goals with existing digital libraries such as ADEPT [1], DLESE [9] and CYCLADES [5]. ADEPT supports the creation of personalized digital libraries of geospatial information ("learning spaces") but owns its resources unlike in G-Portal where the development of the collection depends mainly on users' contributions as well as on the discovery and acquisition of external resources (such as geography-related Web sites). Our model is similar to DLESE although the latter does not support an interactive map-based interface or an environment for online learning. CYCLADES provides a suite of tools for personalizing information access and collaboration but is not targeted towards education or the uniqueness of accessing and manipulating geospatial and georeferenced content.

6 Conclusion

Digital libraries are beginning to play key roles in education. The G-Portal digital library represents a step in this direction and provides a suite of tools for supporting geography and earth system science education. In the paper, we have described the concept of personalized spaces in G-Portal and showed how its features can be used to support learning activities. With users having the flexibility to create and manipulate personalized content, we believe that a better integration between digital library content and learning activities can be achieved.

As part of ongoing work, we are evaluating the use of G-Portal in various field study projects. An initial qualitative study with undergraduate students using G-Portal to investigate a stretch of the local coast and solve geographical problems related to erosion was conducted. Results suggest that G-Portal was useful in supporting information discovery and learning [2]. Future work would involve larger cross-sectional studies with more students and teachers, working on more tasks, as well as longitudinal studies that track the learning experiences of students as they use the system.

Acknowledgements. This work is partially funded by the Center for Research in Pedagogy and Practice, National Institute of Education through Project No. CRP 40/03 LEP.

References

1. A. Coleman, T. Smith, O. Buchel, and R. Mayer. Learning spaces in digital libraries. In P. Constantopoulos and I.T. Sølvsberg (eds.): *Proceedings of the Fifth European Conference on Research and Advanced Technology for Digital Libraries. Lecture Notes in Computer Science, Vol. 2163.* Springer-Verlag, Heidelberg (2001) 251-262.
2. C.H. Chang, J.G. Hedberg, Y.L. Theng, E.P. Lim, T.S. Teh, and D.H. Goh. Evaluating G-Portal for geography learning and teaching. In *Proceedings of the 2005 ACM+IEEE Joint Conference on Digital Libraries (June 2005)* in press.
3. D. Goh, L. Fu, S. Foo. A work environment for a digital library of historical resources. In E.P. Lim et al (eds.): *Proceedings of the 5th International Conference on Asian Digital Libraries. Lecture Notes in Computer Science, Vol. 2555.* Springer-Verlag, Heidelberg (2002) 260-261.
4. E.P. Lim, D.H. Goh, Z.H. Liu, W.K. Ng, C.S.K. Khoo, S.E. Higgins. G-Portal: A map-based digital library for distributed geospatial and georeferenced resources. *Proceedings of the Second ACM+IEEE Joint Conference on Digital Libraries (July 2002)* 351-358.
5. H. Avancini and U. Straccia. Personalization, collaboration, and recommendation in the digital library environment CYCLADES. In *Proceedings of IADIS Conference on Applied Computing (March 2004)* 67-74.
6. M. Agosti, N. Ferro, I. Frommholz, U. Thiel. Annotations in digital libraries and laboratories – facets, models and usage. In R. Heery, L. Lyon (eds.): *Proceedings of the 8th European Conference on Research and Advanced Technology for Digital Libraries. Lecture Notes in Computer Science, Vol. 3232.* Springer-Verlag, Heidelberg (2004) 244-255.
7. M.E. Renda, U. Straccia. A personalized collaborative digital library environment: A model and an application. *Information Processing and Management* 41 (2005) 5-21.
8. P. Oldfather, S. Bonds, T. Bray. Drawing the circle: Collaborative mind mapping as a process for developing a constructivist teacher education program. *Teacher Education Quarterly* 21 (1994) 5-13.
9. T. Sumner and M. Dawe. Looking at digital library usability from a reuse perspective. In *Proceedings of the First ACM+IEEE Joint Conference on Digital Libraries (June 2001)* 416-425.

Evaluation of the NSDL and Google for Obtaining Pedagogical Resources

Frank McCown, Johan Bollen, and Michael L. Nelson

Old Dominion University, Computer Science Department,
Norfolk, VA 23529 USA

{fmccown, jbollen, mln}@cs.odu.edu

Abstract. We describe an experiment that measures the pedagogical usefulness of the results returned by the National Science Digital Library (NSDL) and Google. Eleven public school teachers from the state of Virginia (USA) were used to evaluate a set of 38 search terms and search results based on the Standards of Learning (SOL) for Virginia Public Schools. Evaluations of search results were obtained from the NSDL (572 evaluations) and Google (650 evaluations). In our experiments, teachers ranked the links returned by Google as more relevant to the SOL than the links returned by the NSDL. Furthermore, Google's ranking of educational material also showed some correlation with expert judgments.

1 Introduction

When the question “What is the most venomous snake in the world?” was posted to AskNSDL, the response was, “I did a search in google.com for ‘poisonous snakes world’ and came up with some good information...” [2]. A search for the term ‘google’ at ask.nsd.org reveals that Google is frequently used to answer AskNSDL questions. Why are the questions posed to an NSDL related web site answered using Google instead of the NSDL? The NSDL only accepts educationally sound material into its library [17], so naturally it should produce more trusted results than would Google, which accepts any web-crawlable content.

The National Science Digital Library (NSDL) [24] is a U.S. government funded online library that provides pedagogical resources for science, technology, engineering, and mathematics. In an attempt to provide only relevant and highly educational material, the NSDL obtains its contents from on-line material that is submitted by the education community or from focused crawls [4,5] or from specific, targeted collections [1,14]. Because all NSDL content is available on-line, some of the same material is also indexed by Google. The number of resources incorporated within the NSDL is relatively small when compared to Google because the NSDL acquires content solely for the purpose of supporting science education. Figure 1 illustrates a theoretical view of all the useful educational content that is on the web and how it has been crawled by Google and/or federated into the NSDL.

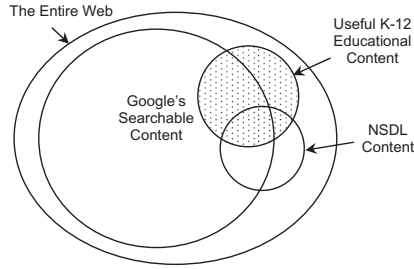


Fig. 1. Educational content on the Web

We wanted to measure the degree to which Google and the NSDL could locate relevant educational material. To do so, we designed a real-world scenario that showed how an educator might use the NSDL and Google to find educational content. This scenario provided an appropriate set of search terms that could be used to query the NSDL and Google. The Commonwealth of Virginia uses the state-wide standard called the Standards of Learning (SOL) for Virginia Public Schools [20] that mandates what should be taught in the public schools to children in grades K-12 (ages 5-18). The SOL lists detailed and specific topics which educators are required to teach and test for at each grade level. These topics can be researched using the web to retrieve background or additional information. Using the SOL for mathematics and the sciences, we devised a set of search terms and submitted them to the NSDL and Google. We used paid volunteer Virginia teachers to evaluate the search results in terms of their pedagogical usefulness. The results of the evaluation show that Google is able to produce more relevant educational material in its top 10 results than the NSDL.

2 Related Work

When evaluating web search engines, focus may be given to any number of factors: user interface, user effort, query response time, retrieval effectiveness, etc. Given that the primary goal of search engines is to locate relevant documents quickly, the most common measurements are precision and recall [6]. Precision, the ratio of retrieved relevant documents to the number of retrieved documents, is the primary measurement used by many web search engine comparisons as their primary evaluation measure [6,8,9,11,13,15,16,21]. Recall is the ratio of retrieved documents to the total number of relevant documents available. Because the total number of relevant documents in a collection is rarely known, recall is much more difficult to measure due to the huge number of returned search results produced by search engines, and has therefore been largely ignored by web search engine comparisons.

When evaluating precision, most studies use human evaluators who issue queries about some topic, browse the returned result set, and record a relevancy score for each link. What qualifies as “relevant” is often specific to the

study. Some studies have used a binary view of relevance [15] where search results are deemed either relevant or not relevant. Others have used a non-binary measurement [8,15,16] allowing a document to be semi-relevant. We have taken the later approach as will be discussed in Sect. 3. In all of these studies, evaluators were limited to viewing the first n returned results. Values of 10 [8,16] and 20 [6,11,13,15,22] were common. This prevented the evaluators from being overwhelmed by the volume of search results. This methodology is in keeping with studies that show most users do not go beyond the first screen of search results anyway [18,19]. Once the relevance scores were obtained for the first n results, a precision at n ($P@n$) was computed and used in statistical analysis. We also limited our evaluation to the first 10 search results from Google and the NSDL.

The difficulty of measuring precision is two-fold: the expense of time-consuming human evaluations, and the reliance on subjective human judgments as to the relevance of a resource. Some methods have been developed to compare search engines that remove the subjective human component from the process [3,6,7]. Although these methods have shown some promise, they are still somewhat new and need to be further validated. These automatic evaluation procedures could be applied to our findings in future research in order to further validate their methods. It is unclear though whether these approaches could be applied based on our definition of relevance as material that is educationally useful.

None of the studies in the literature have been performed against the NSDL. Furthermore, none of the studies addressed the educational usefulness of the search results. An in-depth look at how educators judged the quality of educational material on the web and in digital libraries was performed in [23]. They examined the mindset used by educators when deciding if a web resource was useful for augmenting the classroom experience. Participants of their study showed that educators expect educational digital libraries to save them time and effort when compared to web search engines because DLs filter their content. In the framework of our intended comparison of the NSDL and Google, it was thus expected that the NSDL would contain a collection of educational documents that were better filtered for quality and educational usefulness than Google and that these documents would be rated higher by educators. Although our study did not gauge the worthiness of using the resources that were found by the NSDL and Google in the classroom, several of our evaluators reported that many of the items they evaluated would make excellent classroom resources.

3 Experiment

We examined the SOL for math and scientific educational content that students would need to learn. In examining each standard, we came up with a list of search terms that a student or educator might type into the search facility of the NSDL or Google in order to find out more information about the topic.

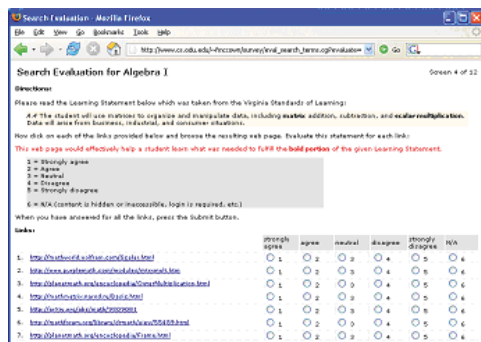


Fig. 2. Randomized and anonymous links in the evaluation system

For example, a teacher might select the terms we have bolded in the following Biology SOL:

BIO.3 The student will investigate and understand the chemical and biochemical principles essential for life. Key concepts include a) **water chemistry** and its impact on **life processes**; b) the structure and function of macromolecules; c) the nature of enzymes; and d) the capture, storage, transformation, and flow of energy through the processes of photosynthesis and respiration.

We paid 11 Virginia public school teachers to judge the quality of the search terms that we devised based on the SOL. The teachers were assigned according to their specialty. In other words, biology teachers were assigned to judge the quality of search terms from the Biology SOL, chemistry teachers judged search terms from the Chemistry SOL, etc. The evaluators used a web-based evaluation system that we developed to perform their evaluations. A screen shot of the system is shown in Fig. 2. After rating each search term, the evaluators were then presented a randomized list of at most 20 search results, combining the top 10 search results from the NSDL and Google. The evaluators were unaware of the source of the search results or rankings as Fig. 2 illustrates. Each search result was viewed by the evaluator, and then a score from 1 to 6 was assigned based on the quality of the search result. The score was based on how well the search result provided educational content (specific to the SOL) that was useful for student education. One was the best score, five was the worst score, and six indicated the web content was inaccessible (due to an old link, hidden content, login was required, etc.).

Each teacher evaluated at least 6 sets of search terms and search results. Each evaluation took about 1.5 hours to complete. The evaluators used common graphical web browsers like Internet Explorer and Netscape to perform the evaluations. The computers used to perform the evaluations had high-speed Internet connections so that our evaluators could work quickly and would not be prone to giving poor ratings to items that downloaded slower. The computers

were capable of displaying Adobe Acrobat files (PDF) and Microsoft PowerPoint files (PPT). PDF and PPT files made up 10% of the search results returned by Google.

The NSDL and Google queries were performed between Dec. 29, 2004 and Jan. 2, 2005. The evaluations (and web site accesses) were performed between Jan. 3 and Jan. 20, 2005. Our evaluation did not use the advanced search features of the NSDL and Google because research has shown that users rarely use advanced features [19]. Besides that, the NSDL lacks many of the advanced search features of Google. The NSDL does allow the user to limit the results to text, images, audio, video, interactive controls, or data, but it does not support boolean operators. In order to perform a fair evaluation of Google and the NSDL, we followed the guidelines suggested by [11,13].

We used a Likert 5-point scale to determine relevancy. For each search result returned, the evaluators gave a score based on “This web page would effectively help a student learn what was needed to fulfill the bold portion of the given Learning Statement” where the Learning Statement was a section from the SOL. A score of 1 indicated they strongly agreed with the statement about the web resource, and 5 indicated they strongly disagreed.

Our definition of “relevant” was based not only on if the search terms produced an item that pertained to the SOL, but also on how educationally useful the web resource was that was being evaluated. Because our evaluators were determining whether or not a resource would help a student learn about a particular subject, a search result might match the search terms quite well and be considered relevant in the traditional sense, but the evaluator may have given the search result a much lower score if the resource did not help teach a student at a particular grade level. This could be because the page content was presented at a lower or higher level, making the content too trivial or difficult to understand.

The eleven teachers used as evaluators in our experiment were asked several questions in regards to their teaching experience, familiarity with the SOL, and familiarity with using the Internet as a teaching tool. The teachers were all from the Norfolk and Virginia Beach school systems. They have taught on average 5.1 years with 3.85 of those years teaching in the fields they evaluated. They were all very familiar with the SOL and using the Internet to augment classroom activities (average scores of 1.15 and 2, respectively, on a five point scale). Most teachers were unfamiliar with the NSDL (4.5 on a five point scale). We asked the teachers after the evaluation about their knowledge of the NSDL to avoid biasing the evaluations.

4 Evaluation Results

The eleven evaluators each spent an average of 1.5 hours to evaluate at least six sets of search terms and the roughly 20 search results generated by each query. Table 1 shows a total of 38 queries were produced from the SOL for all subjects producing a total of 334 (318 unique) links from NSDL and 380 (376 unique) links from Google. The evaluators produced 65 ratings for search

terms and evaluations for 572 NSDL-returned resources and 650 Google-returned resources.

Each search query contained anywhere from 1 to 5 words and averaged 3.28 words per query. This length seems reasonable since it is slightly smaller than the average query length used in some evaluations [6] and slightly larger than the average seen by some popular web search engines [19]. The search terms that we chose to find educational content from the SOL were given an average rating of 2.08 (median=2) by our evaluators indicating agreement that the search terms validly reflected the relevant SOL learning statements. When each query was performed, we took only the top 10 search results. Of the top 10 search results, the NSDL averaged 8.8 hits per query, and Google always produced 10. The lower number of NSDL hits may be due to the following reasons: the NSDL is querying from a smaller pool than Google; the NSDL performs an ANDing of each search term (hurting recall in favor of precision); and the NSDL shows search results for which no link is available (we did not use archived versions from either NSDL or Google).

The links returned by the NSDL and Google had little in common. The 38 queries performed generated only 6 duplicate links in the top 10 results. The same queries produced only 9 duplicates in the top 20 results. One reason for the lack of commonality is because one quarter (78 of 318) of the unique search results produced by the NSDL were not indexed by Google. 31 out of the 334 NSDL search results (9.3%) were given a score of 6 (inaccessible content, login-required access, etc.) compared to 20 out of the 380 Google search results (5.3%). These are web pages that were inaccessible due to any number of reasons: stale links, temporarily off-line servers, intermittent Internet problems, login-only access, etc. In our experiment we had the evaluators group login-only search results with other inaccessible sites because if a user is trying to access on-line material quickly, they are more likely to move on to the next search result rather than going through the time-consuming process of registering for an account. Of the 31 NSDL results given a score of 6, 25 of them (80.6%) were from the Learning Online Network with CAPA at <http://nsdl.lon-capa.org>. LON-CAPA accounted for 41 of the entire 334 NSDL search results (12.3%). 30 of the 41 LON-CAPA URLs (73.2%) were not indexed by Google.

Table 1. Summary of NSDL and Google search responses and evaluations

Subject	Evaluators	Search	Responses		Evaluated	Eval. Resp.	
		Queries	NSDL	Google	Srch Terms	NSDL	Google
Algebra I	1	6	56	60	6	56	60
Algebra II	1	7	56	70	7	56	70
Biology	3	6	52	60	15	134	150
Chemistry	1	6	52	60	6	52	60
Earth Sci.	4	7	62	70	25	218	250
Physical Sci.	1	6	56	60	6	56	60
Totals	11	38	334	380	65	572	650

Table 2. Descriptive statistics on NSDL and Google ratings

	Google	NSDL
Mean	3.16	3.74
Median	3	4
Std	1.57	1.48

Teachers rated each resource from 1 to 5 expressing, respectively, “strong agreement” or “strong disagreement” with the statement that the resources “would effectively help a student learn what was needed to fulfill the bold portion of the given Learning Statement.” A rating of 6 indicated that the particular resource was inaccessible. Throughout the remainder of our evaluation, we will use the term “agree” to indicate *higher relevance* when discussing relevance of the search results. The term “disagree” will indicate *lower relevance*, and the term “neutral” will indicate *uncertainty* about relevance.

We compared the teacher ratings for all NSDL and Google search results to determine which on average performed best in terms of the SOL learning statements. The median rating for all NSDL search results was found to be 4 (disagreement), and its mean score 3.74 with a standard deviation of 1.48. The median rating for all Google search results, in comparison, was found to be 3 (neutral), and its mean 3.16 with a standard deviation of 1.57. These results are listed in Table 2.

Both search engines in the mean did not generate results that would satisfy the educational requirements expressed in the ratings evaluation statement. In other words, most raters did not feel the returned resources would help a student learn what was needed to fulfill a portion of the given Learning Statements (median for Google and NSDL was respectively 3 (neutral) and 4 (disagree)). However, even though both search engines performed poorly in the mean, the median and mean score for the NSDL seemed to suggest it performed worse than Google. In fact, a Wilcoxon signed rank test revealed a statistically significant difference ($p < 0.05$) between the NSDL and Google ratings. This pattern becomes more evident when we examine the actual distribution of ratings, e.g. “how often did a NSDL or Google search produce a satisfactory result indicated by a 1 or 2 score?”

We first determined the precision of NSDL and Google by counting the number of results rated adequate (any score less than or equal to 2), denoted R , compared to the total number of results generated by that search engine, denoted N . To avoid double counts we aggregated multiple ratings for the same search result by calculating the mean rating for that item. Google produced a total of 380 search results for 38 queries, 145 of which were rated at a level ≤ 2 (agree to strongly agree). The precision for Google P_g was defined by:

$$P_g = R_g/N_g = 145/380 = 0.382 \text{ or } 38.2\% \quad (1)$$

NSDL’s precision (P_n) was defined in the same manner. 57 of the 334 NSDL search results were rated at a level ≤ 2 (agree to strongly agree), and P_n was determined as follows:

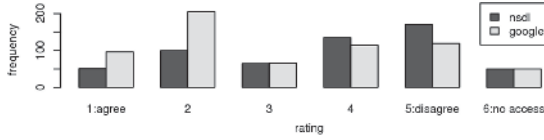


Fig. 3. Ratings for NSDL and Google search results (all domains)

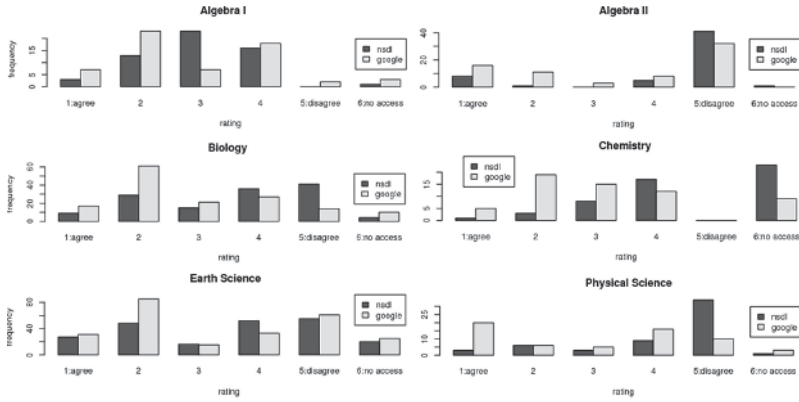


Fig. 4. Ratings for NSDL and Google search results (individual domains)

$$P_n = R_n/N_n = 57/334 = 0.171 \text{ or } 17.1\% \tag{2}$$

Although both precision levels were relatively low, NSDL performance, according to this metric, was significantly worse than Google’s. The fact that the NSDL produced an average of 8.8 results per query compared to Google’s 10 results per query strengthens this conclusion.

A frequency analysis was conducted of the NSDL’s and Google’s ratings to determine the performance differences at a lower level of granularity (Fig. 3). Figure 4 shows the NSDL and Google ratings according to the six different categories that were evaluated.

From Fig. 4 we can see that although NSDL and Google average ratings correspond to at best a neutral (3) or disagree (4), they differ considerably in their distributions. Google’s ratings tend to cluster in the 2-“agree” range, but follow a bi-modal distribution which peaks first at a rating of 2 and to a lesser degree at 4, indicating raters either approved or disliked the results, but tended to approve more than they disliked. The distribution of NSDL ratings is skewed to the right with the highest number of ratings observed in the 4-“disagree” and 5-“strongly disagree” range. Although the median of these distributions differs by only 1 point, these numbers hide strikingly different rating patterns favoring the Google results.

Table 3. Median ratings split according to domain category

	Google	NSDL	p-value
Algebra I	2.5	3	0.437
Algebra II	4	5	0.001 **
Biology	2	4	< 0.001 **
Chemistry	3	4	< 0.001 **
Earth Sci.	3	4	0.209
Physical Sci.	3	5	< 0.001 **

The bar plots in Fig. 4 indicate that although neither Google nor the NSDL perform exceptionally well, Google’s ratings are consistently higher than the NSDL’s for all subject domains. To verify this is indeed the case, we determined the median ratings for Google and the NSDL for the six mentioned domains. The results are listed in Table 3. For each comparison we ran a Wilcoxon signed rank test; p-values are listed in the third column.

P-values ≤ 0.001 (**) indicate statistically significant results. In all but two cases, namely Algebra I and Earth Science, the ratings indicate that Google significantly outperforms the NSDL. Furthermore, while only one of Google’s median ratings slip below a “neutral” and two are actually rated “agree” (2), only one of NSDL’s median ratings is a “neutral” (3), and all correspond to at least a “disagree” (4) rating, and two (Algebra II and Physical Science) correspond to a “strongly disagree” (5).

The effectiveness of a search engine is in great part determined by its ability to rank retrieved materials according to their relevance to the user query. In this case, however, our definition of relevance deviates from the traditional term-based model. We are not looking to match a query consisting of search terms to the resource that best matches those terms, but the resource which best matches the query in terms of its educational effectiveness and merit. We therefore need to determine the extent to which the ranking of search results from Google and NSDL match the obtained ratings that are based on the educational appropriateness and effectiveness of the returned results.

To investigate the degree to which the search result rankings of the NSDL and Google match expert ratings, we determined the Spearman Rank correlation between rankings and ratings. A small but statistically significant correlation was found between Google’s search results rankings and our expert ratings ($\rho=0.125$, $p=0.001$). In the NSDL’s case, no significant correlation was found ($\rho=0.057$, $p=0.173$). Google’s search result ranking to some degree approximates how teachers would rate a set of search results, but NSDL’s does not.

5 Future Work and Recommendations

After performing our experiment and evaluating the results, we have noted several areas that could improve future experiments:

Improve the search terms by getting input from the evaluators. We wanted to limit the amount of work that the teachers needed to perform, so we picked search terms that we believed represented the SOLs. Although our search terms were shown to have been acceptable by the evaluators, some improvements could have been made by giving the evaluators the chance to modify the terms and come to a general consensus as to what the best search terms would be.

Use more evaluators. Volunteers, even compensated volunteers, are hard to enlist. If, for example, principals had required teachers to participate in the study, we would have more evaluators but the quality of their reviews may have suffered if they were coerced into participating.

Provide immediate proctor support during the evaluations. We believe some of our evaluators may have been too harsh or too forgiving in assigning relevance because of a lack of guidance. Some may have not used the relevancy criteria that we explicitly asked them to evaluate. At least one of our evaluators gave consistently lower scores than other evaluators. A more tightly controlled, synchronous environment may have increased the quality of the reviews, but it would have restricted the number of possible participants.

In the course of our evaluation, we uncovered the following areas of improvement for the NSDL:

Rank search results according to quality. According to our analysis, the NSDL does not rank its search results according to perceived relevance. This has been suggested elsewhere [10].

Provide an advanced search capability. Although most users do not use the advanced searching features, those that do could benefit from producing more targeted queries. Originally the NSDL had just an advanced search page, then a usability study [12] suggested a simple search page. That was added, but the functionality of the advanced search page seemed to be reduced.

Provide the ability to target the grade level appropriateness of information. In our experiment, 16.5% of the NSDL results were from arXiv.org, an e-print archive focusing primarily on advanced research in physics, math, and computer science. Of these results, only one of them scored 2 or better. This suggests it may be useful for the NSDL to rate its content based on grade level appropriateness [23].

6 Conclusions

We have performed an experiment that demonstrates how educators might try to find educational material in the NSDL and Google. We created a real-life scenario whereby teachers needed to obtain web-based educational content based on educational requirements from the Virginia Standards of Learning. We queried the NSDL and Google using a variety of search terms and used paid volunteer teachers to evaluate the educational relevance of the search results.

Based on statistical analysis of the evaluation results, we found that Google tended to find more useful educational material than the NSDL; in 4 out of 6 subject areas, Google significantly outperformed the NSDL. Google's precision was found to be 38.2% compared to NSDL's 17.1%. Google's ranking of material outperformed the one applied by the NSDL search engine as indicated by its moderate correlation with expert ratings of resource relevance. Although the NSDL's collection of educational resources may be of higher quality and scope than what Google can offer, the latter's ranking will make those resources more easily and efficiently available. We analyzed the returned results from the NSDL and found that a significant portion of them required registration for accessing. About 1 in 4 NSDL resources were not indexed by Google. There was also very little overlap (6 duplicates in top 10 results from 38 queries) in the results returned by Google and NSDL. Finally we provided some guidelines for improving this type of evaluation in the future. We also provided some suggestions that could be used to improve the NSDL's search engine so that educators can use the NSDL to find relevant educational material more effectively than Google.

References

1. Arms, W., Dushay, N., Fulker, D., Lagoze, C.: A Case Study in Metadata Harvesting: The NSDL. *Library HiTech*. Vol. 21 **2** (2003) 228–237
2. AskNSDL. Response by Melodye Campbell. (March 20, 2004) <https://ask.nsd.org/default.aspx?from=srch&id=8337>
3. Bar-Ilan, J.: Methods for Measuring Search Engine Performance Over Time. *Journal of the American Society for Information Science and Technology*. Vol. 53 **4** (2002) 308–319
4. Bergmark, D.: Collection Synthesis. In *Proceedings from the second ACM/IEEE-CS joint conference on Digital libraries (JCDL 2002) Portland, Oregon, USA (2002)* 253–262
5. Bergmark, D., Lagoze, C., Sbityakov, A.: Focused Crawls, Tunneling, and Digital Libraries. In *Proceedings of the European Conference on Digital Libraries (ECDL), Rome, Italy (September 2002)* 91–106
6. Can, F., Nuray, R., Sevdik, A.: Automatic Performance of Web Search Engines. *Information Processing and Management*. Vol. 40 **3** (2004) 495–514
7. Chowdhury, A., Soboroff, A.: Automatic Evaluation of World Wide Web Search Services. In *Proceedings of the ACM SIGIR conference*. Vol. 25. (2002) 421–422
8. Chu, H., Rosenthal, M.: Search Engines for the World Wide Web: A Comparative Study and Evaluation Methodology. *Proceedings of the ASIS annual meeting*. **33** (1996) 127–135
9. Ding, W., Marchionini, G.: A Comparative Study of Web Search Service Performance. In *Proceedings of the 59th Annual Meeting of the American Society for Information Science*. Baltimore, MD, USA. **33** (1996) 136–142
10. Fulker, D.: Metadata Strategies to Address NSDL Objectives. In *Proceedings of the 5th Russian Conference on Digital Libraries (RCDL 2003)*. St. Petersburg, Russia. (2003)
11. Gordon, M., Pathak, P.: Finding Information on the World Wide Web: The Retrieval Effectiveness of Search Engines. *Information Processing and Management*. Vol. 35 **2** (1999) 141–180

12. Hartson, H., Shivakumar, P., Pérez-Quñones, M.: Usability Inspection of Digital Libraries: A Case Study. *International Journal of Digital Libraries*. Vol. 4 **2** (2003) 108–123
13. Hawking, D., Craswel, N., Bailey, P., Griffiths, K.: Measuring Search Engine Quality. *Information Retrieval*. Vol. 4 **1** (2001) 33–59
14. Lagoze, C., Hoehn, W., Millman, D., Arms, W., Gan, S., Hillmann, Ingram, C., Krafft, D., Marisa, R., Phipps, J., Saylor, J., Terrizzi, C., Allan, J., Guzman-Lara, S., Kalt, T.: Core Services in the Architecture of the National Science Digital Library (NSDL). In *Proceedings from the second ACM/IEEE-CS joint conference on Digital libraries (JCDL 2002) Portland, Oregon, USA.* (2002) 201–209
15. Leighton, H., Srivastava, J.: First 20 Precision Among World Wide Web Search Services (Search Engines). *Journal of American Society for Information Science*. Vol. 50 **10** (1999) 870–881
16. Ljosland, M.: Evaluation of Web Search Engines and the Search for Better Ranking Algorithms. *SIGIR99 Workshop on Evaluation of Web Retrieval* (1999)
17. NSDL Collection Development Policy Draft v.030715. Accessed 1-18-2005. http://content.comm.nsd.org/doc_tracker/docs_download.php?id=452
18. Silverstein, C., Henzinger, M., Marais, J., Moricz, M.: Analysis of a Very Large Alta Vista Query Log. Technical Report 1998-014. COMPAQ Systems Research Center, Palo Alto, CA. (1998)
19. Spink, A., Wolfram, D., Jansen, B., Seracevic, T.: Searching the Web: The Public and Their Queries. *Journal of the American Society for Information Science and Technology*. Vol. 52 **3** (2001) 226–234
20. Standards of Learning for Virginia Public Schools, <http://www.pen.k12.va.us/VDOE/Superintendent/Sols/home.shtml>
21. Su, L.: A Comprehensive and Systematic Model of User Evaluation of Web Search Engines: I. Theory and Background. *Journal of the American Society for Information Science and Technology*. Vol. 54 **13** (2003) 1175–1192
22. Su, L., Chen, H., Dong, X.: Evaluation of Web-based Search Engines from an End-user's Perspective: A pilot study. *Proceedings of the 61st Annual Meeting of the American Society for Information Science*. Pittsburgh, PA, USA. (1998) 348–361
23. Sumner, T., Khoo, M., Recker, M., Marlino, M.: Understanding Educator Perceptions of “Quality” in Digital Libraries. In *Proceedings of the Third ACM+IEEE Joint Conference on Digital Libraries. (JCDL 2003) Houston, Texas, USA.* (2003) 269–279
24. Zia, L.: The NSF National Science, Mathematics, Engineering, and Technology Education Digital Library (NSDL) Program: A Progress Report. *D-Lib Magazine*. Vol. 6 **10** (2000)

Policy Model for University Digital Collections

Alexandros Koulouris and Sarantos Kapidakis

Laboratory on Digital Libraries and Electronic Publishing,
Department of Archive and Library Sciences, Ionian University, Plateia Eleftherias,
Palaia Anaktora, Corfu 49100, Greece
akoul@ionio.gr, sarantos@ionio.gr

Abstract. The access and reproduction policies of the digital collections of ten leading university digital libraries worldwide are classified according to factors such as the creation type of the material, acquisition method, copyright ownership etc. The relationship of these factors is analyzed, showing how acquisition methods and copyright ownership affect the access and reproduction policies of digital collections. We conclude with rules about which factors lead to specific policies. For example, when the library has the copyright of the material, the reproduction for private use is usually provided free with a credit to the source or otherwise mostly under fair use provisions, but the commercial reproduction needs written permission and fees are charged. The extracted rules, which show the common practice on access and reproduction policies, constitute the policy model. Finally, conventional policies are mapped onto digital policies.

1 Introduction

Libraries are in a transition period from conventional to digital formats and have not yet developed common practices and traditions on policy for digital material. This can prevent cooperation and interoperability in libraries, restricting the usefulness of their services. Conventional policies do not map directly onto digital ones with differences primarily due to the easy duplication properties of the digital material.

University libraries have well established traditions of cooperation and they play a leading role in the production and dissemination of digital material. In addition, they have a leading role to play in using new technologies, such as digital libraries. They have implemented an appropriate infrastructure for the development of digital libraries and policies and they facilitate the use of new technologies by students. Due to their leading position in the academic system and in the scientific community, they have prestige and their practices and policies can be easily disseminated to the rest of the library sector (national, public libraries etc.).

In this paper, the access and reproduction policies of the digital collections of twenty leading university digital libraries from three continents were examined and ten of them are presented here; those which are considered to have the most diversified and innovative access and reproduction policies and are the most active in the area of digital libraries. We were interested in libraries that have large digitization projects and preferably, they use or, even better develop, commonly used software

(such as *Greenstone* [16]) to provide their digital content. In addition, we were interested in libraries which have collections with various *creation types* (digitized, born-digital) or content types of material (video, audio etc.), various copyright owners (libraries, individuals, organizations such as publishers etc.), diversified access and reproduction policies and various acquisition methods (license, purchase, digitization of library or third-party material etc.). The libraries studied, corresponding to the above criteria, are central academic libraries of large universities, which act as the coordinating bodies for the entire library system of their universities.

Analyzing and studying a big sample of university libraries, we realized that the majority of them follow similar policy rules and models. We selected twenty of them to analyze further, because they differentiate on policies, material and vision; more libraries will give us more quantitative but not qualitative results. We present here ten of them, because they contain all applied policies that appear on these twenty libraries, which we analyzed further.

Meyyappan [11], who described the status of twenty digital libraries, mentioning also their access policies, has undertaken similar work previously. In addition, Walters [15], presents an introduction to the acquisition of video media (DVD and VHS) in academic libraries, with an emphasis on the procedures most appropriate for undergraduate colleges. However, no previously studies have focused exclusively on access and reproduction policies.

To collect the data for this study, we derived information from the websites, in some cases supplemented by personal communication with the libraries. In section 2, we classify the policies according to factors such as the type of the material, its acquisition method, copyright ownership etc., and we present some quantitative data, indicating which factors lead to specific policies. We also analyze how the factors affect the policies. In section 3, we present our policy model by extracting common practice and deriving generalized rules on policies for digitized and born-digital material separately. We present the most common practices on policies, which the libraries follow. In section 4, we map conventional access and reproduction policies on to their digital counterparts and we compare them, focusing mostly on their differences. We examine which conventional policies can be mapped to new digital policies and discuss the advantages and disadvantages of this transition. Finally, concluding remarks are made in section 5 and future work is illustrated.

Most libraries face difficulties in resolving the problems that arise due to the properties of the digital material, such as, for instance, the access to university digital collections by students on campus and by distance learners. This paper provides useful information on university library practices concerning these problems and about common practices related to access and reproduction policies.

The strategic and philosophical question is, if the policy should be articulated at the beginning of the design of digital library or in practice before making electronic resources available to public. We distinguish the policy implementation mechanisms and the policies themselves. The mechanisms should be designed from the beginning with the necessary flexibility in order to apply policies, which may be determined later on during the use, and should be customizable to the needs of the user community that each digital library focuses on. However, we should have from the beginning an idea of the policies that will be used, but mostly, we have to implement flexible mechanisms. As long as we have the policy implementation mechanisms, we

can choose or change the appropriate policy for each kind of material whenever we need. Situations like these are common, especially for organizational or interoperability reasons.

2 Classification of Policies

In Table 1, we classify the access and reproduction policies of the university digital collections, according to the type of the material, the acquisition method and copyright ownership. The relations among the factors and the policies are analysed and the diversification of policies that arise is illustrated and presented in section 3, where we analyze the proposed policy model.

The majority of *values* in Table 1 are abbreviations, used for formatting reasons. The values are mentioned for each column and case and they are explained below. The table is ordered according to the creation type of the material (third column, value: *C-t*). There are three blocks or categories: digitized (value: *Dig*), digitized and born-digital (value: *Dig, B-d*), and born-digital (value: *B-d*). Subsequently, each block is sorted according to geographical location (second column, value: *Loc*), and finally, the collections or the libraries (first column, value: *Col/Lib*) are ordered alphabetically for each geographical location.

The first (*Col/Lib*) and second (*Loc*) columns give information about the name of the collection and their location. The first column contains the collection names from ten university libraries. Each row represents either a whole library or some of its parts (split into separate rows), because they are diversified in terms of factors and policies from the rest of the collections of the library presented as a whole. We present the abbreviations of the libraries and collections in turn: *Aladin Digital Library* (ADL), *Felix E. Grant Collection* (FEG [14]) of ADL, *Historical Monograph Collection* (HM [4]) of *Cornell University Library* (COUL [3]), *Image Collections* (IC [4]) of COUL, *Northwestern University Library* (NUL [13]), *North Carolina State University Libraries* (NCSUL [12]), and *Samuel J. May Anti-Slavery Collection* (SJMAS [5]) of COUL. *Cambridge University Library* (CUL [1], [2], [8]), *Miguel de Cervantes Digital Library* (MdCDL), *New Zealand Digital Library* (NZDL [16]), *Harvard University Library* (HUL [10]), *Glasgow Digital Library* (GDL [9]), *Dartmouth College Digital Library* (DCDL [6], [7]), *Past Masters* (PM) and *Patrologia Latina* (PL) of COUL.

The third column (*C-t*) shows how the digital material was created. We have two cases: *digitized* (*Dig*) or *born-digital* (*B-d*) and a collection can have either one or both (*Dig, B-d*) types of material. The fourth column (*A-m*) shows the acquisition method of the material, which has five values: *library* (*Lib*), *third-party* (*T-p*), *license* (*Lic*), *purchase* (*Pur*) and *voluntary deposit* (*V-d*). *Library* means that the library has created its own digitized or born-digital material. *Third-party* means that the library has digitized and/or acquired born-digital third-party material, which may be free or restricted by the owner. *License* means that the library has acquired digitized and/or born-digital material through license. *Purchase* means that the library has purchased digitized and/or born-digital material. *Voluntary deposit* means that the library has acquired born-digital material through voluntary deposit.

Table 1. Factors of access and reproduction policies for digital collections

Col/Lib	Loc	C-t	A-m	C-o	Off-c	Offsite	P-r	Commercial reproduct		
								Allow	W-p	Fee
ADL	USA	Dig	Lib	Lib	Yes	Yes	Yes	No	N/A	N/A
FEG	USA	Dig	Lib	Lib, Ind, Org	No	No	Yes	Yes	Own	Own
HM	USA	Dig	Lib	Lib, P-d	Some	Some	Fair	Yes	Lib	Lib
IC	USA	Dig	Lib, T-p	Lib, Ind,	Yes	Some	Fair	Yes	Lib, Own	Lib, Own
NUL, NCSUL	USA	Dig	Lib, T-p	Lib+	Yes	M-yes	Case	Case	Case	Case
NCSUL	USA	Dig	Lib	Vary	Yes	Some	Fair	No	N/A	N/A
SJMAS	USA	Dig	Lib	Lib	Yes	Yes	Fair	Yes	Lib	Lib
CUL	UK	Dig	Lib, T-p	Lib+	Yes	M-yes	Yes	Yes	Lib, Own	Lib, Own
MdCDL	ES	Dig	Lib, T-p	Lib, P-d	Yes	Yes	Yes	Some!	Lib	No
NZDL	NZ	Dig	T-p	Lib, Ind, Org, P-d	Yes	Yes	Yes	No	N/A	N/A
HUL	US	Dig, B-d	Lib	Lib	Yes	M-no	Yes	Yes	Lib	Lib
			Lic, Pur,	Lib, Org	Some	M-no	Yes	Yes	Lib	Lib, Own
MdCDL	ES	Dig, B-d	Lic	Vary	Yes	Yes	Yes	No	N/A	N/A
GDL	UK	Dig, B-d	Lib, T-p	Lib, Ind, Org	Yes	Yes	Yes	Yes	M- Own	Own
DCDL	USA	B-d	Lib	Lib	Yes	Some	Fair	Yes	Lib	Lib
			Lic, Pur	Org	Some	No	Fair	Yes	Own	Own
NUL, NCSUL	USA	B-d	Lic, Pur	Lib, Org	Some	No	Case	Case	Case	Case
NCSUL	USA	B-d	Lic, Pur	Vary	Yes	Some	Fair	No	N/A	N/A
PM, PL	USA	B-d	Lic	Ind	Yes	No	Fair	No	N/A	N/A
CUL	UK	B-d	Lic, Pur, V-d	Org	Vary	M-no	Yes	No	N/A	N/A

+ The library is usually the owner of the digitized material, but sometimes there are other owners or the material is in public domain.
! The commercial reproduction is usually prohibited, but in some cases is merely permitted with written permission from the MdCDL.

The fifth column (*C-o*) presents the copyright owner, which has five values: *library (Lib)*, *individual (Ind)*, *organization (Org)*, *vary* and *public domain (P-d)*. *Library* means that the copyright of the material is owned by the organization that the library belongs to, and is administered by the library. *Individuals* and/or *organizations* mean that the copyright belongs to owners other than the library, which can be individuals and/or organizations respectively; this is the meaning of the term *other*

owner(s) that is used frequently on the following sections. *Vary* means that the copyright varies from collection to collection. *Public domain* means that nobody has or claims the copyright of the material.

Access policies are stated in the sixth and the seventh columns. *On-campus access for onsite users is always free*. The sixth column (*Off-c*) shows the off-campus access policy for onsite users and the seventh (*Offsite*) the offsite access policy. In these columns, we have three values: *yes*, *no* and *some*. *Yes* means that the off-campus onsite and the offsite access are both free. *No* means that the off-campus onsite and the offsite access are not provided. *Some* means that the off-campus onsite and the offsite access are provided in some cases. In off-campus onsite access, we also have the value *vary*, meaning that the off-campus onsite access varies from item to item. In the offsite access column, we also have the values *mostly no (M-no)*, meaning that the offsite access is not provided in most of the cases, and reciprocal, *mostly yes (M-yes)*, meaning that the offsite access is provided in most of the cases.

The off-campus onsite access always refers to onsite users, students, faculty, staff etc., which are affiliated with the university, and they may access the material outside of the university, independently of their location, usually by using user name and password authentication. For example, a Greek student may have access to e-journals of CUL, from its home in Greece, during summer. In contrary, offsite access refers to the rest of the users that are not affiliated with the university, which use the Internet, for accessing the material – without having the privilege of authentication and most of the times with different and restricted access rights from off-campus onsite users.

The eighth column (*P-r*) shows the *private reproduction* policy (or reproduction for private use), which has three values: *yes*, *fair use (Fair)* and *case-by-case (Case)*. *Yes*, means that the private reproduction is free with a *credit* (mention) to the source, *fair use* means that it is provided under fair use provisions and *case-by-case* means that it is on a case-by-case basis.

The ninth, tenth and eleventh columns refer to the commercial reproduction policy (*Commercial reproduct*). The ninth column (*Allow*) shows if commercial reproduction is permitted, which has four values: *yes*, *some*, *no*, and *case-by-case (Case)*. *Yes* means that the commercial reproduction is permitted with written permission from and fees paid to the owner (library and/or other owners), but sometimes (e.g. MdCDL), even if written permission is needed, fees are not charged. *Some* means that the commercial reproduction is sometimes permitted, *no* means that it is not authorized and *case-by-case* means that it is on a case-by-case basis.

The tenth column (*W-p*) states who gives the written permission for the commercial reproduction, if it is needed. We have five values: *library (Lib)*, *owners (Own)*, *owners mostly (M-own)*, *case-by-case (Case)* and *N/A*. *Library* means that the written permission is given by the library, *owners* means that it is given by owners other than the library, *owners mostly* means that it is given mostly by other owners and less by the library. *Case-by-case* means that the written permission is examined on case-by-case basis and *N/A* means that it is not applicable. *Library* and *owners* can appear as value *Lib*, *Own*, if both the library and the owners require written permission.

The eleventh column (*Fee*) states to whom the fee should be paid for commercial reproduction, if it is needed. We have five values: *library (Lib)*, *owners (Own)*, *case-by-case (Case)*, *no* and *N/A*. *Library* means that the fee is paid to the library. *Owners*

mean that the fee is paid to owners other than the library. *Case-by-case* means that the payment of the fee is examined on case-by-case basis, *no* means that a fee is not charged and *N/A* means that it is not applicable. If the value *library, owners (Lib, Own)* appears, the fee should be paid to both library and owners.

Some general rules for the handling of digitized and born-digital material can be derived from Table 1 and its discussion, showing that there is a variety of arrangements depending on ownership of the material and its copyright. We present those rules and their exceptions on section 3, where we analyze and present our policy model.

2.1 Quantitative Analysis of Table and Remarks

From the analysis above, we see that specific factors lead to specific policies. We can derive some quantitative data and remarks that are extracted from Table 1, about which factors lead to specific policies, which are usually related to the copyright factor.

2.1.1 Acquisition and Copyright

- Libraries seem to prefer (79%) digitizing their own material on which they have the copyright.
- Libraries also often digitize free third-party (43%) or public domain (21%) material.
- Born-digital material is acquired mostly (70%) through license and/or purchase from copyright owners (organizations, individuals etc.).
- Sometimes (in 30% of cases) libraries create their own born-digital material on which they usually have the copyright.
- Born-digital material is voluntarily deposited in 10% of cases, meaning, rarely.
- When the acquisition of born-digital material is mostly through license and/or purchase, then the copyright belongs to other owners (43% of cases) or to the library and other owners (29%) otherwise it varies from item to item (28%).

2.1.2 Copyright, off-Campus Onsite and Offsite Access

- In 68% of cases, off-campus onsite access is provided. 22% provide it in a limited sense. Only 5% do not provide off-campus onsite access and, in 5 % of cases, it varies according to the collection.
- 42% of the libraries provide full offsite access. 21% provide limited access and 37% do not provide access at all.
- When offsite access is limited or not provided, there are licensing restrictions and/or the copyright belongs to other owners, or sometimes (in approximately 11% of cases), the library, even if is the copyright owner, provides only onsite access.
- In 47% of cases, off-campus and offsite access are different. In such cases, when off-campus onsite access is limited, offsite access is not provided due to licensing and copyright restrictions. In addition, when off-campus onsite access is provided, offsite access is limited or it is not provided because either the library provides only onsite access or the copyright belongs to other owners. Finally, when off-campus onsite access varies from item to item, offsite access is not provided.

2.1.3 Copyright and Private Reproduction

- Private reproduction is usually free with a credit to the source (in 53% of cases) or under fair use (37%) or it is on case-by-case basis (10%).
- When the library has the copyright of the material, then private reproduction is free with a credit to the source (in 50% of cases), or under fair use provisions (50%).
- When mostly the library, or the library and other owners have the copyright, then private reproduction is free with a credit to the source (in 60% of cases), or it is under fair use provisions (20%) or it is on case-by-case basis (20%).
- When other owners have, the copyright, then private reproduction is under fair use provisions (67% of cases) or it is free with a credit to the source (33%).

2.1.4 Copyright and Commercial Reproduction

- 53% of cases allow commercial reproduction with written permission from and fees paid to the owner (library and/or other owners). 37% prohibited it and 10% decide it on case-by-case basis.
- When the commercial reproduction needs written permission from the owner, then fees are also paid to the owner.
- 75% allow commercial reproduction with written permission from and fees paid to the library. 40% allow it with written permission from and fees paid to the owners (when the library, individuals, and organizations have the copyright).
- 50% allow commercial reproduction when organisations have the copyright with written permission from and fees paid to the owners. In 50% of cases, it is not authorized.
- When individuals have, the copyright, then commercial reproduction is usually not authorized.
- 67% permit commercial reproduction when the library mostly has the copyright with written permission (mostly from the library and rarely from the owners) and fees paid to the library and the owners. The remainder (33%) allow it on case-by-case basis.
- When the library and organizations or library and individuals have the copyright, then the commercial reproduction is usually decided on case-by-case basis, or it is allowed with written permission from and fees paid to owners.

3 Policy Model: Rules and Exceptions

From this analysis, we can derive some generalized rules, policy model, about which factors lead to specific policies. Common practice shows that the on-campus onsite access is always free, independent of copyright ownership and the creation type of the material. In addition, when there are copyright uncertainties, notwithstanding the creation type of the material, a common solution is that reproduction (private and commercial) is decided on case-by-case basis (e.g. NUL, NCSUL).

We present a policy model that contains rules for the digitized (Figure 1) and the born-digital (Figure 2) material separately. The rules refer to the common practices that the majority of university libraries follow and use. We divide the rules mostly by using the factor of copyright ownership. In addition, we present the exceptions of the rules that supplement the policy model.

On the two figures presented below, the thick arrows show the most common rule; the dots indicate the access policies and the dashes the private and commercial reproduction policies. The figures are organised onto three layers. The first contains the creation type of the material and its acquisition methods; directs to the second one that represents the copyright ownership; and finally, directs to the third one that represents the access and reproduction policies, showing how the factors affect the policies.

3.1 Policy Model for the Digitized Material

Most libraries have decided to digitize their own material on which they have copyright. Common approaches showing that the libraries have the copyright for the digitized version of the free third-party and public domain material. A reasonable choice for the libraries would be to provide their own copyrighted digitized material with free onsite (on and off-campus) and offsite access, to permit private reproduction with a credit to the source and to require written permission and fees for commercial reproduction.

When the copyright varies (libraries and other owners, other owners only, varies from item to item) – on licensed or third-party digitized copyrighted material – the common approach is the provision of free onsite and no offsite access. Private and commercial reproduction should be permitted to onsite users only (as the access) with a credit to the source and with written permission from and fees paid to the owner (library and/or other owners) respectively. Another frequently used practice is the prohibition of commercial reproduction, which is used very often when the copyright

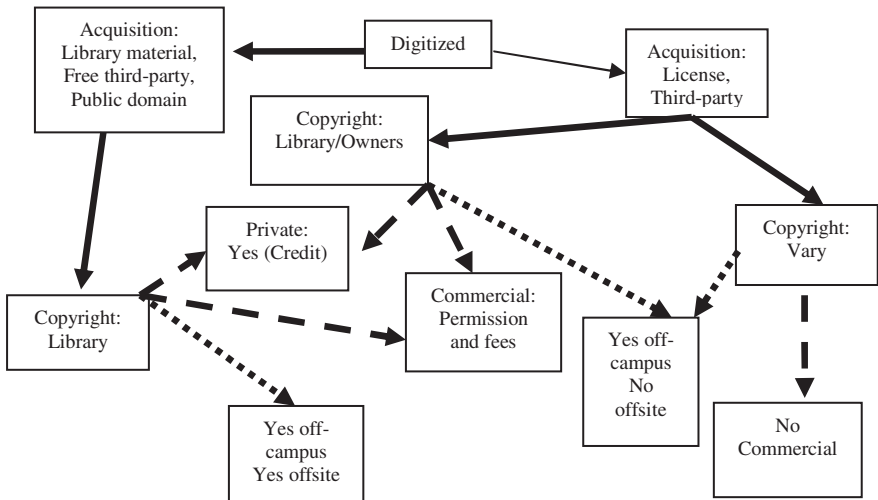


Fig. 1. Policy model for the digitized material (dots = access policies; dashes = reproduction policies)

varies on item-by-item basis. The digitized material is not so often acquired by license or third-party, but when this happens, the previous model and rules are applied on access and reproduction policies.

On the previously presented policy model, there are some exceptions of the rules. The exceptions may be applied on three cases. First, when the libraries have or administer the copyright for the digitized version of free third-party and public domain material, it is possible to provide it with free onsite and no offsite access, to apply fair use provisions for the private reproduction, and to examine the commercial on case-by-case basis.

Second, when libraries and other owners, share the copyright of the free third-party digitized material, the access could be prohibited mostly or be provided for onsite users only and be *limited* (some) for offsite. Fair use provisions may be applied for the private reproduction; and for the commercial, the basic rule of written permission from and fees paid to owners (library and/or other owners) is followed.

Finally, when the copyright varies on item basis, which is encountered mostly on the licensed material, the access may be provided to all users, the private reproduction may follow the fair use doctrine, but the commercial follows the rule of prohibition.

3.2 Policy Model for the Born-Digital Material

Most libraries acquire born-digital material through license and/or purchase with organizations (e.g. publishers) and individuals. Most libraries have decided to acquire licensed born-digital material if they will be responsible for the use of this material. Common practice shows that mostly other owners have the copyright of the licensed material. A reasonable choice would be for the libraries to provide their own

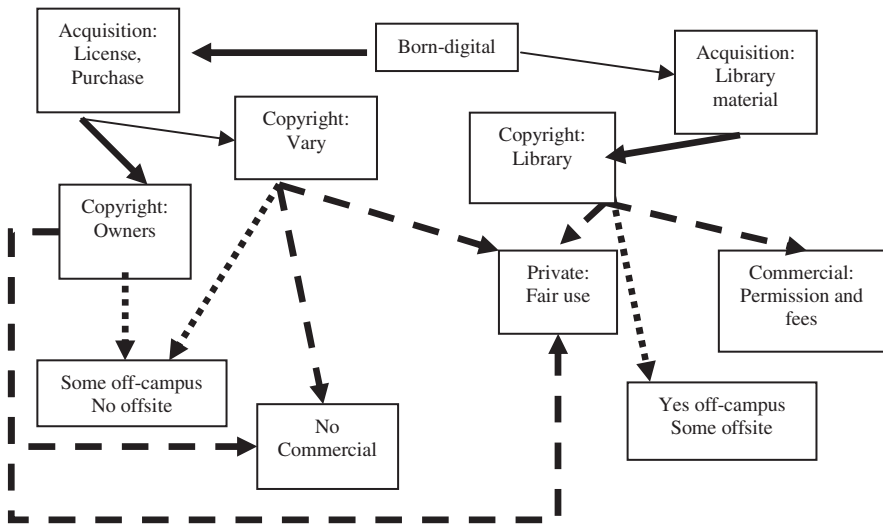


Fig. 2. Policy model for the born-digital material (dots: access policies, dashes: reproduction policies)

copyrighted born-digital material with free onsite and some offsite access, to permit private reproduction under fair use provisions and to require written permission and fees for commercial reproduction. Libraries do not very often create born-digital material, but when this happens, they have the copyright; and the previous model and rules are applied on access and reproduction policies.

When there are licensing restrictions and the copyright varies (libraries and other owners, other owners only, varies on item-by-item basis), the common approach is the provision of free on-campus, some off-campus and no offsite access. Common practice shows that private reproduction is permitted under fair use provisions, and commercial reproduction is not authorized. The previous rules are applied also, when the libraries acquire born-digital material through voluntary deposition – something that happens rarely – and the copyright belongs to other owners.

On the previously presented policy model, there are some exceptions of the rules. The exceptions may be applied on three cases. First, when libraries create their own copyrighted born-digital material, the only exception-difference from the previous model is on access. Instead of providing free onsite and some offsite access, the libraries provide free onsite but prohibit offsite access.

Second, when libraries and other owners share the copyright of the licensed and/or purchased material, which is an alternative approach of the basic rule that other owners have the copyright, the exceptions refer to the private and commercial reproduction; the access follow the rule: free on-campus, some off-campus and no offsite. The private reproduction is sometimes permitted with a credit to the source and the commercial, with written permission from and fees paid to owners, whoever they are, libraries and/or other owners. Another alternative reasonable choice is the examination of private and commercial reproduction on case-by-case basis.

Finally, when the copyright varies on item-by-item basis, the exception refers on access. Instead of following the rule of free on-campus, some off-campus and no offsite access, the libraries may provide free access for all or free for onsite and some for offsite. The private and commercial reproduction, follow the rule of fair use doctrine and prohibition, respectively.

4 Mapping Conventional onto Digital Policies

Conventional access and reproduction policies were mapped onto and compared with their digital counterparts. Differences were apparent. Conventional access inside library premises for printed material corresponds to Internet access inside library premises for digital material. However, Internet access (digital material) can be concurrent and simultaneous through workstations inside library premises, unlike conventional access because of the restricted number of copies (printed material). Conventional library loans for on-campus onsite users correspond to on-campus onsite Internet access. Unlike conventional loans for which the user must visit the library, on-campus onsite Internet access is remote (e.g. campus room, workstations outside of the library).

Conventional *inter-library loan* (ILL) corresponds to off-campus onsite and offsite Internet access. However, ILL is complicated, time-consuming, restricted to users of specific affiliated libraries and needs the intermediation of the librarian. In contrast,

off-campus onsite and offsite Internet access is simple, rapid, and can be remote, independent of the user's affiliation or location, and without intermediation. In general, libraries can implement more liberal digital policies than with conventional material, because of the relaxing of the physical copy restriction or the protection problem. However, copyright limitations may restrict the digital policies too.

Digital reproduction (credit or fair use) corresponds to conventional reproduction (photocopying) inside library premises under fair use provisions and to conventional document delivery procedures. However, the librarian can restrict the extent of conventional photocopying or document delivery procedure and enforce limitations on local users. In contrast, limitations on digital reproduction cannot be enforced and it is the user's responsibility to respect the fair use doctrine. However, in cases where there are licensing and copyright limitations, the library may forbid not only reproduction but also remote access.

5 Conclusions and Future Work

The relationship among specific factors and the access and reproduction policies of the digital collections of leading university digital libraries has been examined. Policies were analysed and classified; quantitative remarks were extracted and a policy model was proposed. The proposed policy model is not only comprised of the most common practices that the libraries implement, but also, of new ones that have not been implemented so far, and may offer solutions on the selection of policies. In addition, it consists of generalised rules, about which factors lead to specific policies, supplemented by their exceptions; and recommendations for decision-makers or library managers in forming policies of digital libraries.

For example, when there are copyright uncertainties, in complex material, notwithstanding the creation type of the material, a common solution is that reproduction, both private and commercial, is decided on case-by-case basis. In addition, copyright ownership defines commercial reproduction policy, which, when allowed, needs written permission from, and fees paid to, the owner. The recommendations given contain not only rules, e.g. previously mentioned, but also exceptions that decision-makers should not follow. For example, for the licensed born-digital material the library should be responsible for the use, in every case, even if it is not the copyright owner; so decision makers should not acquire licensed born-digital material if they do not have control on its use. Another finding is that the university libraries of the USA allow private reproduction mostly by applying the fair use doctrine. The rest of them allow private reproduction with reference to the authors. Consequently, decision makers should follow the rule for private reproduction based on geographical criteria.

At the end, we mapped and compared conventional and digital policies, focusing mostly on their differences. We made this mapping because many problems arise by the fact that conventional policies do not map directly but indirectly to digital ones. We tried to answer the question if digital policies are more liberal than the conventional or restricted by copyright and licensing terms.

For future work, we should try to describe our policy model in a more formal way, e.g. by means of ontologies. In addition, we may examine more libraries and try

applying statistical or data mining methods for our quantitative analysis. Finally, it may be of interest to extend our policy model from university to also national libraries, by providing generalized policy rules that are common, on national and university libraries, or unique, and to compare the diversified policies that may arise to better explore and generalize the similarities and differences between national and university library policies.

References

1. Cambridge University Library, 2003. *Access to Electronic Resources*, <http://www.lib.cam.ac.uk/electronicresources/AccessTable.html> [Accessed 28-02-05]
2. Cambridge University Library, 2002. *Passwords for Electronic Resources*, <http://www.lib.cam.ac.uk/electronicresources/athens.htm> [Accessed 28-02-05]
3. Cornell University Library, 2004. *Cornell Library Digital Collections*, <http://cdl.library.cornell.edu/> [Accessed 28-02-05]
4. Cornell University Library, 2003. *Guidelines for Using Text and Images from Cornell Digital Library Collections*, <http://cdl.library.cornell.edu/guidelines.html> [Accessed 28-02-05]
5. Cornell University Library, 2003. *Samuel J. May Anti-Slavery Collection: Copyright Restrictions*, <http://www.library.cornell.edu/mayantislavery/permissions.htm> [Accessed 28-02-05]
6. Dartmouth College Library, 2003. *Access the Digital Library from Off-Campus*, <http://diglib.dartmouth.edu/libserv/offcampus.shtml> [Accessed 28-02-05]
7. Dartmouth College Library, 2004. *Copyright Policy and Guidelines*, <http://www.dartmouth.edu/copyright/index.html> [Accessed 28-02-05]
8. Department for Culture, Media and Sport Working Party, 1998. *Code of practice for the voluntary deposit of non-print publications*, <http://www.bl.uk/about/policies/codeprac.html> [Accessed 28-02-05]
9. Glasgow Digital Library, 2002. *Collection Development and Management Policy*, <http://gdl.cdlr.strath.ac.uk/documents/gdlcollectionpolicy.htm> [Accessed 28-02-05]
10. Harvard University Library, 1999. *LDI Intellectual Property and Copyright*, <http://hul.harvard.edu/ldi/html/copyright.html> [Accessed 28-02-05]
11. Meyyappan, N., 2000. A review of the status of 20 digital libraries, *Journal of Information Science*, 26 (5) pp. 337-355
12. North Carolina State University Libraries, 2002. *Electronic Resources – Conditions of Use*, <http://www.lib.ncsu.edu/eresources/conditions.html> [Accessed 28-02-05]
13. Northwestern University Library, 2003. *Rights, Permissions and Reproductions Policies*, <http://www.library.northwestern.edu/policy/rpr/index.html> [Accessed 28-02-05]
14. University of the District of Columbia, 2003. *Radio Interviews*, <http://jazz.wrlc.org/fgrant/interviews.html> [Accessed 28-02-05]
15. Walters, W.H., 2003. Video media acquisitions in a college library, *Library Resources & Technical Services*, 47 (4) pp. 160-170
16. Witten, I.H., 2003. Examples of Practical Digital Libraries: Collections Built Internationally Using Greenstone, *D-Lib Magazine*, 9 (3), <http://www.dlib.org/dlib/march03/witten/03witten.html> [Accessed 28-02-05]

Importance of HTML Structural Elements and Metadata in Automated Subject Classification

Koraljka Golub and Anders Ardö

Knowledge Discovery and Digital Library Research Group (KnowLib),
Digital Information Systems, Department of Information Technology, Lund University ,
P.O. Box 118, 22 100 Lund, Sweden
{anders.ardo, koraljka.golub}@it.lth.se
<http://www.it.lth.se/knowlib/>

Abstract. The aim of the study was to determine how significance indicators assigned to different Web page elements (internal metadata, title, headings, and main text) influence automated classification. The data collection that was used comprised 1000 Web pages in engineering, to which Engineering Information classes had been manually assigned. The significance indicators were derived using several different methods: (total and partial) precision and recall, semantic distance and multiple regression. It was shown that for best results *all* the elements have to be included in the classification process. The exact way of combining the significance indicators turned out not to be overly important: using the F1 measure, the best combination of significance indicators yielded no more than 3% higher performance results than the baseline.

1 Introduction

Automated subject classification has been a challenging research issue for several decades now, a major motivation being high costs of manual classification. The interest rapidly grew around 1997, when search engines couldn't do with just full-text retrieval techniques, because the number of available documents grew exponentially. Due to the ever-increasing number of documents, there is also a danger that recognized objectives of bibliographic systems (finding, collocating, choice, acquisition, navigation) ([19], p. 20-21) would get left behind; automated means could be a solution to preserve them (ibid., p. 30). Automated subject classification of text finds its use in a wide variety of applications, such as: organizing documents into subject categories for topical browsing, which includes grouping search results by subject; topical harvesting; personalized routing of news articles; filtering of unwanted content for Internet browsers; and, many others (see [17], [12]).

A frequent approach to Web-page classification has been a bag-of-words representation of a document, in which all parts of a Web page are considered to be of equal significance. However, unlike other text documents, Web pages have certain characteristics, such as internal metadata, structural information, hyperlinks and anchors, which could serve as potential indicators of subject content. For example, words from title could be more indicative of a page's content than headings. The degree to which

different Web page elements are indicative of its content is in this paper referred to as significance indicator.

With the overall purpose of improving our classification algorithm (see section 2.3), the aim was to determine the importance of distinguishing between different parts of a Web page. Significance of four elements was studied: title, headings, meta-data, and main text.

The paper is structured as follows: in the second chapter a literature review is given, evaluation issues are discussed and the algorithm used is described (2 Background); in the third chapter data collection as well as methodology for deriving significance indicators are described (3 Methodology); deriving and testing the significance indicators is presented in chapter 4 (4 Significance indicators). The paper ends with conclusions and further research (5 Conclusion).

2 Background

2.1 Related Work

A number of issues related to automated classification of documents and significance of their different parts have been explored in the literature. A. Kolcz, V. Prabakar-murthi, J. Kalita and J. Alspector [14] studied news stories features and found out that initial parts of a story (headline and first two paragraphs) give best results, reflecting the fact that news stories are written so as to capture readers' attention. J. Pierre [16] gained best results in targeted spidering when using contents of keywords and description metatags as the source of text features, while body text decreased classification accuracy. R. Ghani, S. Slattery & Y. Yang [10] also showed that metadata can be very useful for improving classification accuracy. A. Blum & T. Mitchell [4] compared two approaches, one based on full-text, and one based on anchor words pointing to the target pages, and showed that anchor words alone were slightly less powerful than the full-text alone, and that the combination of the two was best. E. Glover et al. [11] claimed that text in citing documents close to the citation often had greater discriminative and descriptive power than text in target documents. Similarly, A. Attardi, A. Gulli & F. Sebastiani [3] also used information from the context where a URL that refers to that document appears and got encouraging results. J. Fürnkranz [9] used portions of texts from all pages that point to the target page: the anchor text, the headings that structurally precede it, the text of the paragraph in which it occurs, and a set of (automatically extracted) linguistic phrases that capture syntactic role of the anchor text in the paragraph; headings and anchor text proved to be most useful.

On the other hand, R. Ghani, S. Slattery & Y. Yang [10] claim that including words from linked neighborhoods should be done carefully since the neighborhoods could be rather "noisy". Different data collections contain Web pages of various characteristics. If certain characteristics are common to the majority of Web pages in the collection, an appropriate approach taking advantage of those could be applied, but if the Web pages are very heterogeneous, it is difficult to take advantage of any of the Web-specific characteristics (cf. [22], [8], [18]).

2.2 Evaluation Challenge

The problem of deriving the correct interpretation of a document's subject matter has been much discussed in the library science and related literature. It has been reported that different people, whether users or subject indexers, would assign different subject terms or classes to the same document. Studies on inter-indexer and intra-indexer consistency report generally low indexer consistency ([15], p. 99-101). There are two main factors that seem to affect it: 1) higher exhaustivity and specificity of subject indexing both lead to lower consistency (indexers choose the same first term for the major subject of the document, but the consistency decreases as they choose more classes or terms); 2) the bigger the vocabulary, or, the more choices the indexers have, the less likely they will choose the same classes or terms (*ibid.*).

In this study we start from the assumption that manual classes in our data collection are correct, and compare results of automated classification against them. The classification system used in the study is Engineering Information (Ei), which is rather big (around 800 classes) and deep (five hierarchical levels), allowing many different choices. Without a thorough qualitative analysis of automatically assigned classes we cannot be sure if the classes assigned by the algorithm, which were not manually assigned, are actually wrong.

2.3 Description of the Algorithm

This study is based on an automated classification approach [2] that has been developed within the DESIRE project [6] to produce “All” Engineering [1], an experimental module of the manually created subject gateway Engineering Electronic Library (EELS) [7] (no longer maintained).

The algorithm classifies Web pages into classes of the Ei classification system. Mappings exist between the Ei classes and Ei thesaurus descriptors; both the captions of classes and the descriptors are matched against extracted title, headings, metadata, and main text of a Web page. Each time a match is found, the document is assigned the corresponding class, which is awarded a relevance score, based on which term is matched (single word, phrase, Boolean), the type of class matched (main or optional) (*weight[term]*), and the part of the Web page in which the match is found (*weight[loc]*). A match of a phrase (a number of words in exact order) or a Boolean expression (all terms must be present but in any order) is made more discriminating than a match of a single word; a main class is made more important than an optional class (in the Ei thesaurus, main class (code) is the class to use for the term, while optional class (code) is to be used under certain circumstances). A list of suggested classes and corresponding relevance scores (S) is produced using the following algorithm:

$$S = \sum_{locs} \left(\sum_{terms} (freq[loc_j][term_i] * weight[term_i] * weight[loc_j]) \right) . \quad (1)$$

Only classes with scores above a pre-defined cut-off value (cf. section 4.5) are selected as *the* classes for the document. Having experimented with different ap-

proaches for stemming and stop-word removal, the best results were gained when an expanded stop-word list was used, and stemming was not applied. For more information on the algorithm, see [2] and [13].

3 Methodology

3.1 Data Collection

The data collection used in the study comprises a selection of Web pages from the EELS subject gateway [7]. EELS Web pages have been selected and classified by librarians for end users of the gateway.

For the study, only pages in English were kept, the reason being that Ei captions and descriptors are in English. Also, some other pages were removed because they contained very little or no text. (The problem of pages containing hardly any text could be dealt with in the future, by propagating the class obtained for their subordinate pages.) The final data collection consisted of 1003 Web pages in the field of engineering.

The data were organized in a relational database. Each document in the database was assigned Ei classes derived from the following elements:

- title (Title);
- headings (Headings);
- metadata (Metadata); and,
- page's main text (Text).

Each class was automatically assigned a score indicating the degree of certainty that it is the correct one. Every document also had manually assigned Ei classes (Manual), against which the automatically assigned classes were compared.

3.2 Methods for Evaluation and Deriving Significance Indicators

Various measures have been used to evaluate different aspects of automated classification performance [21]. Effectiveness, the degree to which correct classification decisions have been made, is often evaluated using performance measures from information retrieval, such as precision and recall, and F1 measure being the harmonic mean of the two. Solutions have also been proposed to measure partial overlap, i.e. the *degree* of agreement between correct and automatically assigned classes (see, for example, [5]).

In this study, three methods have been used for evaluating and deriving the significance of different Web-page elements:

1. total and partial precision, recall, and F1 measures (using macroaveraging);
2. semantic distance; and,
3. multiple regression.

1. The Ei classification system has a solid hierarchical structure, allowing for a rather credible test on partial overlap. Three different levels of overlap were tested: total overlap; partial overlap of the first three digits, e.g. “932.1.” and “932.2.” are considered the same; and, partial overlap of the first two digits, e.g. “932” and “933” are considered the same. Partial overlap of the first four digits has not been conducted because there were few classes of five-digit length in the data collection.

2. In the literature, different similarity measures have been used for hypermedia navigation and retrieval (see, for example, [20]). Semantic distance, a numerical value representing the difference in meaning between two concepts or terms, is one of them. There are different ways in which to calculate it. For example, the measure of clicking distance in a directory-browsing tree can be used. We used the hierarchical structure of the Ei classification system as the means of obtaining the following (rather arbitrary) measures of semantic distance between any two classes:

- 4, when the classes differ already in the first digit (e.g. 601 vs. 901);
- 2, when the classes differ already in the second digit (e.g. 932 vs. 901);
- 1, when the classes differ in the third digit (e.g. 674.1 vs. 673.1); and
- 0.5, when the classes differ in the fourth digit (e.g. 674.1 vs. 674.2).

Those values reflect how the hierarchical system is structured; e.g. we say that class 6 and class 7 are more distant from each other than classes 63 and 64, which are in turn more distant in meaning than 635.1 and 635.2.

Calculations were conducted using the average distance between manually and automatically assigned classes. For each document, average distances were calculated for each of the four elements, and then the values were averaged for all the documents. When there was more than one manually assigned class per document, the semantic distance was measured between an automatically assigned class and that manually assigned class which was most similar to the automatically assigned one.

3. Multiple regression was used in a rather simplified way: scores assigned based on individual elements of a Web page were taken as independent variables, while the final score represented the dependent variable. The dependent variable was set to either 1000 or 0, corresponding to a correct or an incorrect class respectively.

4 Significance Indicators

4.1 General

In Table 1 basic classification characteristics and tendencies of our data collection are given. All the documents (1003) have at least one, and no more than six manually assigned classes, the majority having up to three classes. Manual assignment of classes was based on collection-specific classification rules.

Concerning automatically assigned classes based on different parts of a page, not all the pages have classes based on all of them. Classes based on text are assigned to the majority of documents, while those based on metadata to the least number of documents. Based on only title, headings, or metadata, less than 50% of the documents would get classified at all. On the average, per every document there are two

manually assigned classes, two classes based on title, four based on headings, nine based on metadata, and some 18 classes based on text.

In the whole collection there are 753 different classes assigned, either manually or automatically. The largest variety comes from the group of classes assigned based on text (675), which is more than twice as many as manually assigned (305).

Table 1. Distribution of classes in the data collection. First data row shows how many documents have been classified, second row how many classes have been assigned in the whole of the data collection, and the last row how many different individual classes, out of some 800 possible, have been assigned.

	Manual	Title	Headings	Metadata	Text
Number of classified doc.	1003	411	391	260	964
In the data collection	1943	827	1504	2227	17089
Different classes	305	174	329	406	675

4.2 Precision and Recall

Fig. 1. shows the degree of automated classification accuracy when words are taken solely from the four different parts of the Web page. While title tends to yield best precision, which is 27% more than the worst element (text), text gives the best recall, but only 9% more than the worst element (title). Precision and recall are averaged using the F1 measure, according to which title performs the best (35%), closely followed by headings (29%), metadata (21%) and text (15%).

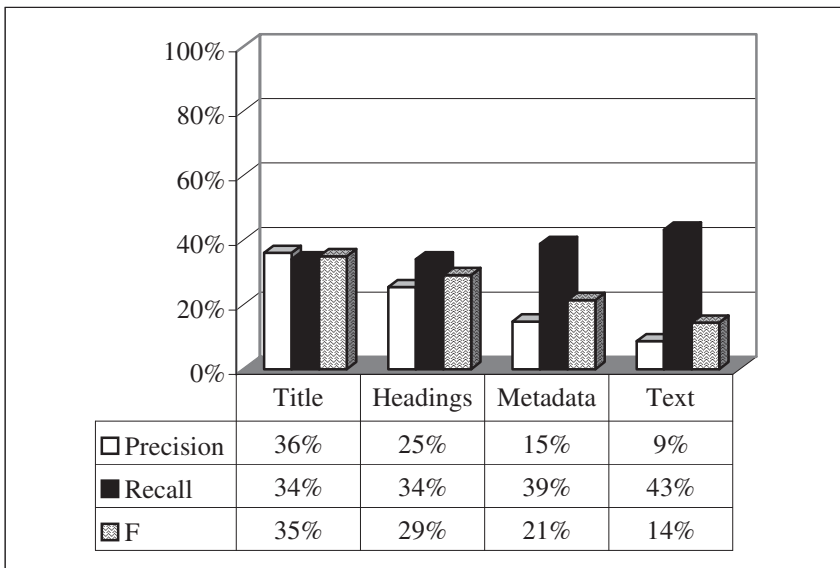


Fig. 1. Precision, recall and F1 measure

Partial Precision and Recall. When testing the algorithm performance for partial overlap (Fig. 2.), precision and recall for all parts of a Web page give much better results (title in 2-digit overlap achieves 59%). The ratio between their performance for both two- and three-digit overlap is the same as for total overlap: title performs the best, followed by headings, metadata and text.

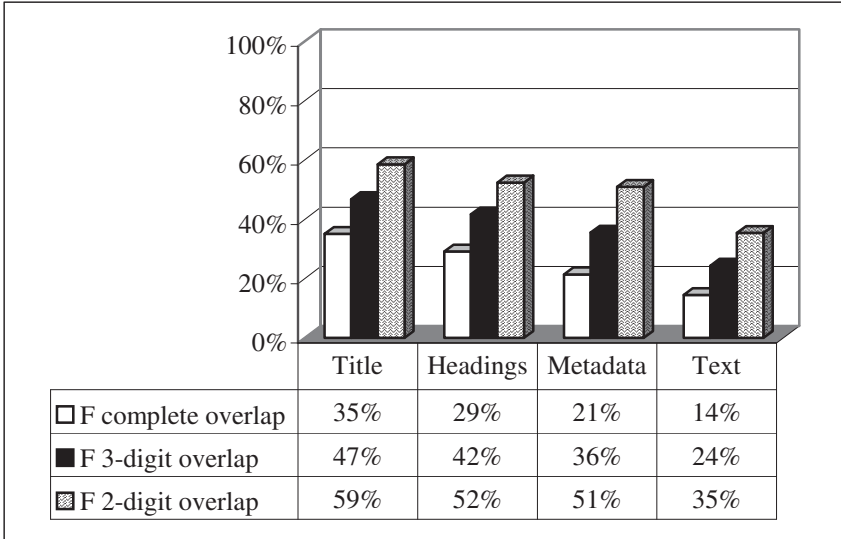


Fig. 2. F1-measure values for total overlap, 3-digit and 2-digit overlap

4.3 Semantic Distance

Using the semantic distance method, the calculations (Table 2) show that automatically assigned classes are on the average wrong in the third and second digits. Just like precision and recall results for partial overlap (cf. section 4.2), best results (smallest semantic distances) are achieved by title, followed by headings, metadata and text.

Table 2. Semantic distance

	Title	Headings	Metadata	Text
Mean distance	1,3	1,7	1,8	2,2

4.4 Deriving Significance Indicators

As we have seen in section 4.1, not every document has all the four elements containing sufficient terms for automated classification. Thus, in order to get documents classified, we need to use a combination of them. How to best combine them has been experimented with in this section, by applying results gained in evaluation using the F1 measure, semantic distance, and multiple regression.

The symbols used in formulae of this section are:

- S – final score for the automatically assigned class;
- ST_i – score for the automatically assigned class based on words in Title;
- SH – score for the automatically assigned class based on words in Headings;
- SM – score for the automatically assigned class based on words in Metadata; and,
- ST_e – score for the automatically assigned class based on words in Text.

The baseline, in which all the elements have equal significance, is represented with the following formula:

$$S = ST_i + SH + SM + ST_e . \quad (2)$$

Based on evaluation results, the following co-efficients, representing significance indicators, have been derived (the co-efficients were normalized by reducing the smallest co-efficient to one and by rounding others to integer values):

I. Based on total overlap and F1 measure values:

$$S = 2*ST_i + 2*SH + SM + ST_e . \quad (3)$$

These co-efficients have been derived by simply taking the F1 measure values of each of the algorithms (cf. Fig. 1). The same co-efficients have also been derived using partial overlap, the only difference being that the co-efficient for SM was two, both in two- and three-digit overlap.

II. Based on multiple regression, with scores not normalized for the number of words contained in title, headings, metadata, and text:

$$S = 86*ST_i + 5*SH + 6*SM + ST_e . \quad (4)$$

III. Based on multiple regression, with scores normalized for the number of words contained in title, headings, metadata, and text:

$$S = ST_i + SH + SM + 5*ST_e . \quad (5)$$

IV. On the basis of semantic distance results, the best significance indicator performs less than twice as well as the worst one, so all co-efficients are almost equal, as in (2).

4.5 Evaluation

Defining a Cut-Off. As described in section 2.3, each document is assigned a number of suggested classes and corresponding relevance scores. Only a few classes with best

scores, those above a certain cut-off value, are finally selected as *the* classes representing the document.

Different cut-offs, that would give best precision and recall results, were experimented with. Also, the number of documents that would be assigned at least one class, and the number of classes that would be assigned per document, were taken into consideration. Best results were achieved when the final classes selected were those with scores that contained at least 5% of all the scores assigned to all the classes, or, if such a class hadn't existed, the class with the top score was selected. In this case, F1 was 27%, there were about 4000 classes assigned as final, and all documents were classified. This is the cut-off we used in the study.

Results. As seen from Table 3, the evaluation showed that different significance indicators make hardly any difference in terms of classification algorithm performance. Co-efficients in (3) and (5) are similar to the ones in the baseline (2), and, compared to the baseline (2), which performs 23% in F1, normalized multiple regression (5) performs worse by 1%, while the formula based on F1 measure (3) performs the same. The best result was achieved using non-normalized multiple regression (4), which performs by 3% better than the baseline. This formula gives big significance indicator to classes that were assigned based on the title.

Table 3. Results of applying different co-efficients as significance indicators

	Baseline (2)	F1 (3)	Regression (4)	Regression N. (5)
Precision	16%	17%	21%	16%
Recall	39%	39%	35%	38%
F1	23%	23%	26%	22%
Number of pages	1003	1003	1003	1003
Number of classes	5174	5063	4073	5147

5 Conclusion

The aim of this study was to determine the significance of different parts of a Web page for automated classification: title, headings, metadata, and main text. The significance indicators were derived using several different methods: (total and partial) precision and recall, semantic distance, and multiple regression. The study showed that using *all* the structural elements and metadata is necessary since not all of them occur on every page. However, the exact way of combining the significance indicators turned out not to be highly important: the best combination of significance indicators is only 3% better than the baseline.

Reasons for such results need to be further investigated. One could guess that this is due to the fact that the Web pages in our data collection were rather heterogeneous; on the other hand, they were selected by librarians for end users of an operational service, and as such they might indicate what such Web-page collections are like. Apart from heterogeneity, the problem could be that metadata were abused, and that

certain tags were misused (e.g. instead of using appropriate tags for making text bold, one used a headings tag, which has the same effect on the screen).

Concerning evaluation of automated classification in general, further research is needed to determine the true value of the classification results. To that purpose information specialists and users could be involved, to compare their judgments as to which classes are correctly assigned. Also, in order to put the evaluation of classification into a broader context, a user study based on different information-seeking tasks would be valuable.

Other related issues of further interest include:

- determining significance of other elements, such as anchor text, location at the beginning of the document versus location at the end, etc.;
- comparing the results with new versions of the Web pages in the collection, e.g. maybe the quality of titles improves with time, and structural tags or metadata get less misused etc.; and,
- experimenting with other Web page collections.

Acknowledgements

The research was funded by ALVIS, an EU Sixth Framework Programme, Information Society Technologies (IST-1-002068-STP), and The Swedish Agency for Innovation Systems (P22504-1 A).

References

1. "All" Engineering Resources on the Internet: A Companion Service to EELS. Available: <http://eels.lub.lu.se/ae/> (2003)
2. Ardö, A., Koch, T.: Automatic Classification Applied to the Full-Text Internet Documents in a Robot-Generated Subject Index. In: Online Information 99, Proceedings of the 23rd International Online Information Meeting, London. (1999) 239-246
3. Attardi, G., Gulli, A., Sebastiani, F.: Automatic Web Page Categorization by Link and Context Analysis. In: Hutchison, C., Lanzarone, G. (eds.): Proceedings of THAI-99, European Symposium on Telematics, Hypermedia and Artificial Intelligence. (1999) 105-119
4. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training. In: Annual Workshop on Computational Learning Theory, Proceedings of the Eleventh Annual Conference on Computational Learning Theory. (1998) 92-100
5. Ceci, M., Malerba, D.: Hierarchical Classification of HTML Documents with WebClassII. In: ECIR. (2003) 57-72
6. DESIRE : Development of a European Service for Information on Research and Education. Available: <http://www.desire.org/> (2000)
7. Engineering Electronic Library. Available: <http://eels.lub.lu.se/> (2003)
8. Fisher, M., Everson R.: When are Links Useful?: Experiments in Text Classification. In: Proceedings of ECIR-03, 25th European Conference on Information Retrieval, Pisa, IT (2003) 41-56
9. Fürnkranz, J.: Hyperlink Ensembles: A Case Study in Hypertext Classification. Information Fusion 3, 4 (2002) 299-312

10. Ghani, R., Slattery, S., Yang, Y.: Hypertext Categorization Using Hyperlink Patterns and Metadata. In: Proceedings of ICML-01, 18th International Conference on Machine Learning. (2001), 178-185
11. Glover, E.J. et al.: Using Web structure for Classifying and Describing Web Pages. In: Proceedings of the Eleventh International Conference on World Wide Web Honolulu, Hawaii, USA. (2002) 562–569
12. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys* 3, 31 (1999) 264-323
13. Koch, T., Ardö, A.: Automatic Classification of Full-Text HTML-Documents from One Specific Subject Area. EU Project DESIRE II D3.6a, Working Paper 2. Available: <http://www.it.lth.se/knowlib/DESIRE36a-WP2.html>. (2000)
14. Kolcz, A., Prabakarmurthi, V., Kalita, J., and Alspector, J.: Summarization as Feature Selection for Text Categorization. In: Proceedings of the Tenth International Information and Knowledge Management (CIKM-01). (2001) 365-370
15. Olson, H.A., Boll, J.J.: Subject Analysis in Online Catalogs. 2nd ed. Libraries Unlimited, Englewood, Colorado (2001)
16. Pierre, J.: On the Automated Classification of Web sites. In: Linköping Electronic Articles in Computer and Information Science 001 (6) (2001). Available: <http://www.ep.liu.se/ea/cis/2001/001/>
17. Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Computing Surveys* 1, 34 (2002) 1–47
18. Slattery, S., Craven, M.: Discovering Test Set Regularities in Relational Domains. In: Proceedings of ICML-00, 17th International Conference on Machine Learning. (2000), 895-902
19. Svenonius, E.: The Intellectual Foundations of Information Organization. MIT Press, Cambridge, MA (2000)
20. Tudhope, D., Taylor C.: Navigation via Similarity: Automatic Linking Based on Semantic Closeness. *Information Processing and Management*, 33(2) (1997) 233-242
21. Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. *Journal of Information Retrieval* 1/2, 1 (1999) 67-88
22. Yang, Y., Slattery, S., Ghani, R.: A Study of Approaches to Hypertext Categorization. *Journal of Intelligent Information Systems*. 2/3, 8 (2002) 219-241

DL Meets P2P – Distributed Document Retrieval Based on Classification and Content

Wolf-Tilo Balke, Wolfgang Nejdl, Wolf Siberski, and Uwe Thaden

L3S and University of Hannover,
Expo Plaza 1, D-30539 Hannover, Telefax +49(0)511/762-9779
{balke, nejdl, siberski, thaden}@l3s.de

Abstract. Peer-to-peer architectures are a potentially powerful paradigm for retrieving documents over networks of digital libraries avoiding single points of failure by massive federation of (independent) information sources. Today sharing files over P2P infrastructures is already immensely successful, but restricted to simple metadata matching. But when it comes to the retrieval of complex documents, capabilities as provided by digital libraries are needed. Digital libraries have to cope with compound documents. Though some document parts (like embedded images) can efficiently be retrieved using metadata matching, the text-based information needs different methods like full text search. However, for effective querying of texts, also information like inverted document frequencies are essential. But due to the distributed characteristics of P2P networks such 'collection-wide' information poses severe problems, e.g. that central updates whenever changes in any document collection occur use up valuable bandwidth. We will present a novel indexing technique that allows to query using collection-wide information with respect to different classifications and show the effectiveness of our scheme for practical applications. We will in detail discuss our findings and present simulations for the scheme's efficiency and scalability.

1 Introduction

Digital libraries today offer a wide variety of content that is usually accessible through central portals. Though this generally is a useful way to access individual sources, it hampers the creation of federated networks of libraries or document collections. But such federations would be especially valuable for finding documents best matching the user's information needs and possibly providing a number of different views or opinions on a topic. One possibility of facilitating federated searches are meta-crawlers. But documents only accessible via a certain portal interface often cannot be crawled, also known as the 'hidden Web problem'. Moreover, crawlers only periodically crawl collections and update their indexes, and thus cannot react flexible to new, previously unknown collections and content changes in existing collections.

Peer to peer systems are a powerful paradigm to address some of these problems. Not relying on central coordination federations of information sources are formed dynamically by independent nodes. At any time new sources can join the network and disseminate their documents in a more timely way than the crawling of central servers can ascertain. File sharing applications, where media files are retrieved based on simple meta-data annotations like file formats or names, have become increasingly popular

due to their ease of use. Also digital library collections can benefit from the advantages of P2P infrastructures, since much of their content (like embedded images) can be annotated by meta-data.

However, compound documents in digital libraries generally also contain textual parts, where the flexible, distributed P2P paradigm still is not readily usable. On one hand this is because queries have to be evaluated over the network at search time, which in basic file sharing applications is usually done by flooding queries through the complete network (or at least within a certain radius). On the other hand almost all effective textual measures for information retrieval not only rely on statistics about the single documents, but also integrate statistics on the entire collection of all documents, e.g. how well individual keywords discriminate between documents with respect to the entire collection (inverted document frequencies). This so-called collection wide information cannot be derived locally.

To improve query efficiency techniques one way are central indexes and distributed hash tables (DHTs)[11,8,1]. Besides the speed up above naive query flooding an additional advantage is that by such a structure also collection wide information can be provided for subsequent querying. A major drawback is that such indexes use up a lot of the available bandwidth by the necessary administrative message exchange for upkeep, because every change in the federated document collection (e.g. content changes within some peer) has to be registered in the index. A contrasting way of gaining query efficiency are local routing indexes that avoid the overhead of constant index upkeep, but due to their local nature face problems with acquiring the necessary collection-wide information. Besides having to be efficient, querying schemes will also have to take into account that the information in digital libraries often is pre-structured. Libraries usually categorize documents following some standardized taxonomies, such that documents on similar topics might be distinguished e.g. by rather taking an economical or sociological point of view, etc. This structure is also used to sometimes resolve ambiguities of keywords.

In this paper we investigate the querying of federated library collections over a peer-to-peer network. But in contrast to central indexing schemes, our aim is to create a local indexing scheme that allows effective indexing with a minimum of management message overhead and even efficiently use collection-wide information. Moreover, we will exploit taxonomies to structure the individual collections and investigate how this pre-structuring interacts with the effectiveness of our local indexing scheme and how to deal with the trade-off between the total number of documents in each category and our respective index size. We will investigate the necessary message traffic, the quality of result sets (as opposed to the perfect results using a central index), and a number of other characteristics of our novel approach. Throughout this paper we will assume a strong cooperation between peers, e.g. in order to ensure consistency of ranking results, score values have to be calculated uniformly by all peers.

We will motivate our approach with the example of federated news collections. News items can also be compound documents and are usually categorized within certain topics like politics or sports. Since we want to focus on the textual retrieval, we use a collection of LA Times news articles from the TREC-5 collection for our evaluations. By assuming periodic changes of user interests we can also experiment with the arrival

or removal of complete corpora from our federation. Since queries in such practical applications usually form a Zipf distribution (i.e. considering the total set of queries very little queries are posed very often, whereas most queries are posed only once in a while), we will present extensive experiments for such a distribution. Given our innovative results, peer-to-peer networks are on the verge of forming efficient infrastructures for federations of digital libraries utilizing even collection-wide information without the communication overhead of central indexes.

2 Information Retrieval in a Distributed Environment

In large document collections information retrieval techniques are mandatory for efficient retrieval. Over centralized repositories these techniques have been investigated since the 70ies and work quite effective, e.g. using inverted file indexes for subsequent retrieval [4]. Maintaining these indexes, however, is a major problem in distributed systems, especially peer-to-peer networks that often share vast numbers of documents and have a high volatility with respect to peers joining and leaving the network. In contrast to static document collections every peer joining or leaving the network registers its document collection or removes it, thus indexes have to be updated very often.

For local query evaluation schemes a particular problem arises when collection wide information is an integral part of the query processing technique. For instance in the case of TFxIDF [16] the term frequency may be locally evaluated for each specific document, however, for the document frequency a snapshot of the entire current content of all active peers needs to be evaluated. Of course this would immediately annihilate any benefits gained by sophisticated local querying schemes.

Consider a simple example to show how local scorings fail, if collection-wide information has to be considered in the retrieval process. Assume the case that we have just two peers that should return their best matches with respect to the most popular information retrieval measure TFxIDF. This measure is a combination of two parts, the term frequency (TF, measures how often a query term is contained in a certain document), and the inverted document frequency (IDF, inverse of how often a query term occurs in the document collection). This measure needs to integrate collection-wide information and cannot be determined locally.

As an instance take a simple conjunctive query Q for the terms 'a' and 'b' posed to two peers P_1 and P_2 that has to be evaluated locally at each peer. Let's assume that P_1 contains three documents D_1, D_2 and D_3 , and P_2 also contains three documents D_4, D_5 and D_6 . For simplicity of our example let us further assume that in D_1 to D_6 , our two keyword occur mutually exclusive in the documents: D_1, D_2 and D_6 contain the keyword 'a', whereas D_3, D_4 and D_5 contain the keyword 'b'. Moreover, assume that all documents are of the same length and the keywords occur in the same number in all documents, such that the respective term frequency is the same for all documents. Evaluating our query Q locally we have now to rank the documents in each peer. Since the keywords are mutually exclusive in our document base and the TFs are equal for each document, the ranking is only determined by the weighting factor of the occurring term in the IDF.

In peer P_1 we have two documents out of three containing term 'a', i.e. an IDF of $\frac{3}{2} = 1.5$, and only one document containing 'b', i.e. an IDF of $\frac{3}{1} = 3$. That means that with respect to Q P_1 ranks D_3 as better than D_1 and D_2 . Symmetrically P_2 ranks D_6 higher than D_4 and D_5 , because here 'b' occurs in two documents and 'a' only in one. Integrating the results from P_1 and P_2 we get a higher ranking of D_3 and D_6 than of the four other documents. In contrast, performing query Q over a central collection containing all six documents D_1 to D_6 , we find that both query terms 'a' and 'b' occur in three of the six documents, i.e. have an IDF $\frac{6}{3} = 2$. Since the TF is still the same, all six documents will be correctly assigned the same score.

As shown, collection-wide information is essential to provide proper document scores. But the index holding this information does not necessarily need to be completely up-to-date; obviously there is a trade-off between index information that is 'still current enough' given the network volatility and the accuracy of the query results. Research on what dissemination level is required in Web IR applications to allow for efficient retrieval showed that a complete dissemination with immediate updates is usually unnecessary, even if new documents are included into the collection [14]. Moreover, the required level was found to be dependent on the document allocation throughout the network [13]: random allocation calls for low dissemination, whereas higher dissemination is needed if documents are allocated based on content. Thus a lazy dissemination usually has comparable effectiveness as a centralized approach for general queries, but if only parts of the networks containing most promising documents with similar content are queried, the collection-wide information has to be disseminated and regularly updated.

3 Approach

As shown in the previous section, we need collection-wide information at each peer to do a correct score computation. The challenges are

- how to compute this information
- where to store it, and
- how to distribute it in the network.

The key to success is the observation that we don't need a complete inverted index to process a query [7,2]. For example, to calculate the correct scores, peers P_1 and P_2 need only IDFs for terms a and b , but not for all terms occurring in their documents.

Storage. To store this information, we use a super-peer network approach: in a P2P-network peers often vary widely in bandwidth and computing power. As discussed in [17] exploiting these different capabilities can lead to an efficient network architecture, where a small subset of peers, called super-peers, takes over specific responsibilities. In our case, we assign the index management responsibility to the super-peers. The super-peers form the network backbone, and each document provider peer is directly connected to one of them.

Distribution. A query consists of a category of the taxonomy which should be searched and a conjunction of keywords that are searched in the content of the documents. Before distributing such a query, a super-peer adds the necessary collection-wide information from its index to it. If it isn't yet in the index, an estimation is provided.

The category in the query is used as a filter for two purposes: First to reduce the number of peers (and thus documents) which must be searched and ranked. Second the user can use the category to avoid ambiguities. If a user is interested in the local sport-team called 'Jaguars', the appropriate category will avoid hits Jaguar-cars and the animal Jaguar. The keywords which are specified in the query will only be used for the documents which are in the named category.

Computation. Responding peers do not only deliver matching documents, but also each add local data necessary to compute the collection-wide information (for TFxIDF this is the document frequency for each query term and the document count). On the way back to the originating super-peer, this data is aggregated. Thus, the originating super-peer gets everything it needs to compute the complete aggregate, and can store the computed result in its index.

The next subsections discuss in detail how this approach is applied to the digital library network context.

3.1 Query Processing

Query distribution at super-peer. Each super-peer maintains an IDF index containing IDF values for the keywords. This is done separately for each category, not for all documents in all categories. Thus, the key for this index is built from a category and one keyword. As mentioned above a query contains the IDF values for the query-terms. The IDFs are taken from the IDF index, or estimated if a keyword is not yet in the index. In the latter case, the average IDF is used as estimation.

Query processing at peer. At each peer, first only documents in the specified category are taken into account. The top- k documents are determined using the TFxIDF algorithm, but based on the IDF values from the query. If this gets enough ($=k$) results in the queried category, these are sent to the super-peer. If the number of documents matching the query is smaller than k , the query is relaxed, first to subcategories and then to the super-categories. This process is repeated until the peer has k results or the root of the taxonomy is reached. The entries found in all newly searched categories are sorted by their similarity to the originating category using the following measure (from Li et al. [5]):

$$\text{sim}(c_1, c_2) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$

where l is the shortest path between the topics in the taxonomy tree and h is the depth level of the direct common subsumer. α and β are parameters to optimize the similarity measurement (best setting is usually $\alpha = 0.2$ and $\beta = 0.6$).

The super-peer then gets the top- k of the peer or a number $n < k$ of documents matching. This query-relaxation is shown in the following code:

```

Initialize a ResultSet results;

Set searchRoot := Category from query

do
  Initialize a new set searchCategories
  Add searchRoot to searchCategories
  while (number of results < k and searchCategories is not empty)
  begin
    Initialize new set allChildren
    for all cat in searchCategories do
    begin
      // retrieve hits matching category exact
      Initialize ordered list matchingDocuments
      for all doc in documents
      begin
        if (document-category = cat
            AND document contains terms from query
            AND number of matching documents < k
            OR doc.score(query.terms) > matchingDocuments.getLastDoc.score)
          then add document to matchingDocuments
        end
        results.addHits(retrieveExact(cat, query))
        allChildren.add(cat.children);
      end
    end

    searchCategories = allChildren; // go one level down in category tree
    removesearchRoot from searchCategories; //do not to traverse subtree twice
  end

  searchRoot := parent of searchRoot // go one level up in category tree
  while(not k results AND searchRoot != nil);

  trim results to k // in case we collected more than k entries

return results;

```

Result merging at super-peer. A super-peer retrieves max. k hits from each of its peers and combines them to the top- k . As described above it is possible that peers also send results which are coming from another category as requested. In this case, the super-peer first takes all hits which match the queried category. If this results in a set smaller than k it takes the next best-matching hits from each peer and combines them using the described sorting.

IDF index update. The IDF index can be updated in two ways:

1. By summing up document frequencies and document counts delivered from connected peers and super-peers, the super-peer where the query originated computes IDF's for each query term and updates its IDF index. If the difference between computed IDF and estimated IDF value exceeds a threshold, the query is redistributed, this time using the computed IDF values.
2. if a super-peer receives a query it checks, if the IDF's contained are marked as estimated. If this is not the case, these values are used to update the IDF index.

3.2 IDF Index Entry Expiration

Viles and French have shown that in a large document collection IDF values change slowly [14]. In our context, this is not strictly applicable, because there are two kinds of changes that may influence our collection-wide information significantly:

1. *New documents with similar content: new peers join the network.*

Imagine a large federation of library servers which offer articles from different newspapers. Let's assume we already have a newspaper like the NY Times in the collection. What can happen if peers join the network offering a new newspaper, i.e. the LA Times? In this case we can be sure that the articles usually will be on nearly similar topics except a few local news. Thus, we do not really have to update our IDFs since the words in the articles are distributed the same way as before.

2. *New documents or new corpora: New library servers join the federation or new documents are included in existing collections, whose content is very different from existing articles and thus shifts IDFs and changes the discriminators.*

Let's look at an example: Assume there is an election e.g. in France and people use our P2P-news-network to search for news regarding this election. This normally will be done using queries like 'election France' and results in a list of news that contain these words. In this case there would be a lot of news containing France, thus 'election' is the discriminator, and the IDFs will give us the correct results. Now think of another election taking place in the US in parallel. The term 'election' will no longer be the best discriminator, but the term 'France' then gets more important.

In these cases we have to solve the problem that entries in the IDF index become outdated over time. We can handle both cases in the same way: Each IDF value gets a timestamp when the term appears for the first time and the term/IDF-pair is stored. After a specific expiration period (depending on the network-volatility) the item becomes invalid and the entry gets deleted. In this way we force IDF recomputation if the term occurs again in a query. By adjusting the expiration period we can trade off accuracy against performance. We reduce the expiration period for terms occurring more frequently, thus ensuring higher accuracy for more popular queries.

3.3 Query Routing Indexes

So far, we still distribute all queries to all peers. We can avoid broadcasting by introducing additional routing indices which are used as destination filters:

- For each category in our taxonomy the *category index* contains a set of all peers which have documents for this category. It is not relevant if this peers did contribute to queries in the past.
- In the *query index* for each posed query the set of those peers which contributed to the top-*k* for the query are stored.

Query Distribution. The super-peer first checks if all query terms are in the IDF index. If this is not the case the query has to be broadcast to permit IDF aggregation. We also broadcast the query if none of the routing indexes contain applicable entries.

If an entry for query exists in the query-index, it is sent to the peers in this entry only, since no other have contributed to the top- k result for the current query.

Otherwise, if the query category is in the category index, the query is sent to all peers to the corresponding category entry.

Index Update. For each delivered result set, a query index entry is created, containing all peers and super-peers which contributed to the result.

For the category index, we need to know all peers holding documents of the specified category, even if they didn't contribute to the result set. Therefore, we collect this information as part of the result set, too, and use it to create category index entries.

As with the IDF index, the network volatility causes our routing indexes to become incorrect over time. We use the index entry expiration approach here as well to delete outdated entries.

4 Evaluation

4.1 Simulation Environment

We use the TREC document collection volume 5 consisting of LA Times articles for our experiments. The articles are already categorized according to the section they appeared in, and we use this information as base for our document classification. To simulate a network of document providers, these articles are distributed among the peers in the network. The simulated network consists of 2000 peers, each providing articles from three categories on average (with a standard deviation of 2.0).

The simulation is based on the framework described in [10]. The super-peers are arranged in a HyperCuP topology [9]. The TFxIDF calculation based on inverse indexes was done using the (slightly modified) search engine Jakarta Lucene¹.

We assume a Zipf-distribution for query frequencies with skew of -0.0. News articles are popular only for a short time period, and the request frequency changes correspondingly. With respect to the Zipf-distribution this means that the query rank decreases over time. Query terms were selected randomly from the underlying documents. In our simulation, we generate 200 new most popular queries every 2000 queries which supersede the current ones and adjust query frequencies accordingly. This shift may be unrealistically high, but serves well to analyze how our algorithm reacts to such popularity changes.

4.2 Results

Index size. Figure 1 shows how the IDF index at each super-peer grows over time. After 10000 queries it has grown to a size of 2015, only a small fraction of all terms occurring in the document collection. A global inverted index we would have had contained 148867 terms. This underlines that much effort can be saved when only indexing terms which are actually appearing in queries.

¹ <http://jakarta.apache.org/lucene/docs/index.html>

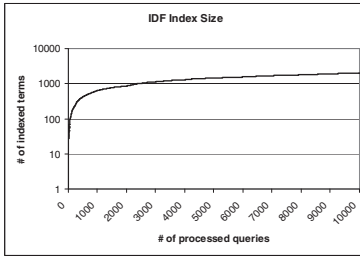


Fig. 1. Index size

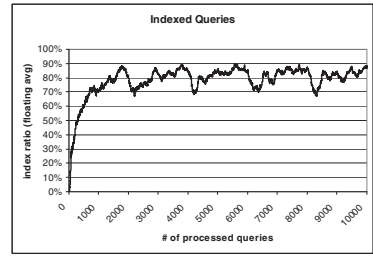


Fig. 2. Coverage of query index

Index effectivity. Both category and query index become quite effective. After nearly 2000 queries, the query index achieves a coverage of 80%. Figure 2 shows how each popularity shift causes a coverage reduction from which the query index recovers after about 1000 queries. This shows that a change in query popularity over time is coped with after a very short while.

As there are only about 120 different categories, after less than 1000 queries the index contains nearly all of them (Figure 3). We assume that news provider specialized on some topics change these topics only very infrequently. Therefore, peers do not shift their topics during the simulation. Thus, the category index serves to reduce the number of contacted peers continuously, also after popularity shifts.

Figure 4 shows how many peers had to be contacted to compute the result. The influence of popularity shifts on the whole outcome can also be seen clearly. The category index takes care that the peaks caused by popularity shifts don't become too high. Summarized, the combination of both indexes yields a high decrease of contacted peers compared to broadcasting.

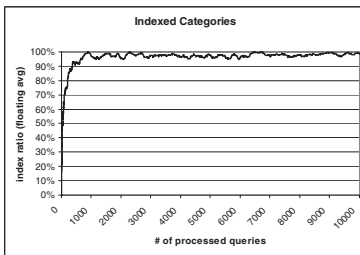


Fig. 3. Coverage of category index

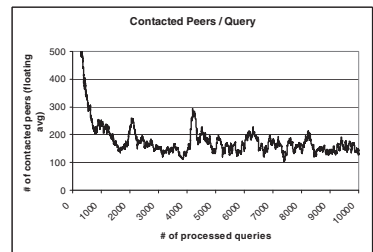


Fig. 4. Contacted peers per query

In the experiments described here we didn't introduce dynamics regarding the peers contents. Therefore, our algorithm yields exactly the same results as a complete index. In [2] (where we didn't take categories into account), we show that if 20% of the peers contents during a simulation run, the error ratio is about 3.5%.

5 Related Work

Since the concepts of the highly distributed P2P networks and the rather centralized IR engines are hard to integrate, previous work in the area is focussing on efficient dissemination of this information. There is a challenging trade-off between reduced network traffic by lazy dissemination however leading to less effective retrieval, and a large network traffic overhead by eager dissemination facilitating very effective retrieval. What is needed is "just the right" level of dissemination to maintain a "suitable" retrieval effectiveness. Thus previous approaches to disseminate collection-wide information rely on different techniques. We will briefly review the techniques from peer-to-peer systems, from distributed IR and Web Search engines and compare them to our approach.

For peer-to-peer systems there are different approaches. The PlanetP system [3] does not use collection-wide information like e.g. the inverted document frequency of query terms directly, but circumnavigates the problem by using a so-called inverted peer frequency estimating for all query terms, which peers are interesting contributors to a certain query. Summarizations of the content in the form of Bloom filters are used to decide what content a peer can offer, which are eagerly disseminated throughout the network by gossiping algorithms. Thus in terms of retrieval effectiveness this scheme describes documents on the summarization level, which is a suboptimal discriminator and by gossiping the system's scalability is limited. The idea of PeerSearch [12] is comparable to our approach, but instead of a broadcast-topology CAN [8] is used in combination with the vector space model (VSM) and latent semantic indexing (LSI) to create an index which is stored in CAN using the vector representations as coordinates. Thus all collection-wide information has to be disseminated again leading to a limited scalability. Also summarizing indexes have been used to maintain global information about a set of documents like e.g. in [15]. Here so-called cell abstract indexes are used for approximate queries. The abstract of a set of documents is some statistics of all documents in the set and the abstract of a peer is an abstract of the shared document set residing in the peer. An abstract index of a P2P system then is an organization of all abstracts of peers in the system. All peers of a system can thus be formed into an overlay network. Every joining peer will be added to a cell that contains its abstract and subsequently queries are routed to those cells that contain their abstract. However, also in this case indexes for all cells have to be updated regularly leading to a high overhead of network traffic. Moreover, peers in the end cells will just deliver all documents to the querying peer not removing suboptimal objects and again causing unnecessary network traffic. As in our approach, [6] use super-peers (called "hub" nodes) to manage indices and merge results. Depending on the cooperation capability/willingness of document providers ("leaf" nodes), hub nodes collect either complete or sampled term frequencies for each leaf peer. This information is used to select relevant peers during query distribution. By using query sampling hubs are able to give an estimate of relevant peers, even in case of uncooperative peers. As with the other systems, indices are built in advance, thus causing possibly unnecessary management messages.

From the perspective of information retrieval the problem of disseminating collection-wide information first occurred when IR moved beyond centralized indexing schemes over collections like e.g. given by TREC, and had to deal with vast distributed document collections like the WWW. Here due to the random-like distribution of

content over the WWW, research on effective retrieval in Web IR applications showed that a complete dissemination with immediate updates is usually unnecessary, thus allowing for a little volatility [14]. The required level of dissemination, however, was found to be dependent on the document allocation throughout the network [13]: random allocation calls for low dissemination, whereas higher dissemination is needed if documents are allocated based on content. In peer-to-peer networks this random-like distribution does usually not hold. We have argued in our news scenario, that in practical applications peers often will not carry a random portion of the entire document collection. Though some newspapers like the New York Times will cover a wide area of topics, specialized newspapers like the Financial Times will limit the range and some publications can even provide corpora that essentially differ in the topics and keywords contained. Moreover, though a lazy dissemination in terms of effectiveness usually is comparable to the centralized approach for general queries, our indexing scheme focuses only on parts of the networks containing most promising documents, thus the collection-wide information has to be disseminated and (at least) regularly updated. Hence, classical Web search engines like Google crawl the Web and individually index the sites, but then all indexing information is transferred over the network and managed in a vast centralized repository for subsequent retrieval. Novel approaches to distribute Web search engines like Google desktop will have to deal with the same problem of dissemination this information efficiently. Therefore, though designed for peer-to-peer infrastructures, our work here can be assumed to have an interesting impact on future developments in distributed Web search engines.

6 Conclusion and Further Work

In this paper we have discussed the important problem of efficiently querying federated library collections using peer-to-peer infrastructures especially if collection-wide information is needed. Though federations of libraries servers would benefit from more dynamic and easy to use the P2P paradigm, previous approaches were only able to support information searches based on exact meta-data matchings. This is because centralized indexing techniques have a high communication overhead, whereas local routing indexes cannot deal with collection-wide information. We have described a practical use-case scenario for the problem and have presented an innovative local indexing scheme which flexibly includes collection-wide information. Our novel indexes are not created in advance, but are maintained query-driven, i.e. we do not index any information which is never asked for. This allows our algorithm to scale, even in more volatile networks. Another improvement is our introduction of a separate category index that allows to prune large portions of the network and thus also enhances scalability.

To ensure result quality in networks of library collections, our future work will also need to consider issues as trust and reputation of individual servers. Super-peers can do consistency checks by continuously taking samples and subsequently exclude outliers from delivering results. Moreover, while in this paper we restricted our analysis on a unique classification scheme for the network, it also essential to support mapping between heterogeneous classifications.

References

1. K. Aberer. P-grid: A self-organizing access structure for p2p information systems. In *Proceedings of the Sixth International Conference on Cooperative Information Systems (CoopIS)*, Trento, Italy, 2001.
2. W.-T. Balke, W. Nejld, W. Siberski, and U. Thaden. Progressive distributed top-k retrieval in peer-to-peer networks. In *Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)*, 2005.
3. F. M. Cuenca-Acuna, C. Peery, R. P. Martin, and T. D. Nguyen. PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. In *Twelfth IEEE International Symposium on High Performance Distributed Computing (HPDC-12)*. IEEE Press, June 2003.
4. R. Korfhage. *Information Storage and Retrieval*. John Wiley, New York, 1997.
5. Y. Li, Z. A. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), 2003.
6. J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *European Colloquium on IR Research (ECIR 2005)*, 2005.
7. W. Nejld, W. Siberski, U. Thaden, and W.-T. Balke. Top-k query evaluation for schema-based peer-to-peer networks. In *Proceedings of 3rd International Semantic Web Conference (ISWC 2004)*, 2004.
8. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2001.
9. M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. HyperCuP—Hypercubes, Ontologies and Efficient Search on P2P Networks. In *International Workshop on Agents and Peer-to-Peer Computing*, Bologna, Italy, July 2002.
10. W. Siberski and U. Thaden. A simulation framework for schema-based query routing in p2p-networks. In *1st International Workshop on Peer-to-Peer Computing & DataBases(P2P& DB 2004)*, 2004.
11. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2001.
12. C. Tang, Z. Xu, and M. Mahalingam. Peersearch: Efficient information retrieval in peer-peer networks. Technical Report HPL-2002-198, Hewlett-Packard Labs, 2002.
13. C. L. Viles and J. C. French. Dissemination of collection wide information in a distributed information retrieval system. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*, pages 12–20. ACM Press, 1995.
14. C. L. Viles and J. C. French. On the update of term weights in dynamic information retrieval systems. In *Proceedings of the 1995 International Conference on Information and Knowledge Management (CIKM)*, pages 167–174. ACM, 1995.
15. C. Wang, J. Li, and S. Shi. Cell abstract indices for content-based approximate query processing in structured peer-to-peer data systems. In *APWeb*, volume 3007 of *Lecture Notes in Computer Science*. Springer, 2004.
16. I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes*. Morgan Kaufman, Heidelberg, 1999.
17. B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the 19th International Conference on Data Engineering (ICDE)*, 2003.

Automatic Detection of Survey Articles

Hidetsugu Nanba¹ and Manabu Okumura²

¹ Hiroshima City University, 3-4-1 Ozuka-higashi, Asaminami-ku
Hiroshima, 731-3194, Japan
nanba@its.hiroshima-cu.ac.jp

² Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku
Yokohama, 226-8503, Japan
oku@pi.titech.ac.jp

Abstract. We propose a method for detecting survey articles in a multilingual database. Generally, a survey article cites many important papers in a research domain. Using this feature, it is possible to detect survey articles. We applied HITS, which was devised to retrieve Web pages using the notions of authority and hub. We can consider that important papers and survey articles correspond to authorities and hubs, respectively. It is therefore possible to detect survey articles, by applying HITS to databases and by selecting papers with outstanding hub scores. However, HITS does not take into account the contents of each paper, so the algorithm may detect a paper citing many principal papers in mistake for survey articles. We therefore improve HITS by analysing the contents of each paper. We conducted an experiment and found that HITS was useful for the detection of survey articles, and that our method could improve HITS.

1 Introduction

Survey articles are defined as research papers, in which research in a specific subject domain is well organized and integrated. We can grasp the outline of the domain in a short time from them. However, how to detect them automatically from a huge number of research papers have not been discussed so far. We therefore study automatic detection of survey articles.

In our study, we pay attention to citation relationships between papers. Survey articles usually cite principal papers in the domain, and this feature can be used to detect them. We first detect principal papers in a domain, and then detect papers cite principal papers.

In this paper, we use the HITS algorithm [2], which ranks Web pages based on the link information among a set of documents. The HITS algorithm assumes two kinds of prominent or popular pages: authorities, which contain definitive high-quality information, and hubs, which are comprehensive lists of links to authorities. In academic literature, survey articles correspond to hubs, while papers initiating new ideas correspond to authorities, respectively. Survey articles should therefore be detected by applying the HITS algorithm to a research paper database, and selecting the papers with outstanding hub scores.

However, the HITS algorithm may also mistakenly detect papers that cite many related papers for a survey article, because the algorithm does not take account of the contents of each document. We therefore aim to detect survey articles with improved HITS algorithm, taking account of the contents of each paper.

In the remainder of the paper, Section 2 introduces the HITS algorithm and some related works. Section 3 describes our method for detecting survey articles. To investigate the effectiveness of our method, we conducted an experiment, described in Section 4. Section 5 reports the experimental results. Section 6 presents conclusions.

2 Related Work

Kleinberg proposed HITS [2], which is an algorithm to determine authoritative pages by an analysis of the link structure. The algorithm considers two kinds of pages: hubs, which are valuable as sources of good links, and authorities, which are valuable because many pages link to them. The algorithm determines authoritative pages in two stages: (1) constructing a focused sub-graph of the WWW, and (2) computing hub and authority scores of each page.

In the first stage, the t highest-ranked pages for the query σ are collected from a text-based search engine. These t pages are called a root set R . Here, t is typically set to about 200. Then, R is expanded into a base set S by adding all pages pointing to $r \in R$, and pointed to by r , to find authoritative pages that do not contain the query σ .

In the second stage, the following equations are applied to the sub-graph that was made in the first step, and then hub and authority scores of each page are then calculated.

$$x_p = \sum_{q \text{ such that } q \rightarrow p} y_q \quad (1)$$

$$y_p = \sum_{q \text{ such that } p \rightarrow q} x_q \quad (2)$$

where “ $q \rightarrow p$ ” means q links to p . The authority score of page x (x_p) is proportional to the hub scores of the pages linking to page p , and its hub score y_p is proportional to the authority scores of the pages to which it links. Again, power iteration is used to solve this system of equations.

Cohn and Chang [1] proposed the probabilistic HITS algorithm (PHITS), and applied it to a full-text citation index on the WWW called Cora¹, constructed by McCallum et al. [4]. The HITS algorithm was also applied to CiteSeer², which is another full-text citation index on the WWW, constructed by Lawrence et al. [3]. In both systems, full-text papers were classified into several categories automatically, and HITS or PHITS was adapted to the papers in each category. The papers in each category were sorted by their hub or authority scores. Though Cohn and Chang reported that PHITS is useful in identifying authoritative papers [1], the effectiveness

¹ <http://www.cs.umass.edu/~mccallum/code-data.html>

² <http://citeseer.ist.psu.edu/>

of using hubs to detect survey articles has not yet been examined. We therefore investigate this, and that our method can improve the HITS algorithm.

3 Detection of Survey Articles

We improve the HITS algorithm by taking account of the features of survey articles, and apply the improved algorithm to a research paper database to detect them.

Section 3.1 describes four features used for the improvement of HITS algorithm. Section 3.2 formulates our method incorporating the four features.

3.1 Features Used in Survey Detection

We show five features as follows.

Title of a Paper (WORD)

A good clue for detecting survey articles is the presence of particular phrases in their titles. Examples of such phrases are “survey,” “sabei (“survey”),” “review,” “rebyu (“review”),” “Trend,” “torendo (“trend”),” “state-of-the-art,” and “doukou (state-of-the-art).” We therefore double (w_{hub_1}) the hub scores of research papers if cue phrases are contained in their bibliographic information, and multiply (w_{auth_1}) authority scores by 0.5 in the opposite case.

Citation Types (CITATION TYPE)

Generally, there are few citations to base on other researchers’ theories or methods, because new methods or theories based on previous works are not usually proposed in survey articles. We therefore calculate r , the fraction of citations that are to other researchers’ theories or methods in a research paper, and multiply the hub scores by $\text{sig}(r)$ (w_{hub_2}), and multiply the authority scores of each paper by $\text{sig}(1/r)$ (w_{auth_2}), where $\text{sig}(x)$ is defined as $2/(1+\exp(1-x))$, which changes the range of the value x from 0.5 to 2. If r is zero, we set w_{auth_2} to two.

We use Nanba and Okumura’s method for determining the reasons for citations [5]. The method identifies the following citation types (reasons for citation) by analysing contexts of citations in research papers using several cue phrases, and obtains an accuracy of 83%.

- Type B: Citations to other researchers' theories or methods.
- Type C: Citations to compare with related work or to point out problems.
- Type O: Citations other than types B and C.

In our study, we use a database, which contains research papers written in both Japanese and English. As, Nanba and Okumura’s identification rules were developed for analysing English research papers, we developed rules for Japanese research papers in a similar manner to Nanba and Okumura’s rules.

Positional Deviation of Citations (DEVIATION)

Survey articles tend to cite related papers all through the articles, while other articles tend to cite them in particular sections, such as introduction and related work. We therefore take account of the positional deviation of citations in research papers. First, we count the number of sentences between citations (d_i). Second, we calculate the deviation of distances between citations using the following equation:

$$D = \sqrt{\frac{\sum_{i=1}^n (\bar{d} - d_i)^2}{n}} \times \frac{1}{text_len} \quad (3)$$

where \bar{d} is an average of all distances between citations. The positional deviation of citations (D) can be obtained by normalizing the standard deviation of distances between citations with the number of sentences ($text_len$) in the research paper. Score D increases as the deviation increases, while D approaches zero when a paper cites related papers at even intervals. We therefore multiply hub scores by $\text{sig}(D)$ (w_{hub_3}), while multiply authority scores by $\text{sig}(1/D)$ (w_{auth_3}).

Size of a Research Paper (SIZE)

Generally, survey articles are longer than others. We compare the length L_i (the number of sentences) of each paper with the average length \bar{L} , then multiply authority scores by $\text{sig}(L/\bar{L})$ (w_{auth_4}), while multiply hub scores by $\text{sig}(\bar{L}/L)$ (w_{hub_4}).

Cue Words (CUE)

Particular phrases, such as “this survey” and “we overview” (we call them positive cue phrases) often appear in survey articles, while phrases, such as “we propose” and “this thesis” (we call them negative cue phrases) do not. We therefore use the following positive and negative cue phrases for detecting survey articles. We double (w_{hub_5}) hub scores of research papers if they contain positive cue phrases, and multiply authority scores by 0.5 (w_{auth_5}) in the opposite case.

- Positive cues: “this survey,” “this review,” “this overview,”
“(honronbun | honkou)dewa...(gaikan | gaisetsu)suru (“In this survey, we overview”)”
- Negative cues: “this thesis,” “this dissertation,” “we propose,”
“teiansuru (“we propose”)”

3.2 Improvement of HITS Algorithm

Using the five features explained in Section 3.1, we improve the HITS algorithm. These features are taken into account by multiplying both the hub and authority scores of the HITS algorithm by the respective weights. The authority and hub scores of each paper are calculated by the following equations.

$$x_p = \prod_{j=1}^5 f(w_{auth_j}, L) \times \sum_{q \text{ such that } q \rightarrow p} y_q \quad (4)$$

$$y_p = \prod_{j=1}^5 f(w_{hub_j}, L) \times \sum_{q \text{ such that } p \rightarrow q} x_q \quad (5)$$

$$f(w, L) = \begin{cases} w \times L & (\text{if } w > 1) \\ w / L & (\text{if } w < 1) \\ 1 & (\text{if } w = 1) \end{cases} \quad (6)$$

where w_{auth_j} and w_{hub_j} indicate the five weights for authorities and hubs, respectively. Both authority and hub scores are normalized in each iteration by $\sqrt{\sum x_p^2}$ and $\sqrt{\sum y_p^2}$, respectively, in the same way as the HITS algorithm. $f(w, L)$ is a function to change the relative importance of each weight among all weights. Changing the values L of each feature and combination of five features, we identify the best combination and optimal weights.

4 Experiments

To investigate the effectiveness of our method, we conducted an experiment. In this section, we first describe the multilingual database used in our examination. Second, we explain the experimental method, and we then report the results.

4.1 Construction of a Bilingual Database

Recently, we have been able to obtain many full-text research papers on the WWW. In this study, we construct a multi-lingual database by collecting Postscript and PDF files on the WWW. We will briefly explain the method as follows;

(1) Collecting Research Papers on the WWW:

We collected Web pages using the Web search engines Google³ and goo⁴ with the combination of five key words (“gyoseki (“work”)” or “kenkyu (“study”)” or “publications”) and (“postscript” or “pdf”). Then we collected all Postscript and PDF files within depth two from each collected page.

(2) Conversion of Postscript and PDF Files into Plain Texts:

We convert Postscript and PDF files into plain texts using prescript⁵ and pdftotext⁶, respectively. A patch for prescript for Japanese was provided by Dr. Noriyuki Katayama of the National Institute of Informatics.

³ <http://www.google.com>

⁴ <http://www.goo.ne.jp>

⁵ <http://www.nzdl.org/html/prescript.html>

⁶ <http://www.foolabs.com/xpdf/>

(3) Analysing the Structure of Research Papers:

We remove lists of references at the ends of files using cue words, such as “sanko bunken (“references”), “References,” and “Bibliography.” Next, we detect the positions of citations using patterns of citation (e.g., 1), (1), [1]). We also extract bibliographic information (a title and authors) within the first five sentences in each paper.

(4) Identification of Citation Relationships Between Papers:

We identify the duplication of bibliographic information extracted in step (3) for analysing whole citation relationships among papers in a database. For each pair of bibliographic records, we compare n-grams in one bibliographic record with those in the other, and count the number of matches. These matches are position-independent. If the number of matches is above a threshold, we consider the pair to be duplicates. We use a value of six for n in English texts, and three in Japanese texts.

(5) Extraction of Citation Information:

Citation types are identified based on several rules using cue phrases [5].

Finally, a bilingual research paper database was constructed. The database includes about 20,000 full-text papers (2,100 Japanese papers and 17,900 English papers) and 296,000 bibliographic references in the domain of computer science, nuclear biophysics, chemistry, astronomy, material science, electrical engineering, and so on.

4.2 Experimental Methods**Alternatives**

We conducted experiments using the following nine methods.

- Our methods
 - WORD, CITATION TYPE, SIZE, DEVIATION and CUE: combination of HITS and the named features.
 - COMB: combination of HITS and five features.
- Baseline
 - HITS: original HITS algorithm
 - BASE-WORD: research papers containing particular words, which were used for WORD, in their titles.
 - BASE-CUE: research papers containing particular cue phrases, which were used for CUE.

As we described in Section 3.2, we change L for each feature manually from zero (the feature is not used) to 10^9 in consideration of the very large range of the hub scores, and identify the optimal values and combinations.

Test Collection

In the same way as the original HITS algorithm, we prepare several base sets using some key phrases, and apply our methods to each set. The procedure to select key phrases was as follows:

1. Apply n -gram analysis to a list of bibliographic records;
2. Select 39 key phrase candidates manually, by checking the list of frequently used expressions from step 1;
3. Collect all bibliographic information including key phrases, and make a root set R for each key phrase candidate;
4. Collect all bibliographic information u that has citation relationships with any $r \in R$, and form a base set S by integrating them with R ;
5. Eliminate key phrases for which S contains very few full-text papers;
6. Select the remaining candidates as key phrases.

We also took account of the variation of research domains. We finally obtained 20 key phrases, all of them are in English.

We then identified survey articles in a base set S of each key phrase. It is necessary to look through all the papers in S to obtain all the survey articles, but this is impossible if

Table 1. Key phrases, size of S , and the number of survey articles

Topics (Key phrases)	Number of bibliographic items (S)	Number of full-text papers	Number of survey articles
applied mathematics	2988	893	16
astronomy	1068	127	6
computer architecture	3280	1097	13
computer graphics	3324	656	14
constraint programming	689	217	5
database systems	3902	1077	33
data mining	2183	403	16
discrete mathematics	1353	299	12
distributed systems	4260	1279	24
high energy	1173	263	11
knowledge engineering	625	186	14
logic programming	4195	1081	30
mathematical physics	1268	199	9
operating systems	4509	1378	21
parallel processing	2776	1118	25
pattern recognition	3319	1054	34
robotics	6374	1246	30
spectroscopy	400	91	3
symbolic computation	753	330	9
wavelet	2641	457	10
Averages	2443.0	646.3	16.2

the set is very large. We therefore used a pooling method [6], known as a method for the construction of large-scale test collections. In this method, a pool of possible relevant documents is created by taking a sample of documents selected by various IR systems. Then human assessors judge the relevance of each document in the pool. We examined the top-ranked 100 documents (full-text papers) from each of eight methods.

We show the 20 key phrases, the size of each base set S , the number of full-text papers in each S , and the number of survey articles in Table 1.

Evaluation Measure

We believe that when more survey articles in a domain are detected, it becomes more efficient for users to grasp the outline of the domain, because the survey articles may be written from different viewpoints, and comparison of such viewpoints is useful for deep understanding and taking a broad view of the domain. We therefore detect as many survey articles as possible.

Eleven-point Recall-Precision is the most typical evaluation measure in the IR community. We evaluate our systems by 11-point R/P using Equations (6) and (7). We also evaluate our system by the precisions of top-ranked documents, because survey articles are written for quickly grasping the outline of the domain, and should be detected in higher ranks. For the calculation of recall and precision, we made use of “trec_eval” (<ftp://ftp.cs.cornell.edu/pub/smart>), which is an evaluation tool developed for Text REtrieval Conference (TREC).

$$\text{Recall} = \frac{\text{The number of survey articles correctly detected by a system}}{\text{The number of survey articles that should be detected}} \quad (7)$$

$$\text{Precision} = \frac{\text{The number of survey articles correctly detected by a system}}{\text{The number of survey articles detected by a system}} \quad (8)$$

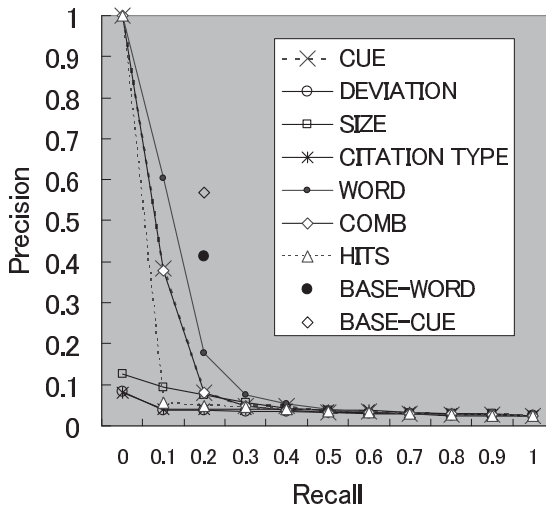
4.3 Results

We optimized values of L for each method to give the best precisions of top-ranked documents. The results are shown in Table 2. Using these values, we evaluated our systems by 11-point Recall-Precision, and by the precisions of the top-ranked documents. Both results are shown in Fig. 1 and Table 3, respectively.⁷ The most striking result in Fig 1 is that WORD produces results that are remotely useful than those of other methods at recall = 0.2. CUE is second best, and both WORD and CUE improved the HITS algorithm. SIZE could also improve HITS when Recall is more than 0.1. DEVIATION and CITATION TYPE made HITS worse. In the evaluation by precisions of top-ranked documents (Table 3), both COMB and CUE are much superior to the others. We can also confirm that CUE and WORD could improve HITS significantly.

⁷ As BASE-WORD and BASE-CUE collect all papers containing particular words (or cue phrases), and do not rank the results, we randomly ranked each result, and calculated the precision scores of both methods in Table 3.

Table 2. The optimal values L of each method

Method	Values of L
CUE	15000-20000
DEVIATION	10^8
SIZE	1000
CITATION TYPE	10^5
WORD	10^7
COMB	CUE: 18000 DEVIATION: 0 SIZE: 0 CITATION TYPE: 10 WORD: 0

**Fig. 1.** Evaluation by 11-point Recall/Precision**Table 3.** Evaluation by precisions of top-ranked documents

Top -n	Our methods						Baseline		
	Single feature					Multiple features COMB	HITS	BASE- WORD	BASE- CUE
	CUE	DEVI- ATION	SIZE	CITE TYPE	WORD				
5	1.000	0.000	0.000	0.000	0.800	1.000	0.400	0.414	0.568
10	1.000	0.000	0.100	0.000	0.500	1.000	0.200	0.414	0.568
15	1.000	0.000	0.067	0.000	0.533	1.000	0.133	0.414	0.568
20	0.950	0.000	0.050	0.000	0.550	1.000	0.100	0.414	0.568
30	0.700	0.033	0.100	0.000	0.633	0.700	0.100	-	0.568
100	0.350	0.070	0.050	0.050	0.540	0.350	0.100	-	0.568

4.4 Discussion

Effectiveness of the HITS Algorithm

From the results in Fig. 1, we can find that BASE-CUE has an outstanding ability to detect survey articles by itself. However, CUE could never obtain precision scores of 1.0 at top-5, 10, and 15 without the HITS algorithm, because BASE-CUE detected non-relevant papers at rates up to 43.2%. In other words, HITS could exclude non-relevant documents from the result of BASE-CUE. We therefore conclude that the HITS algorithm is effective in detecting survey articles.

A List of Cue Phrases

As we could not prepare enough survey articles to apply statistical methods (e.g., n-gram) for the selection of cue phrases, we could only make a list of cue phrases. Fortunately, we found that our list of cue phrases was effective, although it may not be exhaustive. In future, we can add other cue phrases by applying statistical methods to survey articles that are collected automatically using our proposed method “COMB.”

Parameter Tuning

We could not confirm the effects of SIZE in the evaluation by precisions of top-ranked documents, though the precision scores of SIZE are superior to HITS, when the recall score is more than 0.1 in Fig. 1. We could not tune parameters of five features finely, because of the processing time. If we spent the time to examine the parameters more closely, we may confirm the effectiveness of SIZE.

5 Conclusions

In this paper, we proposed a method for detecting survey articles from a multilingual research paper database. We considered HITS, which is an algorithm to retrieve Web pages using the notions of authority and hub. It is considered that important papers and survey articles correspond to authorities and hubs, respectively. It is therefore possible to detect survey articles by applying the HITS algorithm to research paper databases, and selecting papers with outstanding hub scores. However, as HITS does not take account of the contents of each paper, the algorithm might detect papers citing many principal papers in mistake for survey articles. We therefore improved HITS by incorporating five features of survey articles. To investigate the effectiveness of our method, we conducted an experiment. We found that the HITS algorithm was useful for the detection of survey articles. We also found that cue phrases (CUE) could improve the HITS algorithm, and performed better than other methods.

6 Future Work

As the next step of this study, we need to measure the qualities of survey articles and to select the best one among detected candidates. Although it is confirmed in the experiment that our method is useful to detect comprehensive survey articles, the method does not guarantee that there are good-quality comments about the referring

papers, and the task to measure the quality of such comments automatically seems very difficult. However, this problem may be resolved without analyzing survey articles by using NLP techniques. The limited resolution of this issue is to take account of the number of citations from other papers. Good-quality survey articles that contain good-quality comments are considered to be cited from many papers in the subject domain, and to have high authority scores. In our future work, we will investigate with the relations between qualities of survey articles and their authority scores.

Acknowledgements

The authors would like to express our gratitude to anonymous reviewers for their suggestions to improve our paper. This work was supported in part by JSPS (Japan Society for the Promotion of Science) under the grant for Postdoctoral Fellowship.

References

- [1] Cohn, D. and Chang, H. *Learning to probabilistically identify authoritative documents*. In Proceedings of the 17th International Conference on Machine Learning, pp.167–174, 2000.
- [2] Kleinberg, J.M. *Authoritative sources in a hyperlinked environment*. In Proceedings of the 9th Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 668–677, 1998.
- [3] Lawrence, S., Giles, L., and Bollacker, K. *Digital libraries and autonomous citation indexing*. IEEE Computer, 32(6), pp. 67–71, 1999.
- [4] McCallum, A., Nigam, K., Rennie, J. and Seymore, K. *Building domain-specific search engines with machine learning techniques*. In Proceedings of AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace. 1999.
- [5] Nanba, H. and Okumura, M. *Towards multi-paper summarization using reference information*. In Proceedings of the 16th International Joint Conferences on Artificial Intelligence, pp. 926–931, 1999.
- [6] Sparck Jones, K. and Van Rijsbergen, C.J. *Report on the need for and provision of ‘ideal’ test collections*. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

Focused Crawling Using Latent Semantic Indexing - An Application for Vertical Search Engines

George Almpantidis, Constantine Kotropoulos, and Ioannis Pitas

Aristotle University of Thessaloniki, Department of Infomatics,
Box 451, GR-54124 Thessaloniki, Greece
{galba, costas, pitas}@aiaa.csd.auth.gr
<http://www.aiaa.csd.auth.gr>

Abstract. Vertical search engines and web portals are gaining ground over the general-purpose engines due to their limited size and their high precision for the domain they cover. The number of vertical portals has rapidly increased over the last years, making the importance of a topic-driven (focused) crawler evident. In this paper, we develop a latent semantic indexing classifier that combines link analysis with text content in order to retrieve and index domain specific web documents. We compare its efficiency with other well-known web information retrieval techniques. Our implementation presents a different approach to focused crawling and aims to overcome the size limitations of the initial training data while maintaining a high recall/precision ratio.

1 Introduction

Within the last couple of years, search engine technology had to scale up dramatically in order to keep up with the growing amount of information available on the web. In contrast with large-scale engines such as Google [1], a search engine with a specialised index is more appropriate to services catering for specialty markets and target groups because it has more structured content and offers a high precision. Moreover, a user visiting a vertical search engine or portal may have a priori knowledge of the covered domain, so extra input to disambiguate the query might not be needed [2]. The main goal of this work is to provide an efficient topical information resource discovery algorithm when no previous knowledge of link structure is available except that found in web pages already fetched during a crawling phase. We propose a new method for further improving targeted web information retrieval (IR) by combining text with link analysis and make novelty comparisons against existing methods.

2 Web Information Retrieval

The expansion of a search engine using a *web crawler* is seen as task of *classification* requiring supervised automatic categorisation of text documents into specific and predefined categories. The visiting strategy of new web pages usually characterises the purpose of the system. Generalised search engines that seek to cover as much proportion of the web as possible usually implement a *breadth-first* (BRFS) or *depth-first*

search (DFS) algorithm [3]. The BRFS policy is implemented by using a simple FIFO queue for the unvisited documents and provides a fairly good bias towards high quality pages without the computational cost of keeping the queue ordered [4]. Systems on the other hand that require high precision and targeted information must seek new unvisited pages in a more intelligent way. The crawler of such a system is assigned the task to automatically classify crawled web pages to the existing category structures and simultaneously have the ability to further discover web information related to the specified domain. A *focused* or *topic-driven crawler* is a specific type of crawler that analyses its crawl boundary to find the links that are likely to be most relevant for the crawl while avoiding irrelevant regions of the web. A popular approach for focused resource discovery on the web is the *best-first search* (BSFS) algorithm where unvisited pages are stored in a priority queue, known as frontier, and they are reordered periodically based on a criterion. So, a typical topic-oriented crawler performs keeps two queues of URLs; one containing the already visited links (from here on **AF**) and another having the references of the first queue also called *crawl frontier* (from here on **CF**) [5]. The challenging task is ordering the links in the CF efficiently. The importance metrics for the crawling can be either interest driven where the classifier for document similarity checks the text content and popularity or location driven where the importance of a page depends on the hyperlink structure of the crawled document.

2.1 Text Based Techniques in Web Information Retrieval

Although the physical characteristics of web information is distributed and decentralized, the web can be viewed as one big virtual text document collection. In this regard, the fundamental questions and approaches of traditional IR research (e.g. term weighting, query expansion) are likely to be relevant in web document retrieval [6]. The three classic models of text IR are probabilistic, Boolean, and vector space model (VSM). The language independent VSM representation of documents has proved effective for text classification [7]. This model is described with indexing terms that are considered to be coordinates in a multidimensional space where documents and queries are represented as binary vectors of terms. Various approaches depend on the construction of a term-by-document two-dimensional $m \times n$ matrix A where m is the number of terms and n is the number of documents in the collection. We present an extension of the classic method that can be used as classifier in focused crawling in Sect 3.2.

2.2 Link Analysis Techniques

Contrary to text-based techniques, the main target of link analysis is to identify the *importance* or *popularity* of web pages. This task is clearly derived from earlier work in bibliometrics academic citation data analysis where prestige (“impact factor”) is the measure of importance and influence. More recently, link and social network analysis have been applied to web hyperlink data to identify authoritative information sources [8]. In the web, the impact factor corresponds to the ranking of a page simply by a tally of the number of links that point to it, also known as *backlink* (BL) count or *in-degree*. But BL can only serve as a rough, heuristic based, quality measure of a document, because it can favour universally popular locations regardless of the specific query topic.

PageRank (PR) is a more intelligent connectivity-based page quality metric with an algorithm that recursively defines the importance of a page to be the weighted sum of its backlinks' importance values [9]. An alternative but equally influential algorithm of modern hypertext IR is HITS, which categorises web pages to two different classes; pages rich and relevant in text content to the user's query (*authorities*) and pages that might not have relevant textual information but can lead to relevant documents (*hubs*) [10]. Hubs may not be indexed in a vertical engine as they are of little interest to the end user, however both kind of pages can collaborate in order to determine the visit path of a focused crawler.

2.3 Latent Semantic Indexing and SVD Updating

Latent semantic indexing (LSI) is a concept-based automatic indexing method that models the semantics of the domain in order to suggest additional relevant keywords and to reveal the "hidden" concepts of a given corpus while eliminating high order noise [11]. The attractive point of LSI is that it captures the higher order "latent" structure of word usage across the documents rather than just surface level word choice. The dimensionality reduction is typically computed with the help of Singular Value Decomposition (SVD), where the eigenvectors with the largest eigenvalues capture the axes of the largest variation in the data. In LSI, an approximated version of A , denoted as $A_k = U_k S_k V_k^T$, is computed by truncating its singular values keeping only the $k = \text{rank}(A_k) < k_0 = \text{rank}(A)$ larger singular values and their associated left and right eigenvectors are used for retrieval.

Unfortunately, the practical application of matrix decompositions such as SVD in dynamic collections is not trivial. Once an index is created it will be obsolete when new data (terms and documents) is inserted to the system. Adding new pages or modifying existing ones also means that the corpus index has to be regenerated for both the recall and the crawling phase. Depending on the indexing technique followed, this can be a computationally intensive procedure. But there are well-known relatively inexpensive methods such as fold-in and SVD updating that avoid the full reconstruction of the term-by-document matrix [12]. *Folding-in* is based on the existing latent semantic structure and hence new terms and documents have no effect on the representation of the pre-existing terms and documents. Furthermore, the orthogonality in the reduced k -dimensional basis for the column or row space of A (depending on inserting terms or documents) is corrupted causing deteriorating effects on the new representation. *SVD-updating*, while more complex, maintains the orthogonality and the latent structure of the original matrix [12].

3 Focused Crawling

3.1 Related Works in Focused Crawling

Numerous techniques that try to combine textual and linking information for efficient URL ordering exist in the literature. Many of these are extensions to PageRank and HITS. HITS does not work satisfactorily in cases where there is a mutually reinforcing

relationship between hosts (nepotism) [13]. An algorithm where nodes have additional properties and make use of web page content in addition to its graph structure is proposed. An improvement to HITS is probabilistic HITS (PHITS), a model that has clear statistical representations [14]. An application of PageRank to target seeking crawlers improves the original method by employing a combination of PageRank and similarity to the topic keywords [15]. The URLs at the frontier are first sorted by the number of topic keywords present in their parent pages, then they are sorted by their estimated PageRanks. The applicability of a BSFS crawler using PageRank as the heuristic is discussed in [16] and its efficiency against another crawler based on neural networks is tested. In [17] an interesting extension to probabilistic LSI (PLSI) is introduced where existing links between the documents are used as features in addition to word terms. The links can take the form of hyperlinks as is the case of HTML documents or they can take the form of citations, which is the case of scientific journal articles in citation analysis. The hypothesis is that the links contribute to the semantic context of the documents and thereby enhance the chance of successful applications. Two documents having a similar citation pattern are more likely to share the same context than documents with different citation patterns. An intelligent web crawler is suggested based on a principle of following links in those documents that are most likely to have links leading to the topic at interest. The topic is represented by a query in the latent semantic factor space. [18] proposes supervised learning on the structure of paths leading to relevant pages to enhance target seeking crawling. A link-based ontology is required in the training phase. Another similar technique is reinforcement learning [19] where a focused crawler is trained using paths leading to relevant goal nodes. The effect of exploiting other hypertext features such as segmenting Document Object Model (DOM) tag-trees that characterise a web document and propose a fine-grained topic distillation technique that combines this information with HITS is studied in [20]. Keyword-sensitive crawling strategies such as URL string analysis and other location metrics are investigated in [21]. An intelligent crawler that can adapt online the queue link-extraction strategy using a self-learning mechanism is discussed in [22]. Work on assessing different crawling strategies regarding the ability to remain in the vicinity of the topic in vector space over time is described in [23]. In [24] different measures of document similarity are evaluated and a Bayesian network model used to combine linkage metrics such as bibliographic coupling, co-citation, and companion with content-based classifiers is proposed. [25] also incorporates linking semantics additional to textual concepts in their work for the task of web page classification into topic ontologies. [26] uses tunnelling to overcome some of the limitations of a pure BSFS approach.

3.2 Hypertext Combined Latent Analysis (HCLA)

The problem studied in this paper is the implementation of a focused crawler for target topic discovery, given unlabeled (but known to contain relevant sample documents) textual data, a set of keywords describing the topics and no other data resources. Taking into account these limitations many sophisticated algorithms of the Sect. 2.2, such as HITS and context graphs, cannot be easily applied. We evaluate a novel algorithm called *Hypertext Content Latent Analysis* or **HCLA** from now onwards that tries to combine text with link analysis using the VSM paradigm. Unlike PageRank, where simple

eigen-analysis on globally weighted adjacency matrix is applied and principal eigenvectors are used, we choose to work with a technique more comparable with HITS. While the effectiveness of LSI has been demonstrated experimentally in several text collections yielding an increased average retrieval precision, its success in web connectivity analysis has not been as direct. There is a close connection between HITS and LSI/SVD multidimensional scaling [27]. HITS is equivalent to running SVD on the hyperlink relation (source, target) rather than the (term, document) relation to which SVD is usually applied. As a consequence of this equivalence, a HITS procedure that finds multiple hub and authority vectors also finds a multidimensional representation for nodes in a web graph and corresponds to finding many singular values for AA^T or $A^T A$, where A is the adjacency matrix. The main problem is that LSI proves inefficient when the dimensions of the term-document matrix A are small. But in the classification process of un/semi-supervised learning systems the accuracy of LSI can be enhanced by using unlabelled documents as well as labelled training data.

Our main assumption is that terms and links in an expanded matrix are both considered for document relevance. They are seen as *relationships*. In the new space introduced, each document is represented by both the terms it contains and the similar text and hypertext documents. This is an extension of the traditional “bag-of-words” document representation of the traditional VSM described in Sect. 2.1. Unlike [17], we use LSI instead of PLSI. The proposed representation, offers some interesting potential and a number of benefits. First, text only queries can be applied to the enriched relationships space so that documents having only linking information, such as those in CF, can be ordered. Secondly, the method can be easily extended for the case where we also have estimated content information for the documents in CF. This can be done using the anchor text or the neighbour textual context of the link tag in the parent’s html source code, following heuristics to remedy for the problem of context boundaries identification [16]. Moreover, we can easily apply local weights to the terms/rows of the matrix, a common technique in IR that can enhance LSI efficiency. While term weighting in classic text IR is a kind of linguistic favouritism, here this can also be seen as a method of emphasizing either the use of linking information or text content. An issue in our method is the complexity of updating the weights in the expanded matrix, especially when a global weighting scheme is used. For simplicity, we do not use any weighting scheme here. Let A be the original term-document representation while $\begin{pmatrix} L_{m \times a} \\ G_{a \times a} \end{pmatrix}$ and $\begin{pmatrix} O_{m \times b} \\ R_{a \times b} \end{pmatrix}$ are the new document vectors projected in the expanded term-space having both textual (submatrices $L_{m \times a}$ and $O_{m \times b}$ and linking connectivity components (submatrices $G_{a \times a}$ and $R_{a \times b}$). The steps of our method are depicted in Fig. 1 and are described as follows.

- With a given text-only corpus of m documents and a vocabulary of n terms we first construct a text-document matrix $A_{m \times n}$ and perform a truncated Singular Value Decomposition $A_k = SVD(A, k)$. Since this is done during the offline training phase an effort in finding the optimum k is highly suggested.

- After a sufficient user-defined number of pages (a) have been fetched by the crawler, we analyse the connectivity information of the crawler’s current web graph and

$$\begin{array}{l}
 m \text{ word terms} \\
 \mathbf{C} = \\
 a \text{ new outlinks} \\
 \text{from web} \\
 \text{documents in AF}
 \end{array}
 \left\{ \begin{array}{l}
 \left(\mathbf{A}_{m \times n} \right) \\
 \left(\mathbf{O}_{a \times n} \right)
 \end{array} \right\}
 \left(\begin{array}{c|c}
 \mathbf{L}_{m \times a} & \mathbf{O}_{m \times b} \\
 \mathbf{G}_{a \times a} & \mathbf{R}_{a \times b}
 \end{array} \right)$$

Fig. 1. Expanded connectivity matrix in HCLA. Matrix C is $[(m+a) \times (n+a+b)]$. AF=Already Fetched links, CF=Crawl Frontier docs

insert $a = |AF|$ new rows as “terms” (i.e. documents from AF) and $a+b = |AF|+|CF|$ web pages from both AF and CF as “documents” to the matrix. We perform the SVD-updating technique to avoid reconstructing the expanded index matrix. Because the matrices G and R in Fig. 1 are sparse, the procedure is simplified and the computation is reduced. We want to insert $t = a$ terms and $d = a + b$ documents, so we append submatrix $D_{(m+a) \times (a+b)} = \begin{pmatrix} L_{m \times a} & 0_{m \times b} \\ G_{a \times a} & R_{a \times b} \end{pmatrix}$ to $B_{[(m+a) \times n]} = \begin{pmatrix} A_{m \times n} \\ 0_{a \times n} \end{pmatrix}$ which is the new space after inserting terms from the AF.

- Because we do not have any information of direct relationship between any of these web pages and the text documents $\{d_i\}$ of the original corpus, we simply add a terms/rows at the bottom of the matrix A_k with zero elements. This allows the re-computing of SVD with minimum effort, by reconstructing the term-document matrix. If $SVD(A, k) = U_k S_k (V_k)^T$ is the truncated SVD of the original matrix A , and $SVD(B) = U_B S_B (V_B)^T$ the k -SVD of the matrix after inserting a documents, then we have:

$$U_B = \begin{pmatrix} U_{m \times k} \\ 0_{a \times k} \end{pmatrix}, S_B = S_k, V_B = V_k \tag{1}$$

The above step does not follow the SVD-updating technique since the full term-document matrix is recreated and a k -truncated SVD of the new matrix B is recomputed. In order to insert fetched and unvisited documents from the AF and CF queues as columns in the expanded matrix we use an SVD-updating technique to calculate the semantic differences introduced in the column and row space. If we define $SVD(C) = U_C S_C V_C^T$, $F = (S_k | U_B^T D)$ and $SVD(F) = U_F S_F V_F^T$ then, matrices U_C , S_C and V_C are calculated according to [12]:

$$V_C = \begin{pmatrix} V_B & 0 \\ 0 & I_{a+b} \end{pmatrix} V_F, \quad S_C = S_F, \quad U_C = U_B V_F \tag{2}$$

Accordingly, we project the driving original query q in the new space that the expanded connectivity matrix C represents. This is done by simply appending a rows of zeroes to the bottom of the query vector: $q_C = \begin{pmatrix} q_{m \times 1} \\ 0_{a \times 1} \end{pmatrix}$. By applying the driving query q_C of the test topic we are able to compute a total ranking of the expanded matrix

C. Looking at Fig. 1 we deduce that we only need to rank the last $b = |CF|$ columns. The scores of each document in CF are calculated using the cosine similarity measure:

$$\cos \theta_j = \frac{e_j^T V_C S_C (U_C^T q_C)}{\|S_C V_C^T e_j\|_2 \|q_C\|_2} \quad (3)$$

where $\|\cdot\|_2$ is the L_2 norm. Once similarity scores are attributed to documents, we can reorder the CF, select the most promising candidate and iterate the above steps.

4 Implementation – Experimental Results - Analysis

In this work we evaluate five different algorithms. BRFS is only used as a baseline since it does not offer any focused resource discovery. The rest are cases of BSFS algorithms with different CF reordering policies. The 2nd algorithm is based on simple BL count [21]. Here the BL of a document v in CF is the current number of documents in AF that have v as an outlink. The 3rd algorithm (SS1) is based on the Shark-Search algorithm, a more aggressive variant of Fish-Search [28]. The 4th algorithm (SS2) is similar to SS1 but the relevance scores are calculated using a pre-trained VSM that uses a probability ranking based scheme [7]. Since we work with an unlabelled text corpus, we use the topic query to extract the most relevant documents and use them as sample examples to train the system. The 5th algorithm is based on PageRank. Here, no textual information is available, only the connectivity between documents fetched so far and their outlinks. A known problem is that pages in CF do not have known outlinks since they have not been fetched and parsed yet. In order to achieve convergence of the PR we assume that from nodes with no outlinks we can jump with probability one to every other page in the current web graph. In this application, the exact pagerank values are not as important as the ranking they induce on the pages. This means that we can stop the iterations fairly quickly even when the full convergence has not been attained. In practice we found that no more than 10 iterations were needed. The 6th algorithm (HCLA) is the one this paper proposes. In the training phase choosing $k = 50$ for the LSI of the text corpus (matrix A) yielded good results.

The fact that the number of public available datasets suitable for combined text and link analysis is rather limited denotes the necessity of further research efforts in this field. In our experiments we used the WebKB corpus [29]. This has 8275 (after eliminating duplicates) web documents collected from universities and manually classified

Table 1. WebKB Corpus topic queries

Category	Topic keywords
course	course, university, homework, lesson, assignment, lecture, tutorial, book, schedule, notes, grading, handout, teaching, solutions, exam
faculty	faculty, university, professor, publications, papers, research, office
project	project, university, demonstration, objective, overview, research, laboratory
student	student, university, interests, favourite, activities, graduate, home

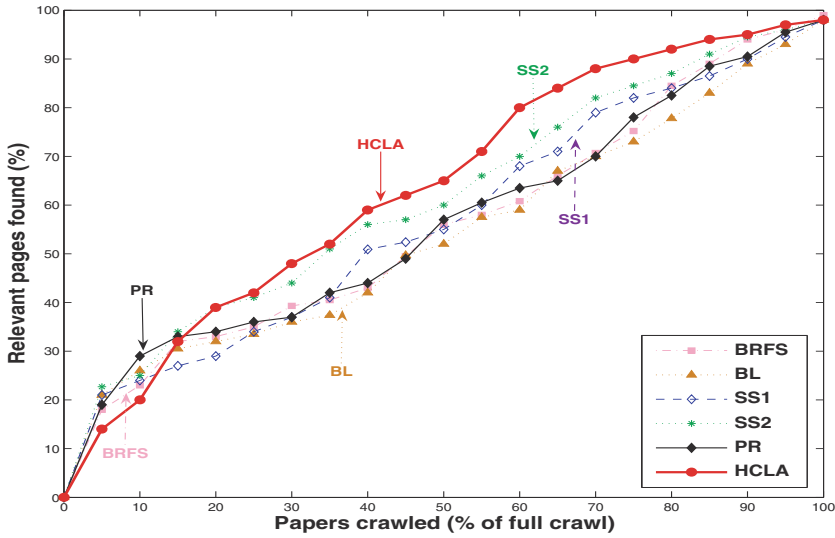


Fig. 2. Algorithm performance for WebKB

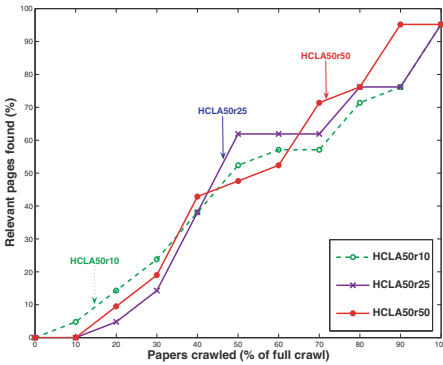


Fig. 3. HCLA performance for category project and university washington of WebKB for different BFSN strategies. HCLA50r10 means we use $k = 50$ features for LSI analysis and reorder the CF every $N = 10$ documents

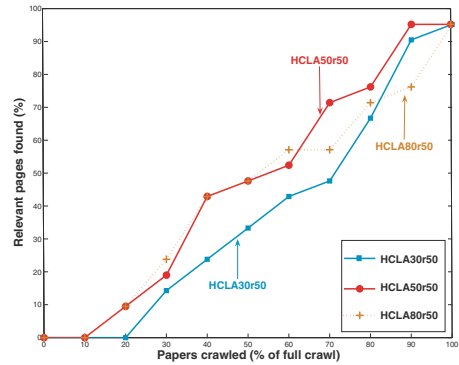


Fig. 4. HCLA performance for category project and university washington of WebKB for different BFSN strategies. HCLA50r10 means we use $k = 50$ features for LSI analysis and reorder the CF every $N = 10$ documents.

in 7 categories. For algorithms SS1, SS2, HCLA we selected each time three universities for training the text classifier and the fourth for testing. Documents from the “misc” university were also used for HCLA since the larger size of the initial text corpus can enhance the efficiency of LSI. Although the WebKB documents have link information we disregarded this fact in the training phase and choose to treat them only as textual data but for the testing phase we took into account both textual and linking information. The keyword-based queries that drive the crawl are also an indicative description of

each category. These were formed by assigning 10 people the task of retrieving relevant documents for each category using Google and recording their queries. In each case as seeds we considered the root documents in the “department” category. This entails the possibility of some documents being unreachable nodes in the vicinity tree by any path starting with that seed, something that explains the $< 100\%$ final recall values in Fig. 2, 4 and 4. Categories having relatively limited number of documents (e.g. “staff”) were not tested. We repeated the experiments for each category and for every university. Evaluation tests measuring the overall performance were performed by calculating the average ratio of relevant pages retrieved out of the total ground-truth at different stages of the crawl. Due to the complexity of PR and HCLA algorithms we chose to follow a BSFSN strategy, applying the reordering policy every N documents fetched for all algorithms (except BRFS). This is supported by the results of [30] which indicate that explorative crawlers outperform their more exploitive counterparts. We experimented with values of $N = 10, 25, 50$. The preprocessing involved fixing HTML errors, converting text encoding and filtering out all external links (outlinks that are not found inside the corpus), stemming [31], and a word stoplist for both the train and test text documents.

The results in Fig. 2 depict the superiority of our method especially at higher recall ranges. We must also consider that in our implementation we didn’t use term weighting, which is argued to boost LSI performance [11]. BRFS performance matched or exceeded in some cases SS1 and BL. This can be attributed to the structure of the WebKB corpus and the quality of the seed documents. The unimpressive results of PR justify the assertion that it is too general for use in topic-driven tasks due to its minimal exploitation of the topic context [16], [23]. In a BSFS strategy it is crucial that the time needed for reorganising the crawl frontier is kept at a minimum. According to [32], the best algorithms for SVD computation of an $m \times n$ matrix take time that is proportional to is $O(P \cdot m^2 \cdot n + Q \cdot n^3)$ (P and Q are constants which are 4 and 22 for a Riemannian SVD algorithm (R-SVD)). This means that the performance of a LSI-based BSFS crawler suffers when new documents and terms are inserted in each iteration. In our work, we do not need to recompute the SVD of the highly dimensional matrix C , but perform calculations on the reduced matrices of Sect. 3.2. Also, we follow a BSFS-N algorithm where the reordering of the CF, and consequently the term-by-document matrix expansion and SVD computation, are performed every N documents fetched. Naturally, value N has a significant influence in the processing time of the algorithm and the efficiency of the reordering analysis [30]. For the results presented here it is $N = 50$. From Fig. 4 we deduce that reordering the CF in higher frequency does not necessarily yield better results. A parameter not well documented is the choice of k (number of important factors) in LSI. While trial and error offline experiments can reveal an optimum value for the text corpus (matrix A), there is no guarantee this will remain optimal for the expanded matrix C . In Fig. 4 we see that selecting too many features can have in fact deteriorating results.

5 Conclusions

This work has been concerned with a statistical approach to text and link processing. We argue that content- and link-based techniques can be used for both the classifier and the

distiller of a focused crawler and propose an alternative document representation where terms and links are combined in an LSI based algorithm. A positive point in our method is that its training is not dependent on a web graph using a previous crawl or an existing generalised search service but only on unlabeled text samples making the problem a case of unsupervised machine learning. Because LSI performance is sensitive to the size of the trained corpus performance can suffer severely when little data is available. Therefore, starting a crawl with a small text-document matrix A is not recommended since at early stages the extra linking-text information from the crawl is minimal. Appending extra text documents in the training phase, even being less relevant to the topics of the current corpus, can enhance the crawling process. At later stages when more information is available to the system these documents can be removed and the model retrained. We also believe that a hybrid strategy where HCLA is facilitated in the early stages of the crawl by a more explorative algorithm can be a practical alternative.

The question remains whether the extra performance gain justifies the complexity it induces in the development of a focused web crawler. Both HCLA and PR methods proved significantly slower requiring more processor power and memory resources. Practically, HCLA was up to 100 times slower than the simple BRFS on some tests and PR performed similarly, something that has been attested by [16]. The dynamic nature of the crawler means that computational complexity increases as more documents are inserted in AF and CF. A solution to the problem is to limit the size of both queues and discard less authoritative or relevant docs at the bottom of the queues during the reordering phase. Another idea worth exploring in the future is using a “folding-in” technique instead of SVD-updating during the reorganisation step of HCLA to reduce the complexity of the algorithm.

[33] also proposes an expanded adjacency matrix that allows for different weighting schemes in different directions and explores the use of eigen-analysis in the augmented matrix. There, not only term-document similarity is modelled but also term-term and document-document. It will be interesting to apply the assumption of word-link semantic equivalence in this representation of web documents. As a first step we can expand the original term-document matrix $A_{m \times n}$ during training by considering the documents as terms, i.e. add n rows to the bottom of A . In the new column vector space, a document is represented as a bag of both terms and citations (outlinks). The significance of this representation will be realised when there is link connectivity previous knowledge between documents available, for example when deploying an incremental crawler. This can lead to semantically richer query definition.

References

1. Google Search Technology. Online at <http://www.google.com/technology/index.html>
2. R. Steele, “Techniques for Specialized Search Engines”, in Proc. *Internet Computing*, Las Vegas, 2001.
3. S. Chakrabarti, M. Berg, and B. Dom, “Focused crawling: a new approach to topic-specific Web resource discovery”, *Computer Networks*, vol. 31, pp. 1623-1640, 1999.
4. M. Najork and J. Wiener, “Breadth-first search crawling yields high-quality pages”, in Proc. 10th *Int. World Wide Web Conf.*, pp. 114-118, 2001.

5. A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan, "Searching the Web", *ACM Transactions on Internet Technology*, vol. 1, no. 1, pp. 2-43, June 2001.
6. K. Yang, "Combining text- and link-based methods for Web IR", in Proc. 10th *Text Retrieval Conf. (TREC-10)*, Washington 2002, DC: U.S. Government Printing Office.
7. G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing". *Communications of the ACM*, vol. 18, no. 11, pp. 613-620, 1975.
8. A. Ng, A. Zheng, and M. Jordan, "Stable algorithms for link analysis", in Proc. *ACM Conf. on Research and Development in Information Retrieval*, pp. 258-266, 2001.
9. S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine", *WWW / Computer Networks*, vol. 30, no. 1-7, pp.107-117, 1998.
10. J. Kleinberg, "Authoritative sources in a hyperlinked environment", in Proc. 9th *Annual ACM-SIAM Symposium Discrete Algorithms*, pp. 668-677, Jan. 1998.
11. M. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*, Philadelphia, PA: Society of Industrial and Applied Mathematics, 1999.
12. G. O'Brien, *Information Management Tools for Updating an SVD-Encoded Indexing Scheme*. Master's thesis, University of Tennessee, Knoxville, TN. 1994.
13. K. Bharat and M. Henzinger, "Improved algorithms for topic distillation in hyperlinked environments", in Proc. *Int. Conf. Research and Development in Information Retrieval*, pp. 104-111, Melbourne (Australia), August 1998.
14. D. Cohn and H. Chang, "Learning to probabilistically identify authoritative documents", in Proc. 17th *Int. Conf. Machine Learning*, pp. 167-174, 2000.
15. P. Srinivasan, G. Pant, and F. Menczer, "Target Seeking Crawlers and their Topical Performance", in Proc. *Int. Conf. Research and Development in Information Retrieval*, August 2002.
16. M. Chau and H. Chen, "Comparison of three vertical search spiders", *Computer*, vol. 36, no. 5, pp. 56-62, 2003.
17. D. Cohn and T. Hoffman, "The Missing Link-A probabilistic model of document content and hypertext connectivity", *Advances in Neural Information Processing Systems*, vol. 13, pp. 430-436, 2001.
18. M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori, "Focused crawling using context graphs", in Proc. 26th *Int. Conf. Very Large Databases (VLDB 2000)*, pp. 527-534, Cairo, 2000.
19. J. Rennie and A. McCallum, "Using reinforcement learning to spider the Web efficiently", in Proc. 16th *Int. Conf. Machine Learning (ICML99)*, pp. 335-343, 1999.
20. S. Chakrabarti, "Integrating the Document Object Model with hyperlinks for enhanced topic distillation and information extraction", in Proc. 10th *Int. World Wide Web Conf*, pp. 211-220, Hong Kong, 2001.
21. J. Cho, H. G. Molina, and L. Page, "Efficient Crawling through URL Ordering", in Proc. 7th *Int. World Wide Web Conf.*, pp. 161-172, Brisbane, Australia 1998.
22. C. Aggarwal, F. Al-Garawi, and P. Yu, "Intelligent Crawling on the World Wide Web with Arbitrary Predicates", in Proc. 10th *Int. World Wide Web Conf.*, pp. 96-105, Hong Kong, 2001.
23. F. Menczer, G. Pant, M. Ruiz, and P. Srinivasan. "Evaluating topic-driven web crawlers", in Proc. *Int. Conf. Research and Development in Information*, pp. 241-249, New Orleans, 2001.
24. P. Calado, M. Cristo, E. Moura, N. Ziviani, B. Ribeiro-Neto, and M.A. Goncalves, "Combining link-based and content-based methods for web document classification", in Proc. 12th *Int. Conf. Information and Knowledge Management*, pp. 394-401, New Orleans, Nov. 2003.
25. I. Varlamis, M. Vazirgiannis, M. Halkidi, and B. Nguyen, "THESUS: Effective thematic selection and organization of web document collections based on link semantics", *IEEE Trans. Knowledge & Data Engineering*, vol. 16, no. 6, pp. 585-600, 2004.

26. D. Bergmark, C. Lagoze, and A. Sbityakov, "Focused Crawls, Tunneling, and Digital Libraries", in Proc. 6th *European Conf. Research and Advanced Technology for Digital Libraries*, pp. 91-106, 2002.
27. S. Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*. San Francisco, CA: Morgan Kaufmann Publishers, 2002.
28. M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, "The shark-search algorithm. An Application: tailored Web site mapping", *Computer Networks and ISDN Systems*, vol. 30, pp. 317-326, 1998.
29. CMU World Wide Knowledge Base and WebKB dataset.
Online at <http://www-2.cs.cmu.edu/~webkb>
30. G. Pant, P. Srinivasan, and F. Menczer, "Exploration versus exploitation in topic driven crawlers", in Proc. 2nd *Int. Workshop Web Dynamics*, May, 2002.
31. M. Porter, "An algorithm for suffix stripping". *Program*, vol. 14 no. 3, pp. 130-137, 1980.
32. G. Golub and C. Van Loan. *Matrix Computations*, 3/e. Baltimore: Johns Hopkins University Press, 1996.
33. B Davison. "Unifying text and link analysis", in Proc. *IJCAI-03 Workshop Text-Mining & Link-Analysis (TextLink)*, Acapulco, August 9, 2003.

Active Support for Query Formulation in Virtual Digital Libraries: A Case Study with DAFFODIL*

André Schaefer, Matthias Jordan, Claus-Peter Klas, and Norbert Fuhr

University of Duisburg-Essen
{a.schaefer, mjordan, klas, fuhr}@uni-duisburg.de

Abstract. DAFFODIL is a front-end to federated, heterogeneous digital libraries targeting at strategic support of users during the information seeking process. This is done by offering a variety of functions for searching, exploring and managing digital library objects. However, the distributed search increases response time and the conceptual model of the underlying search processes is inherently weaker. This makes query formulation harder and the resulting waiting times can be frustrating. In this paper, we investigate the concept of proactive support during the user's query formulation. For improving user efficiency and satisfaction, we implemented annotations, proactive support and error markers on the query form itself. These functions decrease the probability for syntactical or semantical errors in queries. Furthermore, the user is able to make better tactical decisions and feels more confident that the system handles the query properly. Evaluations with 30 subjects showed that user satisfaction is improved, whereas no conclusive results were received for efficiency.

1 Introduction

It is a well known fact, that query formulation for digital libraries is a difficult task. While web search is mostly based on ranked retrieval, most digital library interfaces and OPACs (Online Public Access Catalogues) offer field-based boolean search interfaces. The average user has to learn the conceptual implications of this search modality and has to understand the underlying vocabulary problems. In *virtual* digital libraries, which search in a distributed way over several underlying digital libraries, this conceptual burden tends to increase.

The digital library system DAFFODIL¹ is such a virtual digital library, targeted at strategic support of users during the information search process. For searching, exploring, and managing digital library objects it provides user-customisable information seeking patterns over a federation of heterogeneous digital libraries. Searching with DAFFODIL makes a broad range of information

* The project DAFFODIL was funded by the German Science Foundation (DFG) as part of the research program "Distributed Processing and Delivery of Digital Documents"

¹ <http://www.daffodil.de>

sources easily accessible and enables quick access to a rich information space. However, the response times are well above that of single digital libraries, since DAFFODIL acts as meta-search engine for the 14 digital libraries currently included, so the result can only be shown and ranked after the slowest digital library answered the user's query. In such a situation, the problem is aggravated if the query is syntactically or semantically incorrect, therefore retrieving no results and wasting expensive time.

DAFFODIL already provides a broad range of tools to help the user in query formulation, e.g. a thesaurus, a spell-checker, a classification browser, and a related-term service. However, all these tools are currently separated, and are not integrated in the context of the query formulation (cf. [1]). The previous system version offered automatic suggestions of search terms that were shown in a separate window, a feature generally well-liked by users. However, our own findings in evaluations of DAFFODIL [2] indicated, that actually using terms provided in such a manner constitutes a high cognitive burden for many searchers as this feature was underutilized. Thus, in order to improve the usability of DAFFODIL, we decided to bring the suggestion more to the focal point of the interaction.

As a solution to reduce the workload and cognitive work of users, mixed initiative and interface agent-based approaches have been proposed in [3,4,5] for other application areas. In this paper, we want to investigate the question if these approaches can be applied to query formulation for searching in digital libraries. It is well known that in information seeking tasks, the user feels uncertain, has an ill-defined goal and a very limited attention span. Thus, the mental costs of pushed suggestions may be counterproductive in this situation.

These considerations led us to a user-oriented iterative redesign of the query form in the user interface of DAFFODIL [6]. The specific goal was to provide a mixed-initiative query formulation interface, which should help the user

1. to make fewer low-level errors,
2. decrease the cognitive load in early stages of the query reformulation cycles,
3. increase confidence in the constructed query,

and therefore increase user efficiency and effectiveness.

In the remainder of the paper we will first describe the problem in more detail and provide a brief overview on related work, followed by a description of our concept for proactive user support. Then we present the iterative design and evaluation process for integrating this feature into DAFFODIL in a user-friendly way. In addition, we evaluate the extension of the mixed-initiative approach for suggesting tactical or strategic moves and actions. A summary and an outlook conclude this paper.

2 Problem Description and Related Work

When users attempt to instantiate *known items* by specifying known facts about the documents, e.g. author and time range, the field-based boolean modality suits

the users' needs quite well. But in user tests [7,2] it is observed that users still need an average of four to five queries to find the information they were looking for. One major cause of the need for repeated querying is faulty queries. Most of these "errors" fall into the category of misspellings or typographical errors. Sometimes problems with query syntax lead to unexpected results.

The situation becomes even more complicated in the case of *topical search*. If the topic is not well known and the user needs to explore the information space, she is immediately hit by the vocabulary problem, e.g. *which search terms give the desired results, which related terms or synonyms are needed here*. This adds to the uncertainty the user already experiences due to his *anomalous state of knowledge* [8] that led to the topical search in the first place.

The psychology behind these problems [9] is well known and existing models of information seeking behaviour [10,11,12,13] explain the situation quite well. The users start with a high degree of uncertainty and learn about their original need during the iterations of their query reformulations and result inspections. Low level errors, like spelling mistakes or inadequate boolean logics, can be explained as an expression of uncertainty or fear when starting a search.

Bates [10] proposed a medium level of system support where the user is supported doing her task, but still remains in control. She differs between four levels of search activities: MOVE, TACTIC, STRATAGEM and STRATEGY. While moves can be any activity associated with searching, tactical activities try to drive the search in a specific direction, e.g. BROADER or NARROW. A medium level of system involvement means that the system should suggest corrections and tactical term additions or substitutions. Additionally activities outside the search form could be suggested, e.g. doing a journal run or computing a coauthor network for a more efficient strategy.

Together with Bates' concepts of system support, where users enter search terms, it makes much sense to present corrections, mark errors and suggest search terms or other possible moves right from the beginning of a search cycle[14,15]. Other systems suggest mostly terms for query *reformulation*. In our approach, we want to increase the system support significantly, by helping the user already in the important step of initial query formulation, thus decreasing the user's high uncertainty at the beginning of the search. Since the *Anomalous State of knowledge* (ASK) (cf. [8]) problem has been recognised, it is clear that term suggestions are a useful addition to information systems' search interfaces. Schatz et al. [14] have demonstrated and analysed the usefulness of term suggestions based on a subject thesaurus and term co-occurrence list. Brajnik et al.[15] have conducted a case study to investigate the value of query reformulation suggestions, terminological and strategic help, and the best way to provide them. *User-controlled interaction* appears to be preferred by most users, and they find support for Bates' hypothesis that users do not want fully automated search systems to which they can delegate the whole search process (cf.[11]).

In a similar effort to support users in formulating their queries, Google² provides a selection of interesting methods, one of which is "Did you mean".

² <http://www.google.com>

This facility suggests reformulations of queries that Google regards as a possible misspelling. The experimental *Google Suggest*³ interface provides users with an online prefix completion facility. According to Google's online documentation, this facility suggests term completion based on overall popularity of search strings, not on the user's preferences or search history. The main difference to "Did you mean" is that Google Suggest supports the user during the initial query formulation. Altavista⁴ also provides this kind of help with the variation that Altavista conducts searches both for the misspelled keyword and its automatic correction. The Scirus search engine for scientific information⁵ provides term suggestions, based on a term co-occurrence analysis in form of clickable links that modify the original query.

We argue for incorporation of as many help features as possible into the query formulation part of the interface, as this is the most difficult cognitive work for the user. It is also most important for successful reformulation and the effectiveness of the search process. Existing systems still leave much room for improvements. While approaches exist, they still lack integrated efforts with more than one method. Thus, a modular system of support components seems to be a reasonable approach.

3 Proactive Support

As described above, users require terminological and strategic help while searching. As they are often unaware of what they can do and what is possible within a system, they fail to actively use tools like thesauri, subject classifications, or dictionaries. The implication is that the system should observe the user's search tactics and offer help and suggestions at the right time and the right spot.

Thus, trying to support users *proactively*, means presenting suggestions or spelling corrections *without their explicit request*. In the given scenario of query formulation this means that some agent observes the query construction process and takes appropriate action if necessary or useful. This should happen in a way that matches the users' expectations and does not leave them puzzled or distracted by the offered suggestions. These methods have been successfully applied in other domains like office software or integrated development environments for quite some time and have proven to be very popular. Therefore we argue that these methods will also provide benefit for search interfaces and should always be considered when designing them. For this purpose all tools in DAFFODIL for entering and submitting queries are extended by an observation service. The proactive service should achieve several goals to improve efficiency and effectiveness:

- Mark potential errors on the typographical and syntactical level
- Suggest useful search terms, marking the partially constructed query

³ <http://www.google.com/webhp?complete=1&hl=en>

⁴ <http://www.altavista.com>

⁵ <http://www.scirus.com>

- Give useful hints on semantic query improvement
- Suggest useful actions and moves to add valuable facts to the query, e.g. looking up co-authors for the already known authors

These suggestions and markings have to appear

- at the right time, e.g. while the user is hesitating or thinking about possible extensions of the query
- in an unobtrusive manner – the user should be able to ignore the suggestions and to concentrate on the task
- in context – suggestions too far from the current focus of attention often will not be considered
- with clear semantics – the user needs to comprehend the goal of the suggestion and the reason the system is presenting the suggestion

3.1 Design of the Interface

Based on the goals described above, the subtasks of the design are to decide

- how to mark errors and words that the system can provide help on, and
- how to present suggestions (e.g. terms and actions)

The proactive functions should show the user *markers* and *suggestions* at the point of her work, which can be any query form.

- The user’s focus of attention is at that place⁶ and she should not be distracted with out-of-focus annotations and suggestions.
- The suggestions should appear in form of popup lists near the current cursor or mouse position, because they can be placed near the focus of the user without too much interruption and are a familiar concept. Icons, markup, and colours will be used to visualise for which items the proactive functions can provide suggestions.
- To provide semantical explanation some additional text may be useful, but it should not exceed the absolutely necessary level of verbosity.

The cognitive burden of the user will increase significantly if she is forced to think about the suggestions, e.g. by a separate dialog which *gets in the way*. It is desirable, that the cognitive model of what is displayed and why, is easy to learn and maintain, and also easy to ignore, if the user is sure about her input.

In the design of the *markers* we tried to use well-known symbolisms. Red curly lines (see figure 2) are used to mark erroneous words in a text. We expanded this scheme to mark words where the system can provide help, even if the word itself is not wrong. We used orange as a code for “warning”, marking words that might be, but not necessarily are wrong. We choose blue as a code for “information”, to mark words the system has suggestions for. We were aware

⁶ most users do in fact look at their keyboard instead of the form on the screen, but that is another matter

that colour codes might be misleading, because some colours like blue don't have a widely accepted meaning in the context of computer messages.

Another possible kind of "error" in our setting is located on the field level. These are mostly semantic errors, that we decided to mark this kind of errors by red square brackets around the affected text field. An example for this would be a query for documents of a specific author in a range of years where the author did not publish. While there is no error within any field that could be marked, the combination of the two field contents would lead to an empty result set. A similar error that is marked in this way is the conjunction of two year terms (see below). We chose this marker to avoid the visual overload resulting from using red boxes around a text field and to help colour blind people notice the markers by their distinctive shape.

The design of the *popup list* had to provide a means to present different kinds of information to the user – preferably in the same list at the same time. An additional goal was to structure the information to reduce the mass of visual stimuli the user has to process at once when the list pops up. Some ideas for this have been suggested in [16]. To implement such a hierarchical popup list we first wanted to develop and test a design for a flat list to expand it later after improving the shortcomings of the flat design.

Figure 1 shows the popup list presenting a selection of co-occurring terms related to the term *usability*, which is marked by a blue curly line. The user selected the term *user centred design* by keyboard navigation, which resulted in the informational message on the right side of the popup list. Accepting the selection would add the selected term to the query, connected to *usability* by

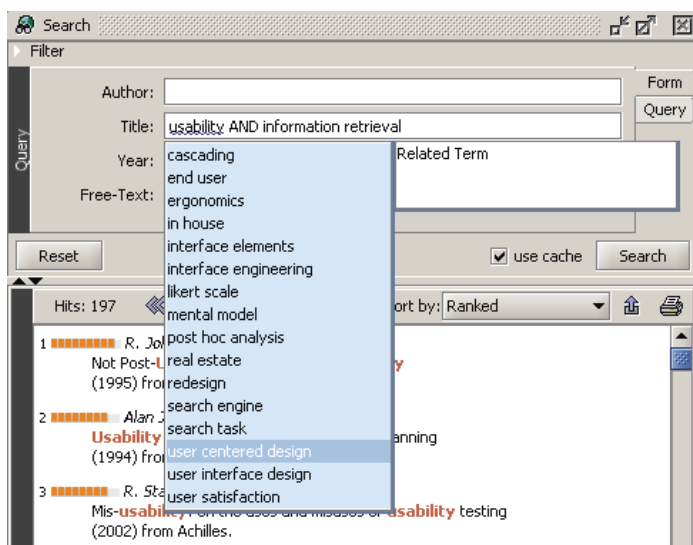


Fig. 1. The popup list presenting related terms and an info message

The screenshot shows a search window titled "Search" with a "Filter" section. The "Query" field contains "usability AND information retrieval". The "Year" field contains "2001 and 2002" and is highlighted with a red square bracket. Other fields include "Author:", "Free-Text:", and "Form". There are "Reset" and "Search" buttons, and a "use cache" checkbox.

Fig. 2. DAFFODIL marked the year field with red square brackets to notify the user about the over-constrained year clause

The screenshot shows the same search window. The "Title" field contains "information retrieval AND user interface". The "Year" field contains "interface" and "interfere". The "Year" field is highlighted with a blue background, and the word "interfere" is underlined in red. There are "Reset" and "Search" buttons, and a "use cache" checkbox.

Fig. 3. DAFFODIL suggests spelling corrections

an OR operator. The function would also put parentheses around the two terms, to be more specific on the query.

Figure 3 shows the first term marked in blue to notify the user about available suggestions and the second term in red because it is a possible error. The popup list suggests corrections. In this situation the user can only tell which term the suggestions are for based on the cursor position.

3.2 Design of the Functions

The agent observing a query form has several modules with different subgoals, providing an easily extensible set of functionality. The observer takes action if the user hesitates for a short moment, but the action can also be triggered actively with a key shortcut. When the cursor is on or directly behind a word for which suggestions are available, a popup list with the information (currently only terms) is shown. When pointing or navigating to a term, by mouse or keyboard, a short explanation of the reasons behind the suggestion is shown to the right. The list vanishes if the user continues typing, clicks somewhere else, or selects a term. Some aiding functions which are more expensive in terms of computation time (compared to user interaction) will take action when the user submits a query – to offer their suggestions at query reformulation time. The

form input is checked for typographical errors. If a misspelling is detected, terms are highlighted immediately with a red curly line. The query history module provides a convenient method to search the query history for old queries. Each list item presents the query and three example-hits that were found using the query for episodic memory support as in in [17]. The history of the user is also scanned to suggest terms which complete the currently entered prefix to previously entered terms. The related-terms module fetches terms, based on term co-occurrence, for the given query from the associated DAFFODIL service. Since it is quite expensive and fetching the results typically takes longer than half a second, the results are only being offered for query reformulation, after submission of the initial query. The thesaurus module delivers synonyms for given terms. It also offers broader or narrower terms if the number of results indicates that the query was too broad, respectively too narrow. Finally, the author-names module suggests correctly spelled author names, from a list of known authors.

4 Evaluation

The design of the proactive support methods was accompanied by three different qualitative user tests: heuristic evaluation of slide show mockups, controlled tests with two groups of students with and without the proactive functions, and single user loud thinking tests with video recording. Our expectations were that:

- the users accept the proactive functions
- the delay timing, until the proactive functions are triggered, is a good choice
- the users like the enhanced DAFFODIL more than the baseline version
- the users use the proactive functions often and even invoke them explicitly, once they found out how to do so
- the users have a clear understanding of the proactive functions

4.1 Methodology and Results

Heuristic evaluation with mockups. For the heuristic evaluation (HE) we displayed mockups of our proposed ideas in a mockup slide show. We wanted to see if users understood the displayed suggestions. We asked if they would accept this kind of suggestion while specifying some query. The users commented on the mockups, and on problems and ambiguities they found. They were asked to assign their problems to a heuristic problem category, as supposed in [18].

This heuristic evaluation showed that the design of the proactive functions was generally accepted; however, some aspects had to be re-worked, one of which was the form and colour of the error markers. Information which was displayed out of focus, e.g., hit count predictions in the status bar at the screen bottom, was requested to be made more visible. The markers inside of the form fields were generally accepted.

Controlled user test. Based on the revised design following from the heuristic evaluation, we implemented a first version of proactive support. This system version was tested with a group of 20 undergraduate students of computer science. They had to perform a set of five tasks, similar to those described in [2]. The tasks were a mixture of known item instantiation and topical searches. In addition to system logging of the users' interactions, we asked the users to fill a questionnaire after finishing the tasks.

The baseline evaluation, where half of the users performed the same tasks without any proactive support by the system, showed that approximately 45% of the submitted queries were erroneous. There were several classes of problems which occurred frequently:

1. Spelling mistakes in author names and query terms
2. Inadequate use of boolean operators or phrase syntax
3. Inadequate use of language (e.g. German instead of English)
4. Over-constrained queries – queries which cannot have answers, because of contradictory expressions in conjunctions
5. Under-constrained queries – queries using highly frequent search terms or only time ranges without any further expression

In the second part of this phase, the other half of the students had to perform the same tasks with proactive support enabled. In comparison to the baseline group, these test subjects worked in a more confident way, asked less questions, and had fewer negative comments. They were not more successful, however, and still lacked strategic skills to perform the topical task efficiently. These results are consistent with the findings of Brajnik et al., who observed that the searchers are often unaware of their need for strategic help, while they are actively requesting help at the terminological or syntactical level. Search times and success rate did not vary significantly between the two groups, but the number of participants was not sufficient for good quantitative results anyway.

Single user loud thinking tests. The findings of the first two test runs formed the basis of a third, refined implementation which was tested with ten single users in controlled loud thinking tests with video recording for in depth analysis of the process and the remaining problems. These users performed the same tasks as the twenty students in phase two. This group was mainly formed by nine research assistants. Furthermore a librarian, who normally teaches students to search in bibliographic databases.

Since the users were thinking loudly, a more in depth inspection about their cognitive models and expectations was possible then in the HE test at the beginning of our evaluation process. In addition we inspected how the timing of the suggestions worked. Timing is an important attribute of the proactive suggestions, as it turned out that the expected behaviour varies notably between users. Some users still requested lists which stay longer on the screen and are more eye catching, as they didn't notice them right away — mostly because they had to concentrate on the keyboard instead of the screen. One user had a general

disapproval of proactive functions, also in word processors or similar tools, and requested the ability to stop the observer (which is already possible, through the personalisation of DAFFODIL).

Overall Results

General acceptance. Overall the acceptance of the proactive functions varied. While most users started to generalise (requesting and expecting further proactive help in more situations), some refused to use the proactive functions. The conclusion is that each proactive function needs to be personalisable, so the user can turn the feature on or off.

Users used them multiple times during each task, depending on their level of confidence. In general, with growing confidence test subjects used the proactive functions less. Many users noted that they liked not having to type each word themselves. Users also stated that knowing that there are no spelling errors in their query makes them feel more confident, and that it saves them time because they don't have to reformulate a misspelled query.

Visual markers. The similarity between the marks for misspelled words and words for which term suggestions are available (red underline vs. blue underline) was not helpful for most users. While the design goal was to let users implicitly use the same mechanisms to open the popup list and choose the right substitution, it did not work smoothly. As the semantics are obviously clearly different, the graphical representation has to be as well.

Timing and Usage. The timing seemed to be a good compromise between a high level of disturbance and a long waiting time. No user complained about popup lists showing up too frequently, and no user stated that the time to provoke the list is too long. In fact, one user triggered the popup list during a short typing break but later answered that he did not see it. Some users wanted the lists to be on the screen longer. Currently, it closes immediately when the user continues typing and this was often done before realising that the suggestions might have been helpful in the current context.

Most users who actively provoked the suggestion lists, did so by simply deleting and re-typing a part of a word and waiting until the list opened. An easy way to open the suggestions with the mouse is needed, e.g. mouse-over or click events. None of the users tried to employ the context menu (right mouse button click) for this purpose, although this feature offers additional functions in several other DAFFODIL tools.

Understanding. Users generally require an explanation of the reasons behind a term suggestion, especially if the terms suggested are not known to the searchers (e.g., the system suggested the term "simulator sickness" after users searched for "cybersickness"). Some users were reluctant to use suggestions which they could not understand. To improve this, more semantics and reasoning should be offered in the comments that are given for each suggestion; however, since most

of the related terms suggested by the system are based on statistical correlation, this may not be very helpful for the user.

Overall. An assessment of the overall satisfaction after using the DAFFODIL system was requested as part of the questionnaire after the session. On average, users gave 7-8 out of ten points for retrieval quality and 6-7 points for usability. Considering this result and the comments of the users given during our loud thinking tests we are confident that proactive functions significantly improve the satisfaction of users searching for publications in DAFFODIL.

5 Summary and Outlook

DAFFODIL provides strategic support for searching in digital libraries. In this paper, we have focused on the usability of the most crucial functionality, namely query formulation in the search tool. In an iterative design and evaluation process, we have integrated proactive functions for supporting query formulation at the focus of the user's attention. The experimental results show that user satisfaction is increased and uncertainty is reduced.

The evaluation also showed a general acceptance and the request for even more proactive support, like faster query refinement and quick indication of empty result sets; we have started working on the development of appropriate methods for implementing these features.

A deeper personalisation, e.g. in form of individual user term-spaces instead of global ones, will aid the user even more. These term-spaces can be created from the user-specific data-flow in the DAFFODIL system; this will also avoid too large lists of synonyms or author name completions.

The presented evaluation of the proactive functions used qualitative methods, because of limited resources and the lack of a standardised document base. The next step is a quantitative evaluation, in order to show that proactive functions not only improve the user-friendliness of a digital library systems by reducing the typing effort during query formulation, but also increase the average retrieval performance in general and especially for novice users.

References

1. Kriewel, S., Klas, C.P., Schaefer, A., Fuhr, N.: Daffodil - strategic support for user-oriented access to heterogeneous digital libraries. *D-Lib Magazine* **10** (2004)
2. Klas, C.P., Fuhr, N., Schaefer, A.: Evaluating strategic support for information access in the DAFFODIL system. In Heery, R., Lyon, L., eds.: *Research and Advanced Technology for Digital Libraries. Proc. European Conference on Digital Libraries (ECDL 2004)*. Lecture Notes in Computer Science, Heidelberg et al., Springer (2004)
3. Maes, P., Wexelblat, A.: Interface agents. In: *CHI 96*. (1996) 2
4. Maes, P.: Agents that reduce work and information overload. *Communications of the ACM* **37** (1994) 30–40

5. Horvitz, E.: Principles of mixed-initiative user interfaces. In: CHI'99, Pittsburgh, PA, USA, ACM (1999) 159–166
6. Fuhr, N., Klas, C.P., Schaefer, A., Mutschke, P.: Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In: Research and Advanced Technology for Digital Libraries. 6th European Conference, ECDL 2002, Heidelberg et al., Springer (2002) 597–612
7. Tombros, A., Larsen, B., Malik, S.: The interactive track at inex 2004. In Fuhr, N., Lalmas, M., Malik, S., Szlavik, Z., eds.: INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the Third INEX Workshop. Dagstuhl, Germany, December 6–8, 2004. (2005)
8. Belkin, N.J.: Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science* **5** (1980) 133–143
9. Borgman, C.L.: The user's mental model of an information retrieval system. In: Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, ACM (1985) 268–273
10. Bates, M.J.: The design of browsing and berrypicking techniques for the online search interface. *Online Review* **13** (1989) 407–424
11. Bates, M.J.: Where should the person stop and the information search interface start? *Information Processing and Management* **26** (1990) 575–591
12. Marchionini, G.: Information seeking in electronic environments. (1995)
13. Kuhltau, C.C.: Developing a model of the library search process: Cognitive and affective aspects. *RQ* **28** (1988) 232–242
14. Schatz, B.R., Cochrane, P.A., Chen, H., Johnson, E.H.: Interactive term suggestion for users of digital libraries: Using subject thesauri and co-occurrence lists for information retrieval. In: DL'96: Proceedings of the 1st ACM International Conference on Digital Libraries. (1996) 126–133
15. Brajnik, G., Mizzaro, S., Tasso, C.: Evaluating user interfaces to information retrieval systems: A case study on user support. In Frei, H.P., Harman, D., Schäuble, P., Wilkinson, R., eds.: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, ACM (1996) 128–136
16. Sanderson, M., Coverson, C., Hancock-Beaulieu, M., Joho, H.: Hierarchical presentation of expansion terms. (2002)
17. Ringel, M., Cutrell, E., Dumais, S., Horvitz, E.: Milestones in time: The value of landmarks in retrieving information from personal stores. (2003) 184–191
18. Nielsen, J.: Heuristic evaluation. In: Usability Inspection Methods. John Wiley and Sons, New York (1994)

Expression of Z39.50 Supported Search Capabilities by Applying Formal Descriptions

Michalis Sfakakis and Sarantos Kapidakis

Archive and Library Sciences Department / Ionian University ,
Plateia Eleftherias, Paleo Anaktoro GR-49100 Corfu, Greece
{sfakakis, sarantos}@ionio.gr

Abstract. The wide adoption of the Z39.50 protocol from the Libraries exposes their abilities to participate in a distributed environment. In spite of the protocol specification of a unified global access mechanism, query failures and/or inconsistent answers are the pending issues when searching many sources due to the variant or poor implementations. The elimination of these issues heavily depends on the ability of the client to make decisions prior to initiating search requests, utilizing the knowledge of the supported search capabilities of each source. To effectively reformulate such requests, we propose a Datalog based description for capturing the knowledge about the supported search capabilities of a Z39.50 source. We assume that the accessible sources can answer some but possibly not all queries over their data, and we describe a model for their supported search capabilities using a set of parameterized queries, according to the Relational Query Description Language (RQDL) specification.

1 Introduction

The Z39.50 client/server information retrieval protocol [1] is widely used in Libraries electronic communication for searching and retrieving information from a number of diverse, distributed, heterogeneous and autonomous sources. According to Z39.50 architecture, every client can communicate with multiple servers (in parallel or sequentially), and every server can publish many sources not necessarily with the same structure and search capabilities.

The protocol unifies the access to the sources by providing an abstract record-based view model, hiding the logical structure and the access methods of the underlying sources. The supported query mechanism, utilizes sets of predefined Access Points combined with specific attributes (i.e. Attribute Sets), in a number of different query language specifications (i.e. query types). The general conformance requirements of the protocol, for the accomplishment of the standard search primitives, specify that at least the Access Points defined in the attribute set Bib-1 and the query Type-1 for the query formulation has to be recognized (although not necessarily implemented).

The consequences of these general conformance requirements are the arbitrary support of different subsets of the attribute set Bib-1 and also the different capabilities of the Type-1 query language, in the working Z39.50 environments. When a Z39.50 server does not support a requested Access Point or its attribute type values, the

response is either a message for unsupported search (query failure), or an arbitrary substitution of the unsupported attributes with others supported, giving unpredictable results. The client, can either restrict the available search characteristics to the set of the lowest common dominants, or reject all the attribute types' values for the query term and let each server apply any interpretation for them. Both approaches avoid query failures, but they either limit the querying facilities of the sources or produce inconsistent results.

When searching many sources, it is apparent that the elimination of the query failures and the improvement of the consistency for the answers depend on the client's ability: (i) to discover the supported search capabilities of every Z39.50 source and; (ii) based on this knowledge, to make decisions and probably to transform the query, prior to initiating search requests. For discovering the information about a Z39.50 source, the conformance to an implementation profile (e.g. Bath [16]) or the Explain facility of the protocol can be used. The ability of the client to decide and also to determine efficiently the appropriate query transformations heavily depends on the representation model used to capture the supported search capabilities of every source.

In the area of databases, a number of methods have been proposed for the representation and manipulation of the supported search capabilities from sources, based on formal descriptions [18]. Some of these describe the source's supported search capabilities by infinite families of queries, using a set of parameterized queries.

This work describes the supported search capabilities of a Z39.50 server at a higher level than the already existing mechanisms in the family of the Explain services, using a logic based language. The accessible sources are treated as sources which can answer some but not all possible queries over their data. Their supported search capabilities are described using a set of parameterized queries according to the Relational Query Description Language (RQDL) specification [12].

The rest of this work is organized as follows: section 2 presents the related work concerning the integrated access to multiple sources. Section 3 highlights the Z39.50 protocol, its access model, and describes the issues when searching many sources. Section 4, after a short introduction to the RQDL basics, presents the description of the supported search capabilities of a Z39.50 server. Finally, section 5 concludes and presents a number of interesting issues arrived from this work for further research.

2 Related Work

The problem of providing integrated access to multiple, distributed, heterogeneous and autonomous sources (databases, or other) has received considerable attention over a decade in the database research community, and is referred as constructing answers to queries using logical views. A common information integration architecture, shown in fig. 1, is based on the Mediators and Wrappers approach [20]. In this architecture, every source is wrapped by software (wrapper) which translates between the underlying source query language and data model to a common global language and data model. The Mediator receives queries from a client or a user, which are expressed in the global language and data model, and translates them into new queries, according to the wrapper capabilities description, which are sent to the wrappers. The translated queries are also expressed in the common language and model. Thus a mediator can

be thought as a global view of the integrated system, the wrapper as a local view of the underlying source and the problem of the information integration as constructing answers to queries using views that represent the capabilities of the information sources.

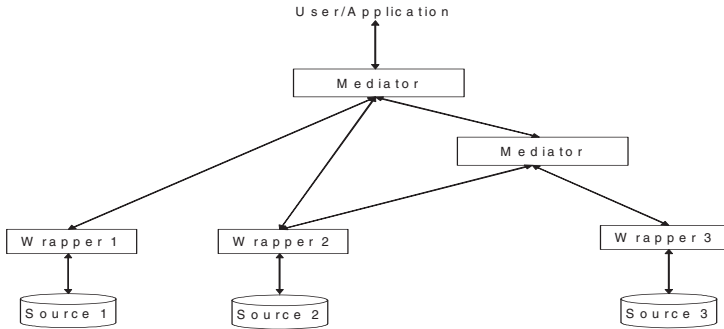


Fig. 1. Common Information Integration Architecture

Depending on the way that the global and the local views are constructed, there are two main approaches. The first one is the Local as View (LaV), where the global schema is defined independently of the local sources schemas. Each source is described in terms of the global schema, thus the sources are viewed as materialized views of the global schema. Using this approach is easy to add new sources in the system, but query transformation has exponential time complexity. The second approach is the Global as View (GaV), where the global schema is defined in terms of the local schemas. In this approach query transformation can be reduced to rule unfolding, but when a new source is added to the system, in most cases, the global schema has to be reconstructed. The Information Manifold [9] and the TSIMMIS [4] are two representative systems based on the Mediator/Wrapper architecture and implementing the two deferent view models respectively.

In the literature, a number of formal methods have been proposed [8] dealing with the problem of answering a query, posed over a global schema on behalf of representative source schemas. Most of these methods are based on the assumption that there is unrestricted access to the participated sources and their data schema, which in many cases is not a realistic one. Later extensions of the query/view model describe the access to sources by infinite families of queries [11, 10]. These approaches view the sources as processors that can answer some but not all possible queries over their data and describe those using a set of parameterized queries.

3 Z39.50 Protocol and the Multiple Search Problem Description

The Z39.50 is a state-full protocol based on the client/server model [1, 6]. It defines a standard manner for the communication between the client and the server, giving them the ability to interoperate independently of the underlying source structure, search procedures and computer systems. The system level interoperability is

approached by the definition of a set of specific services, which is accomplished by the exchange of specific messages, between the client and the server. For the content semantics of the published sources (databases), the protocol defines a standard model in a record-based abstract view, hiding the logical structure of the underlying source.

3.1 Access Model and the Explain Facility

For the implementation of the search primitives, the protocol utilizes the concept of the abstract Access Point, which the client can only use to query the sources. A server can supply access to many sources and for every source a different subset of the global set of Access Points could be supported.

In order to formulate a query, the protocol specifies many different query types (called Type-0, Type-1, etc.) mainly affecting the syntax of the query. For every search term, we have to define its characteristics by declaring the Attribute Set it belongs to. The Attribute Set defines the valid Access Points (i.e. what entities represent the search terms) from a specific set of attribute types, the way the system will match them against the underlying data, and the form in which the terms have been supplied. For the most commonly used Attribute Set Bib-1, the following attribute types exist: *Use* (e.g. Title, Author, etc.), *Relation* (e.g. Equal, less than, etc.), *Position* (e.g. First in field, any position in field, etc.), *Structure* (e.g. phrase, word, word list, etc.), *Truncation* (e.g. right, do not truncate, etc.) and *Completeness* (e.g. complete field, etc.).

According to the protocol, if a target does not support a given attribute list, the target should fail the search (i.e. query failure) and supply an appropriate diagnostic message, or the target will substitute it according to the 'Semantic Action' value. In most cases, the vast majority of the running Z39.50 servers ignores the 'Semantic Action' value and makes an arbitrary substitution of the unsupported attributes, without informing the client.

The Explain facility is the build-in mechanism in the protocol for a client to obtain the implementation details of a server. According to the service specification, among the information which a client can acquire from a server is the list of the supported Access Points with their Attribute Type combinations for every available source (database). The complexity of the implementation of the Explain facility, results to a small number of existing implementations. The latest approach to solve the problem of discovering information about a Z39.50 database is the ZeeRex [3], based on the experiences of the previous approaches. All the Explain approaches publish the supported access characteristics of a source by enumerating them in a list, without providing any information on the way they should be used.

3.2 Multiple Search Problem Description and Correlation to SRW

When searching multiple sources, the different implementations of the protocol result to query failures and/or inconsistent answers, despite of the unified access mechanism of the protocol. The different implementations mostly differ either to the subsets of the supported attribute types, or to the supported query language characteristics. The following examples illustrate some real world circumstances when a client tries to search many sources.

Example 1 (supported access point with different combinations of values for the other attribute types). Consider two sources, both of them answering queries using the Access Point (Use attribute) *Title*. Also both of them could combine this Access Point with the values *Phrase* or *Word* for the attribute type *Structure* and the last one supports additionally the value *Word List*. Finally the supported values for the attribute type *Truncation* are *Right* or *Do Not Truncate* on any *Structure* value. In summary the allowed searches for the Access Point *Title* by these two sources are:

```
(S1): Structure-(phrase, word),
      Truncation-(right, do not truncate)
(S2): Structure-(phrase, word, word list),
      Truncation-(right, do not truncate).
```

Obviously, Q1 (i.e. Search for the bibliographic records having the Title 'Data Structures in Pascal') is one supported query by both sources:

```
Q1: (Title, 'Data Structures in Pascal')
     (Structure, phrase) (Truncation, do not truncate).
```

The query Q2 is not supported by the source S1 due to the unsupported value *Word List* for the attribute type *Structure*.

```
Q2: (Title, 'Data Structures') (Structure, word list)
     (Truncation, do not truncate).
```

If a client knows that this query is not supported by the source S1, it could rewrite Q2 with the equivalent Q3, for the source S1, in a preprocessing step before sending it to the server, and will achieve the same recall and precision from the answer, as follows:

```
Q3: (Title, 'Data') (Structure, word)
     (Truncation, do not truncate) AND (Title, 'Structures')
     (Structure, word) (Truncation, do not truncate).
```

In order to simplify the description of the example, we made the assumption that both sources support the same value combinations for the remaining attribute types (i.e. *Relation*, *Position*, *Completeness*), for the used Access Point *Title*. In this example, the assumptions for the attribute types *Relation*, *Position*, *Completeness* were the values *Equal*, *First in Field*, *Complete Field*, respectively.

Example 2 (unsupported access point). Both sources support the Access Point *Author* with the following attribute types:

```
Access Point: Author
(S1): Structure-(phrase, word), Truncation-(right)
(S2): Structure-(phrase, word), Truncation-(right).
```

Also the S2 source additionally supports the Access Point *Author Personal Name*

```
Access Point: Author Personal Name
(S2): Structure-(phrase, word, word list),
     Truncation-(right, do not truncate).
```

Q4 is an unsupported query from the source S1, due to the unsupported Access Point *Author Personal Name*:


```
Q4: (Author Personal Name, 'Ullman')  
    (Structure, word) (Truncation, right).
```

A smart client must take into account the semantics of the Access Points (e.g. *Author Personal Name* is a subclass of *Author*) and transform the query for the S1 source, with less precision than the original one, as follows:

```
Q5: (Author, 'Ullman') (Structure, word)  
    (Truncation, right).
```

Closing the description of the issues concerning the Z39.50 environment, one interesting point is to address how these issues might impact the deployment of the Search and Retrieve Web Service (SRW) protocol [14]. The SRW is building on the Z39.50 semantics and retains the main concept of the abstract Access Points used in the access model of the Z39.50 protocol [15]. Also, in spite of the differences in the used terminologies (e.g. Z39.50 uses Attribute Sets and Attribute Types, SRW uses Context Sets and Indexes), the CQL query language used in SRW attempts to combine the simplicity and intuitiveness of expression with the richness of the Z39.50's Type -1 query language [5].

Also, the Explain facility of SRW is mandatory and uses the ZeeRex description for publishing the supported search capabilities of a source. As we saw in section 3.1, ZeeRex simply lists the supported Access Points without providing a representation model for effective management and use of the supported search capabilities.

It is apparent that the same issues still exist when searching multiple sources in the SRW environment, and consistent searching requires the description of the supported search capabilities of the underlying sources in a higher level than the one offered from the ZeeRex. Also, a higher-level description can be used as a bridge between the multiple sources when searching them either via the Z39.50 or the SRW protocol.

4 Description of a Z39.50 Server Supported Search Capabilities

In our approach we treat a Z39.50 server as a wrapper for the sources, able to answer some but not all possible queries over the data of every individual source. We recall that, all possible elements which can participate in a query are those defined as the Access Points in an Attribute Set and for every Access Point additional attributes could define its supplied form and the matching criteria against the data. Also, the queries are formulated according to a specific language (query type). Finally, an answer to a query is the set of all unique identifiers of the metadata records fulfilling the search criteria.

4.1 RQDL Basics

As the language for the description of the supported capabilities of the source, we use the Relational Query Description Language (RQDL). RQDL is a Datalog-based rule language, first proposed by Papakonstantinou et al. [12], for the description of a wrapper's supported queries. Its main advantages are the ability to describe infinite query sets and the support of schema-independent descriptions. The language focuses on

conjunctive queries and is powerful enough to express the abilities of many sources. Also, its extended version [19] can describe the set of all conjunctive queries. Due to the Datalog-based nature of the RQDL, we express the queries using Datalog [17].

We informally introduce the basic characteristics of the description language. The complete language specification is in [12], and the formal specification for its extended version is in [19]. An RQDL description is a finite set of RQDL rules, each of which is a parameterized query (i.e. query template). A query template has a ‘constant placeholder’ instead of the constant value of an actual query, thus represents many queries of the same form. For the restrictions on the actual values, which will replace the constant placeholders, the description language provides metapredicates on them.

4.2 Access Point Templates

We consider that a source wrapped by a Z39.50 server exports a predicate *metarec(Id)* representing the set of the unique identifiers of its metadata records. Also the source exports the predicate of the general form:

```
property(Id, Pname, Pattribute1, ..., Pattributen, Pval).
```

The relation expressing the meaning of the predicate *property* contains all the unique Ids from the metadata records having a property *Pname* with value that matches the *Pval* argument, according to the criteria specified from the additional *Pattribute_j*, $j=1, \dots, n$ attributes. Thus a valid element making the predicate *property* successful is:

```
property(X, use_Title, rel_Equal, pos_FirstInField,
        str_Phrase, tru_DoNotTruncate,
        com_CompleteField, 'Data Structures')
```

stating that the metadata record *X* has a property *use_Title* (i.e. Title) with value that matches the last argument ‘*Data Structures*’ according to the matching criteria defined from the third to seventh attributes (i.e. exact match).

From the predicate *property* we use the argument *Pname* to describe the supported Access Point from the source. Also, for the description of the other attribute types *Relation*, *Position*, *Structure*, *Truncation* and *Completeness*, defined in the Bib-1 attribute set, we use the other five arguments *Pattribute_j*. The values for the arguments *Pname* and *Pattribute_j*, in the predicate *property*, are the constants defined for the attribute types in the Bib-1 attribute set of the Z39.50 protocol. For readability purposes, we combine symbolic names and we do not use the actual numeric values as specified in the protocol. So, the symbolic name *use_Title* stands for the pair values (1, 4) representing the *Use* (i.e. Access Point) attribute type with value 4 (*Title*).

According to the RQDL specification, in order to define a representation for the set of the same form queries, in our case the queries concerning an Access Points and its characteristics, we have to define a query template using ‘constant placeholders’. These constant placeholders will be replaced in the actual queries with constant values. Thus, for the description of the family of the queries which use an Access Point with its attributes, we use the Access Point template:

```
property(Id, Pname, Pattribute1, ..., Pattributen, $Pval).
```

The identifiers started with the '\$' are the constant place holders (e.g. $\$Pval$) following the syntax of the RQDL. As an example for an Access Point template which specifies that the source supports the Access Point Title combined with the values *Equal*, *First in Field*, *Phrase*, *Do not Truncate* and *Complete Field*, for the other attribute types *Relation*, *Position*, *Structure*, *Truncation* and *Completeness* respectively (i.e. an exact match for the Title), is the following:

```
property(Id, use_Title, rel_Equal, pos_FirstInField,
         str_Phrase, tru_DoNotTruncate,
         com_CompleteField, $Pval).
```

The matching process of an Access Point specification, used in a query, and an Access Point template is accomplished by replacing the constant placeholders (e.g. $\$Pval$) with the corresponding actual constants and then by applying standard unification procedures

The number of the Access Point templates we have to write, in order to describe all possible combinations of the attribute types for a single Access Point, is the product of $(|Pa_j|+1)$ for $j= 1, \dots, 5$, where Pa_j is the set of the constant values defined for the j^{th} attribute type (including the null value). Thus we have 23,936 possible templates for every Access Point from the Attribute Set Bib-1, according to the protocol specification [1]. As described in the semantics of the Attribute Set Bib-1 [2], there is a number of conflicting or meaningless combinations of Attribute Types which decrease the above number of possible templates. E.g., Position with value 'First in subfield', where 'subfield' has no meaning or, Position attribute 'any position in field' is compatible only with the 'incomplete subfield' Completeness attribute, etc. In practice, we expect the number of the required templates to be small, according to wide adopted implementation profiles like the Bath [16]. As an indication for the order of magnitude of the number, we refer that there are totally only five attribute type combinations for each Access Point Author, Title and Subject in Bath profile (Functional Area A, Level 1). Thus we need five Access Point Templates only for each of the three Bath's Access Points.

According to the protocol, the specification of a query term may omit values for some attribute types. This leads to unspecified arguments when we construct the predicate property for the used term in the query. In this case the underscore '_' symbol can be used, and in the unification process it matches with any value in the corresponding argument of the template. When using unspecified arguments, there is a possibility that the corresponding predicate to the requested query term, will match with more than one Access Point Templates. For the decision of the matching Access Point template and in case were only one source is in use, we can make an arbitrary selection reflecting the intentions of the user and without conflicting with the protocol. A more interesting approach, which we have to examine further, is to select the template by taking into account user preferences limiting the degree of the expansion or the restriction of the query results. When many sources are involved, the primary criterion could be the selection of a template supported from all sources, or a common one after taking into account user preferences for achieving the same semantic changes for all sources. Both approaches satisfy the consistency of the answers but they differ on the achieved recall and precision.

Using the Access Point Template we can enumerate the supported Access Points and their attributes by a source. Even if we know that a source supports all the Access Points used in the query we may not be able to decide for the support of the whole query, due to the possibility of unsupported combinations in the query expressions.

4.3 Query Templates

This section extends the description of the Z39.50 supported search capabilities so that, we will be able to decide if a template describes a specific class of queries. Combining the predicates *metarec* and *property*, we can write a query requesting all the metadata records from a source which supports exact *Author* search, as:

```
(Q1): answer(X):- metarec(X), property(X, use_Author,
    rel_Equal, pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, 'Ullman').
```

A query template (D1), using the RQDL specification, which describes the capabilities of a source that supports only exact *Author* searches, is the following:

```
(D1): answer(Id):- metarec(Id), property(Id,
    use_Author, rel_Equal, pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pval).
```

A query is described by a template if each predicate in the query matches one predicate in the template and vice versa, and also, any metapredicate in the template evaluates to true when the constant value will replace the constant placeholder. The order of the predicates does not affect the matching process.

Query (Q1) matches the template of the description (D1), because the predicates used in the query match the predicates used in the template description and vice versa, with the following unification assignments: $X=Id$, $\$Pval = 'Ullman'$. Thus description (D1) describes the query (Q1). In case were other Access Points are supported from the source, in order the description (D1) to describe the whole capabilities (i.e. the set of all supported queries) we have to supplement D1 with a similar template for every other supported Access Point.

For the description of large or infinite sets of supported capabilities, we can use recursive rules. RQDL utilizes the concept of the *nonterminals* (as in context-free grammars), representing them by identifiers starting with an underscore ($_$) and a capital letter. A template containing *nonterminals* forms a *nonterminal template*. An expansion of a query template *qt* containing nonterminals is obtained by replacing each nonterminal of *qt* with one of the nonterminal templates that define it until there is no nonterminal in *qt*. Finally, a query template *qt* containing nonterminals describes a query *q* if there is an expansion of *qt* that describes *q*.

As an example, let's consider a source that supports the Access Points referred in the previous example, and also supports exact matches for the *Subject* and the *Title* Access Points plus any possible combination of them. A representative supported query by the server could be:

```
(Q2): answer(X):- metarec(X), property(X, use_Author,
    rel_Equal, pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, 'Ullman'),
    property(X, use_Author, rel_Equal, pos_FirstInField,
    str_Phrase, tru_DoNotTruncate,
    com_CompleteField, 'Garcia-Molina'),
    property(X, use_Subject, rel_Equal, pos_FirstInField,
    str_Phrase, tru_DoNotTruncate,
    com_CompleteField, 'Datalog'),
    property(X, use_Title, rel_Equal, pos_FirstInField,
    str_Phrase, tru_DoNotTruncate,
    com_CompleteField, 'Database Systems').
```

Using the nonterminal templates `_Cond` and `_Cond1`, a description for the supported queries from the server could be:

```
(D2): answer(Id):- metarec(Id), _Cond(Id)
(NT2.1) _Cond(Id):- _Cond(Id), _Cond1(Id)
(NT2.2) _Cond(Id):- _Cond1(Id)
(NT2.3) _Cond1(Id):- property(Id, use_Title,
    rel_Equal, pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pvalue)
(NT2.4) _Cond1(Id):- property(Id, use_Subject,
    rel_Equal, pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pvalue)
(NT2.5) _Cond1(Id):- property(Id, use_Author,
    rel_Equal, pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pvalue).
```

Also, the (E1) is an expansion of the query template (D2):

```
(E1): answer(Id):- metarec(Id),
    property(Id, use_Title, rel_Equal,
    pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pv1),
    property(Id, use_Subject, rel_Equal,
    pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pv2),
    property(Id, use_Author, rel_Equal,
    pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pv3),
    property(Id, use_Author, rel_Equal,
    pos_FirstInField, str_Phrase,
    tru_DoNotTruncate, com_CompleteField, $Pv4).
```

We recall that the order of the predicates does not affect the matching process between the query and the query template. Also, before starting the expansion, all of the variables of the template are renamed to be unique. This expansion describes the query (Q2), because the predicates used in the query match the predicates used in the template (and vice versa) with the unification assignments $X=Id$, $\$Pv1 = \text{'Database Systems'}$, $\$Pv2 = \text{'Datalog'}$, $\$Pv3 = \text{'Garcia-Molina'}$, $\$Pv4 = \text{'Ullman'}$.

4.4 Deciding for the Support of a Query

Having an RQDL description of the supported search capabilities of a Z39.50 source, the next step is to decide if the source is able to answer a given query. We recall that we express the queries (conjunctive) using the Datalog and also that an RQDL rule is a Datalog-based rule using constant placeholders in addition to variables and constants. The process of finding a supporting query in an RQDL description is reduced to the problem of determining whether a conjunctive query is contained in a Datalog program [11, 12, 19].

The Query Expressibility Decision (QED) [19] and the X-QinP [11] are two extensions of the classic algorithm for deciding query containment in a Datalog program [13, 17]. When the supported capabilities are described using recursive rules, the query template has an infinite number of expansions. Furthermore, we have to check the query for one or more matches within the infinite number of expansions in order to decide if a source is able to answer a query. In this case, a variant of ‘magic set rewriting’ [17] makes the process of deciding the support of a query more efficient [12].

Closing our approach for the description of a Z39.50 server supported search capabilities, we emphasize the importance of the applicability of the well-studied theory and algorithms from the area of the deductive databases.

5 Conclusions and Future Research

In this work we have addressed the need for the formal description of the supported search capabilities of Z39.50 sources, especially when multiple sources have to be searched. The proposed logic based description enables the client to make decisions prior to initiating the search requests. Also, the existing Explain family services can be used complementary to our description by providing input information, when they are implemented. The accessible sources are treated as sources which can answer some but not all possible queries over their data. We describe the search capabilities supported by a source using a set of parameterized queries, according to the Relational Query Description Language (RQDL) specifications.

From this work, a number of interesting points arrives for future development and research. Currently, our approach can help the client or a mediator to decide if a query is directly supported or not by a Z39.50 source (i.e. the server which publishes the source is able to answer the query as is, without any substitution of any attribute). In case where the query is not directly supported by a source, a powerful extension will be the transformation of the query to a different query or a set of queries, so that (preferably) identical or (otherwise) similar semantics are obeyed. Finding ways to extend the description templates using characteristics of the underlying data models and schemata of the sources will improve the overall process of deciding if a source supports a query, directly or indirectly. Also, the relations among the query language operations and the correlations between the Access Points hierarchies could really enforce the transformation procedures, especially when the query can be transformed only to a similar query.

References

1. ANSI/NISO: Z39.50 Information Retrieval: application service definition and protocol specification: approved May 10, 1995.
2. Attribute Set BIB-1 (Z39.50-1995): Semantics. <ftp://ftp.loc.gov/pub/z3950/defs/bib1.txt>.
3. An Overview of ZeeRex. 28th August 2002. <http://explain.z3950.org/overview/index.html>.
4. Chawathe, S., Garcia-Molina, H., Hammer, J., Irelandand, K., Papakonstantinou Y., Ullman, J. and Widom, J. The TSIMMIS Project: Integration of Heterogeneous Information Sources. IPSJ, Tokyo, Japan, October 1994.
5. CQL – Common Query Language, version 1.1, February 2004. Available from: <http://www.loc.gov/z3950/agency/zing/cql.html>
6. Finnigan, S., Ward, N. Z39.50 Made Simple. Available from: <http://archive.dstc.edu.au/DDU/projects/Z3950/zsimple.html>.
7. Gill, Tony and Miller Pall. Re-inventing the Wheel? Standards, Interoperability and Digital Cultural Content. D-Lib Magazine vol. 8:num. 1 (January 2002).
8. Halevy, A. Answering Queries using views: A Survey. The VLDB jour. 10: 270-294 (2001).
9. Kirk, T., Levy, A., Sagiv, Y. and Srivastava, D. The Information Manifold. AAAI Spring Symposium on Information Gathering, 1995.
10. Levy, A., Rajaraman, A., Ullman, J. Answering Queries Using Limited External Query Processors. PODS 96, Montreal Quebec Canada.
11. Papakonstantinou Y., Gupta, A. Garcia-Molina, H. Ullman, J. A Query Translation Scheme for Rapid Implementation of Wrappers. Proceedings of the Conference on Deductive and Object Oriented Databases, DOOD-95.
12. Papakonstantinou Y., Gupta A., Hass L. Capabilities-Based Query Rewriting in Mediator Systems. 4th International Conference on Parallel and Distributed Information Systems (PDIS-96), December 18-20, 1996.
13. Ramakrishnam, R., Sagiv, Y., Ullman, J. and Vardi, M. Proof Tree Transformation Theorems and their Applications. Proc. 8th ACM Symposium on Principles of Database Systems, pp. 172-181, 1989.
14. Sanderson, R. A Gentle Introduction to SRW. Available from: <http://www.loc.gov/z3950/agency/zing/srw/introduction.html>
15. SRW – Search/Retrieve Web Service: SRW's Relationship to Z39.50. January 22, 2004. Available from: <http://www.loc.gov/z3950/agency/zing/srw/z3950.html>
16. The Bath Profile: An International Z39.50 Specification for Library Applications and Resource Discovery. Available from: <http://www.ukoln.ac.uk/interop-focus/bath/current/>
17. Ullman, J. Principles of Database and Knowledge-Based Systems, v. I, II. Computer Science Press, New York, 1988 & 1989.
18. Ullman, J. Information Integration Using Local Views. LNCS, Proceedings of the 6th International Conference on Database Theory, pages 19-40, 1997.
19. Vassalos, V., Papakonstantinou Y. Expressive Capabilities Description Languages and Query Rewriting Algorithms. Jour. of Logic Programming, vol. 43, number 1, 2000, 75-122.
20. Wiederhold, G. Mediators in the architecture of future information systems. IEEE Computer, 25: 25-49, 1992.

A Comparison of On-Line Computer Science Citation Databases

Vaclav Petricek¹, Ingemar J. Cox¹, Hui Han²,
Isaac G. Councill³, and C. Lee Giles³

¹ University College London,
WC1E 6BT, Gower Street, London, United Kingdom
v.petricek@cs.ucl.ac.uk, ingemar@ieee.org

² Yahoo! Inc., 701 First Avenue, Sunnyvale, CA, 94089
huihan@yahoo-inc.com

³ The School of Information Sciences and Technology,
The Pennsylvania State University, University Park, PA 16802, USA
igc2@psu.edu giles@ist.psu.edu

Abstract. This paper examines the difference and similarities between the two on-line computer science citation databases DBLP and CiteSeer. The database entries in DBLP are inserted manually while the CiteSeer entries are obtained autonomously via a crawl of the Web and automatic processing of user submissions. CiteSeer's autonomous citation database can be considered a form of self-selected on-line survey. It is important to understand the limitations of such databases, particularly when citation information is used to assess the performance of authors, institutions and funding bodies.

We show that the CiteSeer database contains considerably fewer single author papers. This bias can be modeled by an exponential process with intuitive explanation. The model permits us to predict that the DBLP database covers approximately 24% of the entire literature of Computer Science. CiteSeer is also biased against low-cited papers.

Despite their difference, both databases exhibit similar and significantly different citation distributions compared with previous analysis of the Physics community. In both databases, we also observe that the number of authors per paper has been increasing over time.

1 Introduction

Several public¹ databases of research papers became available due to the advent of the Web [1,22,5,3,2,4,8] These databases collect papers in different scientific disciplines, index them and annotate them with additional metadata. As such, they provide an important resource for (i) finding publications, (ii) identifying important, i.e. highly cited, papers, and (iii) locating papers that cite a particular paper. In addition, author and document citation rates are increasingly being

¹ By public, we mean that access to the database is free of charge. Commercial databases are also available, the most well-known being the science-citation index [6]

used to quantify the scientific impact of scientists, publications, journals and funding agencies.

Within the computer science community, there are two popular public citation databases. These are DBLP [5] and CiteSeer [22]. The two databases are constructed in very different ways. In DBLP, each entry is manually inserted by a group of volunteers and occasionally hired students. The entries are obtained from conference proceeding and journals. In contrast, each entry in CiteSeer is automatically entered from an analysis of documents found on the Web. There are advantages and disadvantages to both methods and we discuss these issues in more detail in the next Section.

In Section 2 we compare the two databases based on the distribution of number of authors. We reveal that there are very pronounced differences which appear to be primarily due to the absence of very many single author papers in the CiteSeer database. A probabilistic model for document acquisition is then developed that provides an intuitive explanation for this phenomenon in Section 2.2.

There have been a number of studies on the distribution of citations [23,28,19,12] and the number of collaborators [26] using other on-line databases. This literature is reviewed in Section 3. We replicate some of these studies and show that citation distributions from both DBLP and CiteSeer differ considerably from those reported in other research communities.

2 The DBLP and CiteSeer Databases

There are a number of public, on-line computer science databases [1,22,5,3,2,4]. The CS BiBTeX database [4] contains a collection of over 1.4 million references. However, only 19,000 entries currently contain cross-references to citing or cited publications. The Compuscience database [2] contains approximately 400,000 entries. The Computing Research Repository CoRR [3] contains papers from 36 areas of computer science and is now part of ArXiv [1] that covers Physics, Mathematics, Nonlinear Sciences, Computer Science and Quantitative Biology. Networked Computer Science Technical Reference Library is a repository of Computer Science Technical Reports located at Old Dominion University.

DBLP was created by Michael Ley in 1998 [5]. It currently contains over 550,000 computer science references from around 368,000 authors. CiteSeer was created by Steve Lawrence and C. Lee Giles in 1997 [22]. It currently contains over 716,797 documents.

We chose to examine DBLP and CiteSeer due to the availability of detailed citation information and their popularity.

In our analysis we focus on the difference in data acquisition and the biases that this difference introduces.

2.1 The Differences Between DBLP and CiteSeer Databases

While both the DBLP and CiteSeer databases contain computer science bibliography and citation data, their acquisition methods greatly vary. In this section

we first discuss these differences in acquisition methods, then we look at the distribution of papers over time in each dataset, and after that we compare the distribution in the number of authors per paper. Section 2.2 then describes acquisition models for both DBLP and CiteSeer.

Data acquisition. At the time of writing, DBLP contains over 550,000 bibliographic entries. Papers in DBLP originally covered database systems and logic programming. Currently DBLP also includes theory of information, automata, complexity, bioinformatics and other areas. Database entries are obtained by a limited circle of volunteers who manually enter tables of contents of journals and conference proceedings. The volunteers also manually entered citation data as part of compiling the ACM anthology CD/DVDs. Corrections that are submitted to the maintainer are also manually checked before committing. Though the breadth of coverage may be more narrow than CiteSeer, DBLP tries to ensure comprehensive and complete coverage within its scope. The coverage of ACM, IEEE and LNCS is around 80–90%. The narrower focus of DBLP is partially enforced by the cost associated with manual entry. Although there is the possibility of human error in the manual process of DBLP, its metadata is generally of higher quality than automatically extracted metadata².

In our analysis we used a DBLP dataset consisting of 496,125 entries. From this we extracted a dataset of 352,024 papers that specified the year of publication and the number of authors. Only papers published between 1990 and 2002 were included, due to the low number of papers available outside of this range.

CiteSeer currently contains over 716,797 bibliographic entries. Automatic crawlers have the potential of achieving higher coverage as the cost of automatic indexing is lower than for manual entry. However, differences in typographic conventions make it hard to automatically extract metadata such as author names, date of publication, etc.

CiteSeer entries can be acquired in two modes. First, the publication may be encountered during a crawl³. In this case, the document will be parsed, and title, author and other information will be entered into the database. Second, during this parsing operation, a document's bibliography is also analyzed and previously unknown cited documents are also entered into the database.

CiteSeer is continuously updated with user submissions. Currently updates are performed every two weeks. However, it was not updated at all during the period from about March 2003 to April 2004. Prior to March 2003 crawls were made with declining regularity. As of July 2004 CiteSeer has been continuously crawling the web to find new content using user submissions, conference, and journal URLs as entry points.

In our analysis, we used a CiteSeer dataset consisting of 575,068 entries. From this we extracted a dataset of 325,046 papers that specified the year of publication and the number of authors. Once again, only papers published between

² This remains true, despite the recent improvement of automatic extraction algorithms by use of support vector machines [14].

³ CiteSeer is not performing a brute force crawl of the web but crawling a set of starting pages to the depth of 4-7

1990 and 2002 were considered. It is also important to note that this dataset only contained entries that CiteSeer acquired by parsing the actual document on the Web, i.e. documents that were only cited but not actually parsed, were not included. We assume that the number of parsing errors is independent of the number of authors and does not introduce any new bias.

CiteSeer may be considered a form of self-selected on-line survey - authors may choose to upload the URL where their publications are available for subsequent crawling by CiteSeer. This self-selection introduces a bias in the CiteSeer database that we discuss later. A fully automatic scientometric system is also potentially susceptible to “shilling” attacks, i.e. authors trying to alter their citation ranking by, for example, submitting fake papers citing their work. This later issue is not discussed further here, but appears related to similar problems encountered by recommender systems [20].

Accumulation of papers per year. In order to compare the two databases, we first examined the number of publications in the two datasets for the years 1990 through 2002. These years were chosen to ensure that a sufficient number of papers per year is available in both datasets.

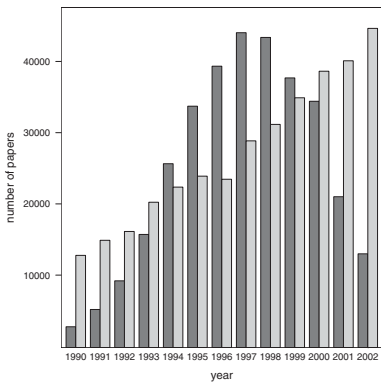


Fig. 1. Number of papers published in the years from 1990 to 2002 present in the DBLP (light) and CiteSeer (dark) databases

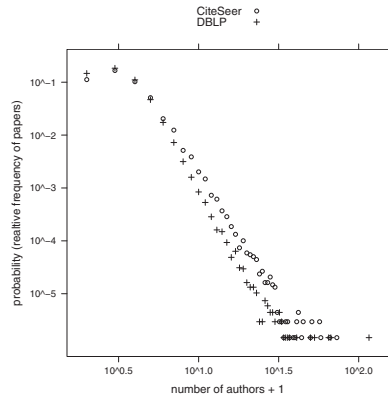


Fig. 2. Probability histogram of number of authors. (double logarithmic scale)

Figure 1 shows a considerable difference in the number of papers present in the two databases on an annual basis.

The increase in the papers per year exhibited by DBLP is probably explained by a combination of (i) the increasing number of publications each year [27,24] and (ii) an increase in the coverage of DBLP thanks to additional funding and improvement in processing efficiency⁴.

⁴ Personal communication with Michael Ley

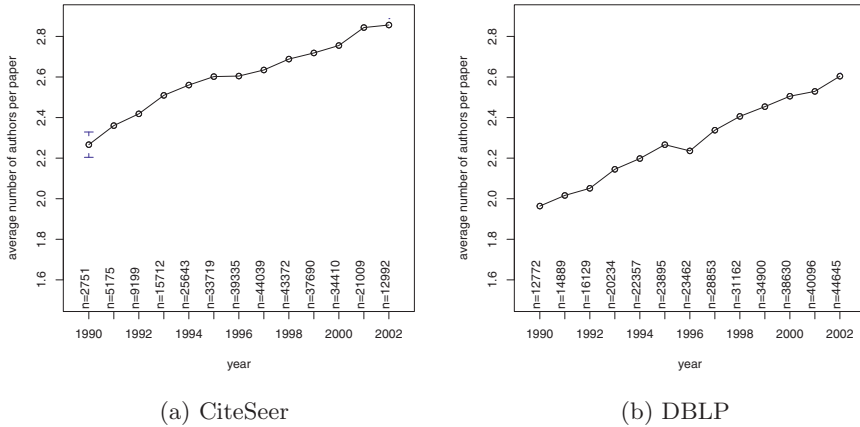


Fig. 3. Average number of authors per paper for the years 1990 to 2002

The decrease in the number of papers per year exhibited by CiteSeer since 1997 is mainly due to (i) declining maintenance, although (ii) declining coverage (iii) intellectual property concerns (iv) dark matter effect [9] (v) end of web fever and (vi) specifics of submission process, may also have contributed.

Team size. We also examined the average number of authors for papers published between 1990 and 2002, see Figure 3. In both datasets, the average is seen to be rising. It is uncertain what is causing this rise in multi-authorship. Possible explanations include (i) funding agencies preference to fund collaborative research and/or (ii) collaboration has become easier with the increasing use of email and the Web. We observe that the CiteSeer database contains a higher number of multi-author papers.

Bias in number of authors. Figure 2 examines the relative frequency of n -authored papers in the two datasets. Note that the data is on a log-log scale. It is clear that CiteSeer has far fewer single and two authored papers. In fact, CiteSeer has relatively fewer papers published by one to three authors. This is emphasized in Figure 4 in which we plot the ratio of the frequency of n -authored papers in DBLP and CiteSeer for one to fifty authors. Here we see the frequency of single-authored papers in CiteSeer is only 77% of that occurring in DBLP. As the number of authors increases, the ratio decreases since CiteSeer has a higher frequency of n -authored papers for $n > 3$. For $n > 30$, the ratio is somewhat random, reflecting the scarcity of data in this region. We therefore limit our analysis to numbers of authors where there are at least 100 papers in each dataset. This restricts the number of authors to less than 17.

As we see in Figure 2 the number of authors follows a power law corresponding to a line with slope approximately -0.23 for DBLP and -0.24 for CiteSeer. There is an obvious cut-off from the power law for papers with low number of authors. For CiteSeer, we hypothesize that (i) papers with more authors are more

likely to be submitted to CiteSeer and (ii) papers with more authors appear on more homepages and are therefore more likely to be found by the crawler. These ideas are modeled in Section 2.2.

However none of these factors is relevant to DBLP, which also exhibits a similar drop off in single-authored papers. Other explanations may be that (i) single author papers are less likely to be finished and published, (ii) funding agencies encourage collaborative and therefore multi-authored research and (iii) it is an effect of limited number of scientists in the world [19].

2.2 DBLP and CiteSeer Data Acquisition Models

To explain the apparent bias of CiteSeer towards papers with larger numbers of authors, we develop two possible models for the acquisition of papers within CiteSeer. We also provide a simple acquisition model for DBLP.

The first CiteSeer model is based on authors submitting their papers directly to the database. The second CiteSeer model assumes that the papers are obtained by a crawl of the Web. We show that in fact, both models are equivalent.

To begin, let $\text{citeseer}(i)$ be the number of papers in CiteSeer with i authors, $\text{dblp}(i)$ the number of papers in DBLP with i authors and $\text{all}(i)$ the number of papers with i authors published in all Computer Science.

For DBLP, we assume a simple paper acquisition model such that there is a probability α that a paper is included in DBLP and that this probability is independent of the number of authors.

For CiteSeer we assume that the acquisition method introduces a bias such that the probability, $p(i)$ that a paper is included in CiteSeer is a function of number of authors of that paper. That is,

$$\text{dblp}(i) = \alpha \cdot \text{all}(i) \tag{1}$$

$$\text{citeseer}(i) = p(i) \cdot \text{all}(i) = p(i) \cdot \frac{\text{dblp}(i)}{\alpha} \tag{2}$$

CiteSeer Submission Model. Let $\beta \in (0, 1)$ be the probability that an author submits a paper directly to CiteSeer then $p(i) = 1 - (1 - \beta)^i$ where $(1 - \beta)^i$ is the probability that none of the i authors submit their paper to CiteSeer.

Substituting to (2) and re-arranging, we have

$$r(i) = \frac{\text{dblp}(i)}{\text{citeseer}(i)} = \frac{\alpha}{1 - (1 - \beta)^i} \tag{3}$$

It is clear from Equation 3 that as the number of authors, i , increases, the ratio, $r(i)$, tends to α , i.e. we expect that the number of i -authored papers in CiteSeer will approach $\text{all}(i)$ and thus from Equation 1 the ratio tends to α . For single authored papers, i.e. $i = 1$, we have that $r(1) = \frac{\alpha}{\beta}$ and since we know that DBLP has more single-authored papers, it must be the case that $\beta < \alpha$. More generally, we expect the ratio, $r(i)$, to monotonically decrease with the number of authors,

i , reaching an asymptote of α for large i . This is approximately observed in Figure 4, ignoring points for $n > 30$ for which there is a scarcity of data.

In Figure 5 we plot the proportion $r(i)$ for numbers of authors i where we have at least 100 papers available. We fit Equation 3 to the data in Figure 5⁵. We see the fit is not perfect suggesting that this is not the only mechanism in play.

The value to which the data points are converging for high numbers of authors is $\alpha \approx 0.3$. We have to take into account that we only used 71% of DBLP papers and 57% of CiteSeer papers in our analysis – the papers that have both year and number of authors specified. Substituting α into (4) we get the value of $\alpha' \approx 0.24$. If our model is correct, this would suggest that the DBLP database covers approximately 24% of the entire Computer Science literature.

$$\alpha' = \frac{\text{complete_dblp}(i)}{\text{complete_citeseer}(i)} = \frac{0.57}{0.71} \cdot \frac{\text{dblp}(i)}{\text{citeseer}(i)} = 0.8 \cdot \alpha \tag{4}$$

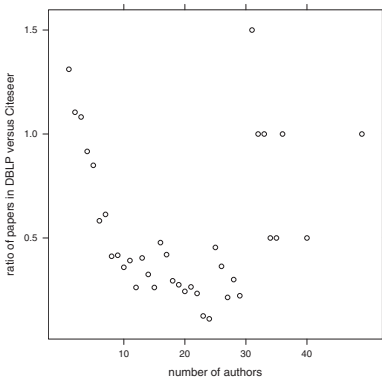


Fig. 4. Ratio of the number of authors in DBLP to CiteSeer as a function of the number of authors of a paper

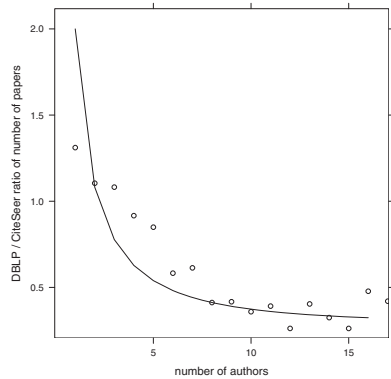


Fig. 5. Fit of model (3) for values of $\alpha = 0.3$ and $\beta = 0.15$ for numbers of authors where there are at least 100 documents in both datasets in total

CiteSeer Crawler Model. CiteSeer not only acquires papers based on direct submission by authors, but also by a crawl of the Web.

To begin, let $\delta \in (0, 1)$ be the probability that an author puts a paper on a web site (homepage for example). Then the average number of copies of an n -authored paper on the Web is $n \cdot \delta$. Let us further assume that the crawler finds each available on-line copy with a probability γ . If $pp(i, c)$ denotes the probability that there will be c copies of an i -authored paper published on-line, then we have:

⁵ Note that this is the same data as Figure 4 but restricted to $n < 17$.

authors	pp(i,c)	
1	$pp(1, 1) = \delta$	1 copy online
	$pp(1, 0) = 1 - \delta$	0 copies online
2	$pp(2, 2) = \delta^2$	2 copies online
	$pp(2, 1) = 2\delta(1 - \delta)$	1 copy online
	$pp(2, 0) = (1 - \delta)^2$	0 copies online
\vdots		
n	$pp(n, c) = \binom{n}{c} \delta^c (1 - \delta)^{n-c}$	c copies online of an n -authored paper

The probability, $pf(c)$, of crawling a document with c copies online, is

$$pf(c) = 1 - (1 - \gamma)^c \tag{5}$$

thus the probability that CiteSeer will crawl an n -authored document, $p(n)$ is

$$\begin{aligned}
 p(n) &= \sum_{c=0}^n pp(n, c) pf(c) \\
 &= \sum_{c=0}^n pp(n, c) (1 - (1 - \gamma)^c) \\
 &= \sum_{c=0}^n \left(\binom{n}{c} \delta^c (1 - \delta)^{n-c} \right) (1 - (1 - \gamma)^c) \\
 &= 1 - \sum_{c=0}^n \left(\binom{n}{c} \delta^c (1 - \delta)^{n-c} \right) (1 - \gamma)^c \quad (\text{sum of probabilities equals 1}) \\
 &= 1 - \sum_{c=0}^n \left(\binom{n}{c} ((1 - \gamma)\delta)^c (1 - \delta)^{n-c} \right) \\
 &= 1 - (\delta(1 - \gamma) + (1 - \delta))^n \quad (\text{from binomial theorem}) \\
 &= 1 - (\delta - \gamma\delta + 1 - \delta)^n \\
 &= 1 - (1 - \gamma\delta)^n \tag{6}
 \end{aligned}$$

where $(1 - \gamma\delta)^n$ is the probability that no copy of an n -author paper is found by CiteSeer.

Once again, if we substitute Equation (6) in 2, we have

$$r(i) = \frac{\text{dblp}(i)}{\text{citereer}(i)} = \frac{\alpha}{1 - (1 - \gamma\delta)^i} \tag{7}$$

which is equivalent to the “submission” model of Equation 3. That is, both models lead to the same bias.

3 Prior Work

There has been considerable work in the area of citation analysis and a comprehensive review is outside of the scope of this paper. Broadly, prior citation analysis has examined a wide variety of factors including (i) the distribution of citation rates [28,23,12,19], (ii) the variation in the distribution of citation rates across research fields and geographical regions [23,15], (iii) the geographic distribution of highly cited scientists [10,11] (iv) various indicators of the scientific performance of countries [25] (v) citation biases and miscitations [17,18,29] (vi) collaboration networks [26] (vii) distribution of references in papers [30], and (viii) visualization and navigation [16,13].

The number of citations is the most widely used measure of academic performance and as such it influences decisions about distribution of financial subsidies. The study of citation distributions helps us understand the mechanics behind citations and objectively compare scientific performance.

With regard to the distribution of citations, Laherrere *et al* [19] argued that a stretched exponential⁶ is suitable for modeling citation distributions as it is based on multiplicative processes and does not imply an unlimited number of authors. Redner [28] then analyzed the ISI and Physical Review databases and showed that the number of citations of highly cited papers follows a power-law. Lehmann [23] attempted to fit both a power law and stretched exponential to the citation distribution of 281,717 papers in the SPIRES [7] database and showed it is impossible to discriminate between the two models.

So far most of the research on citation distributions has come from the Physics community. Surprisingly little work has been done on computer science papers. The ISI dataset contains computer science papers but these were usually studied together with other disciplines despite the fact that their dynamics may differ. The only work the authors are aware of [26] is based on a small dataset (13000 papers) and was concerned with the distribution of the number of collaborators.

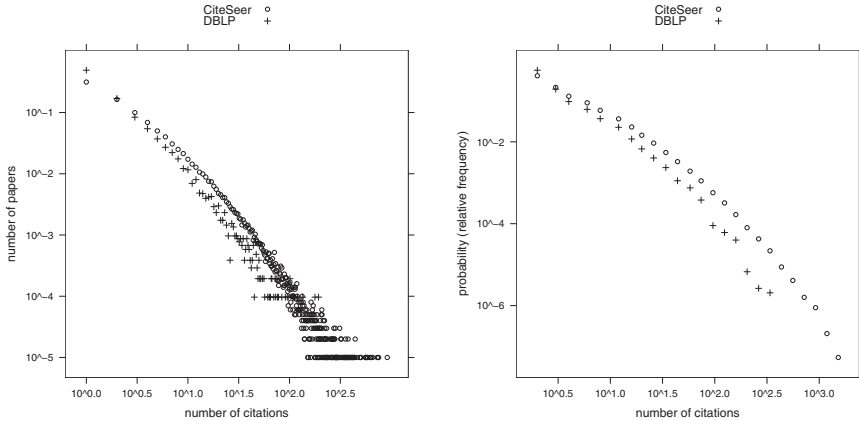
In the next section we examine the distribution of citations in both the CiteSeer and DBLP datasets.

3.1 Citation Distributions for Computer Science

Citation linking in DBLP was a one-time project performed as a part of the 'ACM SIGMOD Anthology' - a CD/DVD publication. The citations were entered manually by students paid by ACM SIGMOD. As a result DBLP now contains a significant number of new papers that have not been included in this effort. To mitigate against this distortion, we limit ourselves in both datasets to papers that have been cited at least once (CiteSeer 100,059 papers, DBLP: 10,340 papers).

Figure 6(a) compares citation distributions in CiteSeer versus DBLP. We see that DBLP contains more low cited papers than CiteSeer. We currently do not have an explanation for this phenomenon. However, it may be related to Lawrence's [21] observation that articles freely available online are more highly cited.

⁶ Stretched exponential distribution has the form $exp(-(x/w)^c)$



(a) atomic histogram (b) histograms after exponential binning

Fig. 6. Probability histograms on double logarithmic scales for number of citations in the two datasets

Table 1. Slopes for Figure 6(b) representing the parameter of the corresponding power-laws

number of citations	slope		
	Lehmann	CiteSeer	DBLP
< 50	-1.29	-1.504	-1.876
> 50	-2.32	-3.074	-3.509

We use exponential binning (Figure 6(b)) to estimate the parameters of the citation distribution in CiteSeer and DBLP. Exponential binning is a technique where the data are aggregated in exponentially increasing ‘bins’. In this manner we obtain a higher number of samples in each bin, which reduces the noise in the data.

The slopes in Table 1 correspond to linear interpolation of exponentially binned data as displayed in Figure 6(b). Higher slopes in our datasets indicate a more uneven distribution of citations. The papers in each dataset have been divided into two groups – papers with more than and less than 50 citations.

For both datasets we obtain parameters bigger in absolute value than Lehmann [23] derived for Physics. This means that highly cited papers acquire a larger share of citations in Computer Science than in Physics. However, there is also a significant difference between CiteSeer and DBLP.

4 Conclusions

This paper compared two popular online science citation databases, DBLP and CiteSeer, which have very different methods of data acquisition. We showed that

autonomous acquisition by web crawling, (CiteSeer), introduces a significant bias against papers with low number of authors (less than 4). Single author papers appear to be disadvantaged with regard to the CiteSeer acquisition method. As such, single authors, (who care) will need more actively submit their papers to CiteSeer if this bias is to be reduced.

We attempted to model this bias by constructing two probabilistic models for paper acquisition in CiteSeer. The first model assumes the probability that a paper will be submitted is proportional to the number of authors of the paper. The second model assumes that the probability of crawling a paper is proportional to the number of online copies of the paper and that the number of online copies is again proportional to the number of authors. Both models are equivalent and permit us to estimate that the coverage of DBLP is approximately 24% of the entire Computer Science literature.

We then examined the citation distributions for both CiteSeer and DBLP and observed that CiteSeer has a fewer number of low-cited papers. The citation distributions were compared with prior work by Lehmann [23], who examined datasets from the Physics community. While the CiteSeer and DBLP distributions are different, both datasets exhibit steeper slopes than SPIRES HEP dataset, indicating that highly cited papers in Computer Science receive a larger citation share than in Physics.

Acknowledgments

The authors thank Michael Ley for his assistance in understanding DBLP.

References

1. Arxiv e-print archive, <http://arxiv.org/>.
2. Compuscience database, <http://www.zblmath.fiz-karlsruhe.de/COMP/quick.htm>.
3. Corr, <http://xxx.lanl.gov/archive/cs/>.
4. Cs bibtex database, <http://liinwww.ira.uka.de/bibliography/>.
5. Dblp, <http://dblp.uni-trier.de/>.
6. Scientific citation index, <http://www.isinet.com/products/citation/sci/>.
7. Spires high energy physics literature database, <http://www.slac.stanford.edu/spires/hep/>.
8. Sciencedirect digital library, <http://www.sciencedirect.com>, 2003.
9. P. Bailey, N. Craswell, and D. Hawking. Dark matter on the web. In *Poster Proceedings of 9th International World Wide Web Conference*. ACM Press, 2000.
10. M. Batty. Citation geography: It's about location. *The Scientist*, 17(16), 2003.
11. M. Batty. The geography of scientific citation. *Environment and Planning A*, 35:761–770, 2003.
12. T. C and de Albuquerque MP. Are citations of scientific papers a case of nonextensivity?, 2000.
13. D. Cosley, S. Lawrence, and D. M. Pennock. REFEREE: An open framework for practical testing of recommender systems using researchindex. In *28th International Conference on Very Large Databases, VLDB 2002*, Hong Kong, August 20–23 2002.

14. H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines, 2003.
15. M.-J. Kim. Comparative study of citations from papers by korean scientists and their journal attributes, 1998.
16. S. Klink, M. Ley, E. Rabbidge, P. Reuther, B. Walter, and A. Weber. Browsing and visualizing digital bibliographic data, 2004.
17. J. S. Kotiaho. Papers vanish in mis-citation black hole, 1999.
18. J. S. Kotiaho. Unfamiliar citations breed mistakes, 1999.
19. J. Laherrre and D. Sornette. Stretched exponential distributions in nature and economy: 'fat tails' with characteristic scales. *The European Physical Journal B - Condensed Matter*, 2(4):525-539, 1998.
20. S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, pages 393-402. ACM Press, 2004.
21. S. Lawrence. Online or invisible? *Nature*, 411(6837):521, 2001.
22. S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67-71, 1999.
23. S. Lehmann, B. Lautrup, and A. D. Jackson. Citation networks in high energy physics. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 68(2):026113, 2003.
24. L. M. The dblp computer science bibliography: Evolution, research issues, perspectives, 2002.
25. R. M. May. The scientific wealth of nations. *Science* 275 793 795, 1997.
26. M. E. J. Newman. The structure of scientific collaboration networks, 2000.
27. D. D. S. Price. Price, d. de solla, little science, big science, columbia univ. press, new york, 1963., 1963.
28. S. Redner. How popular is your paper? an empirical study of the citation distribution. *European Physics Journal B* 4 131 134, 1998.
29. M. Simkin and V. Roychowdhury. Read before you cite!, 2002.
30. A. Vazquez. Statistics of citation networks, 2001.

A Multi-document Summarization System for Sociology Dissertation Abstracts: Design, Implementation and Evaluation

Shiyan Ou, Christopher S.G. Khoo, and Dion H. Goh

Division of Information Studies,
School of Communication & Information,
Nanyang Technological University,
Singapore, 637718
{pg00096125, assgkhoo, ashlgoh}@ntu.edu.sg

Abstract. The design, implementation and evaluation of a multi-document summarization system for sociology dissertation abstracts are described. The system focuses on extracting variables and their relationships from different documents, integrating the extracted information, and presenting the integrated information using a variable-based framework. Two important summarization steps – information extraction and information integration were evaluated by comparing system-generated output against human-generated output. Results indicate that the system-generated output achieves good precision and recall while extracting important concepts from each document, as well as good clusters of similar concepts from the set of documents.

1 Introduction

Multi-document summarization has begun to attract much attention in the last few years [6]. A multi-document summary has several advantages over the single-document summary. It provides an integrated overview of a document set indicating common information across many documents, unique information in each document, and cross-document relationships, and allows users to zoom in for more details on aspects of interest.

Our present work aims to develop a method for automatic summarization of sets of sociology dissertation abstracts that may be retrieved by a digital library system or search engine in response to a user query. Recently, many digital libraries have begun to provide online dissertation abstract services, since they contain a wealth of high-quality information by specifying research objectives, research methods and results of dissertation projects. However, a dissertation abstract is relatively long about 300–400 words and browsing too many of such abstracts results in information overload. Therefore, it would be helpful to summarize a set of dissertation abstracts to assist users in grasping the main ideas on a specific topic.

The main approaches used for multi-document summarization include sentence extraction, template-based information extraction, and identification of similarities and differences between documents. With sentence extraction, documents or sentences across all the documents are clustered, following which, a small number of sentences

are selected from each cluster [1,7,12]. Some multi-document summarizers, such as SUMMONS [5], RIPTIDES [14] and GITEXTER [2], use information extraction techniques to extract pieces of information to fill in a pre-defined template. Another important approach for multi-document summarization is to extract information that is common or repeated in several documents plus selected unique information in individual documents to generate the summaries [4]. In addition, cross-document rhetorical relationships are used to create multi-document summaries [11, 15]. However, these existing summarization approaches focus more on physical granularities (words, phrases, sentences and paragraphs) and rhetorical relations based on shallow analysis, without paying much attention to higher-level semantic content and semantic relations expressed within and across documents. Another problem is that different users have different information needs. Thus, an ideal multi-document summarization should provide different levels of detail for different aspects of the topic according to the user's interest. But these approaches usually construct fixed multi-document summaries.

In our work, we do not use the traditional summarization approaches. Instead, our work focuses on semantic-level research variables and their relationships. A variable is a specific concept explored in a particular research whose value changes from case to case. For example, gender can be considered a variable because it can take two values "male" and "female". A relationship refers to the correspondence between two variables [13]. In a set of related sociology dissertation abstracts, similar concepts across documents are usually investigated in various projects in different contexts or from different perspectives. This means that the similarities and differences across dissertation abstracts are mainly transferred through variables and their relationships. Therefore, a variable-based framework is developed to integrate variables and their relationships extracted from different abstracts and thus summarize a set of dissertation abstracts on a specific topic [8]. The framework has a hierarchical structure in which the summarized information is at the top level and the more detailed information is found at lower levels. It integrates four kinds of information:

- *Main variables*: The main variables are usually common concepts investigated by most dissertation abstracts in a document set.
- *Relationships between variables*: For each main variable, the descriptive values or the relationships with other variables are investigated in different dissertations.
- *Contextual relations*: Some studies explore variables and relationships through the perception of a group of people or in the context of a framework or model.
- *Research methods*: To explore the attributes of a variable or relationships between a pair of variables, one or more research methods are used.

The framework not only provides an overview of the subject area but also allows users to explore details according to their interest. Based on the framework, an automatic summarization method for sociology dissertation abstracts is developed. The method extracts variables and their relationships from different documents, integrates the extracted information across documents, and presents the integrated information using the variable-based framework. Although the summarization method was developed based on sociology dissertation abstracts, it is also applicable to other domains, such as psychology and medicine, which adopt the same research paradigm of seeking to investigate concepts and variables and their relationships and use a similar research report structure.

2 Overview of the Multi-document Summarization System

The summarization system follows a pipeline architecture with five modules as shown in Figure 1. Each module accomplishes one summarization process. In data preprocessing, the input dissertation abstracts in HTML format are parsed into sentences and further into a sequence of word tokens with part-of-speech tags. In macro-level discourse parsing, each sentence is categorized into one of five predefined sections using a decision tree classifier. In information extraction, important concepts are extracted from specific sections and their relationships are extracted using pattern matching. Further, research methods and contextual relations are identified using a list of identified indicator phrases. In information integration, a hierarchical concept clustering method is used to group similar concepts extracted across documents hierarchically and summarize them using a broader concept. The same types of relationships linking similar concepts are integrated and normalized using uniform expressions. Finally, the integrated information is formatted based on the variable-based framework to generate the final summaries. The system is implemented using Java and the final summaries are presented in a Web-based interface.

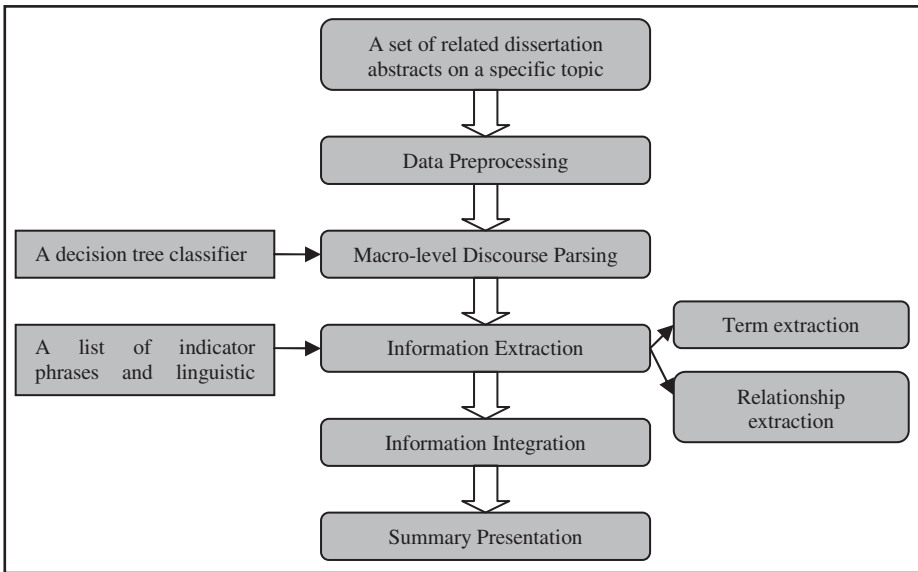


Fig. 1. Summarization system architecture

2.1 Data Preprocessing

The input files are a set of related dissertation abstracts on a specific topic retrieved from the Dissertation Abstracts International database indexed under *Sociology* and *PhD degree*. Each file contains one dissertation abstract in HTML format. First, each file was transformed into a uniform XML representation. Next, the text was segmented into sentences using a short list of end-of-sentence punctuation marks,

such as periods, question marks and exclamation points. The exclamation point and question mark are less ambiguous as end-of-sentence indicators. However, since a period is not used exclusively to indicate sentence breaks (e.g., it can be used to indicate an abbreviation, a decimal point and parts of an e-mail address), a list of common abbreviations (e.g., “i.e.”, “u.s.”) and an algorithm for detecting decimal, e-mail address and ellipsis was used to ensure more reliable identification of sentence boundaries. Finally, each sentence was parsed into a sequence of word tokens using the Conexor Parser [10]. For each word token, its document_id, sentence_id, token_id (word position in the sentence), word form (the real form used in the text), lemma (base word), and part-of-speech tag were indicated.

2.2 Macro-level Discourse Parsing

In previous work, an automatic method for parsing the macro-level discourse structure of sociology dissertation abstracts was developed by using decision tree induction to categorize each sentence into one of the five predefined sections or categories – *background*, *research objectives*, *research methods*, *research results* and *concluding remarks* [9]. The decision tree classifier made use of the normalized sentence position and single indicator words to identify the categories and obtained an accuracy rate of about 72% when applied to structured dissertation abstracts.

It was observed however that some sentences in the *research objectives* and *results methods* sections contained clear indicator phrases at the beginning of the sentences. For example, “*The purpose of this study was to investigate ...*” and “*The present study aims to explore ...*” often appeared at the beginning of sentences in the *research objectives* section, whereas “*The results indicated...*” and “*The study suggested...*” often appeared in sentences in the *research results* section. These indicator phrases can identify *research objectives* and *research results* sentences more precisely than the single indicator words used by the decision tree classifier.

Therefore, the sentence categories assigned by the decision tree classifier were adjusted further using the indicator phrases to improve the accuracy of identifying *research objectives* and *research results* sentences.

2.3 Information Extraction

Four kinds of information were extracted from the dissertation abstracts – *research variables*, *relationships between variables*, *research methods* and *contextual relations*, using indicator phrases or linguistic patterns. Research variables, research methods and contextual relations are concepts which can be extracted using term extraction. The research methods and contextual relations were identified from the extracted terms using a list of indicator phrases, whereas the variables were identified by focusing on the *research objectives* and *research results* sentences. To extract the relationships between variables expressed in the text, pattern matching was performed to identify the segments of the sentence that match with each pattern.

2.3.1 Term Extraction

After data preprocessing, a sequence of word tokens was obtained for each sentence in each document. Sequences of contiguous words of different lengths (i.e. 2, 3, 4, 5 words) were extracted from each sentence to construct n-grams (where n is the

number of words). Surface syntactic patterns were used to differentiate between terms and non-terms among the n-grams. A list of surface syntactic patterns was constructed for recognizing 1, 2, 3, 4 and 5-words terms (see Table 1 for examples).

Table 1. Some of surface syntactic patterns used for identifying terms

ID	1	2	3	4	5	Example Term
1	N					teacher
2	A	N				young child
3	N	PREP	N			ability of organization
4	N	PREP	A	N		effectiveness of early childhood
5	N	PREP	A	N	N	effectiveness of early childhood teacher

Using the surface syntactic patterns, terms of different numbers of words were extracted from the same position in the sentences. These terms represent concepts at different levels (narrow and broad concepts). For example, in the sentence “*The present study assessed the effectiveness of preschool teachers of India with respect to their interactions with young children and parents*”, one set of extracted terms were:

- 1-word terms: effectiveness, preschool, teacher, India, child, parent
- 2-word terms: preschool teacher, young child
- 3-word terms: effectiveness of preschool, teachers of India, child and parent
- 4-word terms: effectiveness of preschool teacher, preschool teachers of India, young child and parent
- 5-word terms: -

To identify the full terms in the sentences, the terms of different lengths extracted from the same position are compared and terms which cannot be covered by other terms are retained, e.g. “*effectiveness of preschool teacher*” and “*preschool teachers of India*”. Then, the terms which have overlapping tokens are connected to form a full term representing a specific full concept in the text, e.g. “*effectiveness of preschool teacher of India*”.

The concepts relating to research methods and contextual relations were identified from the full terms using a list of indicator phrases derived manually from 300 sample documents. Some of the indicator words and phrases for research methods and contextual relations are given in Table 2.

To identify the variable concepts, the full terms extracted from the *research objectives* and *research results* sentences were selected, since these two sections focus more on variables and their relationships.

Table 2. Some of indicator words and phrases for research methods and contextual relations

Types	Subtypes	Indicator words and phrases
Research methods	Research design	<i>interview, field work, survey, qualitative study</i>
	Sampling	<i>convenience sampling, snowball sampling</i>
	Data analysis	<i>univariate analysis, time series analysis</i>
Contextual relations		<i>perception, attitude, insight, perspective, view, thought, model, hypothesis, assumption, context</i>

2.3.2 Relationship Extraction

Extraction of relationships between variables involves looking for certain linguistic patterns that indicate the presence of a particular relationship. In this study, we used linear regular expression patterns. A linear pattern is a sequence of tokens each representing one or more of the following:

- A literal word in the text which has been converted to a base form (i.e. lemma);
- A wildcard which can match with one or more words in the sentence;
- A part-of-speech tag, e.g. N, V, A, ADV;
- A slot to be filled in by words or phrases in the text.

Some tokens are compulsory whereas others are optional. In this way, the patterns are more general and flexible enough to match more relationship expressions used in the text. The following is an example of a linear pattern that describes one way that cause-effect relationship can be expressed in the text:

- *[slot:IV] have <DET> (<A>) effect/influence/impact on [slot:DV]*

The tokens within square brackets represent slots to be filled by words or phrases in the text. The slots indicate which part of a sentence represents the independent variable (IV) and which part represents the dependent variable (DV) in a cause-effect relationship. The tokens within round brackets represent optional words. For example, the above pattern can match with the following sentence:

- *Changes in labor productivity have a positive effect on directional movement.*

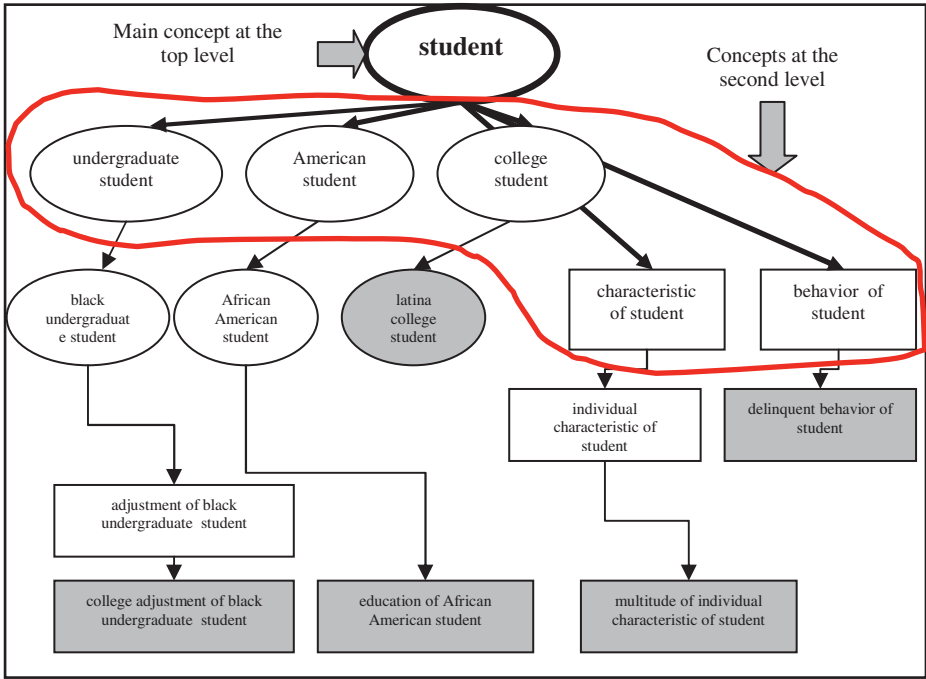
We derived 126 relationship patterns as a result of analyzing 300 sample dissertation abstracts. These patterns belong to five specific types of relationships which are often investigated in sociological research:

1. *Cause-effect relationships*: one variable causes a change or effect in another variable;
2. *Correlations*: a change in one variable is accompanied by a change in another;
3. *Comparative relationships*: There are differences between two or more variables;
4. *Predictive relationships*: One variable predicts another one;
5. *Second-order relationships*: The relationship between two or more variables is influenced by a third variable.

A pattern matching program was developed to identify the segments of the text that match with each pattern. If a text segment matches with a pattern, then the text segment is identified to contain the relationship associated with the pattern. A pattern typically contains one or more slots, and the parts of the text that match with the slots in the pattern represent the variables related by the relationship.

2.4 Information Integration

A hierarchical concept clustering method was used to group similar concepts into a tree or a hierarchy. However, concept integration is more than a simple clustering. It involves summarizing a group of similar concepts with a broader concept. Therefore, traditional clustering methods such as hierarchical agglomerative clustering were not used. Instead, we used a clustering method that links similar concepts into a hierarchical structure. It includes three phases:



• The highlighted concepts are full concepts occurring in the text.

Fig. 2. A cluster of similar concepts linked in a hierarchical structure

1. *Segment the full terms:* Each full term occurring in the text was segmented into 1, 2, 3, 4, 5-word terms and only high frequency 1-word terms above a specific threshold value were retained. Stop words and all indicator words were removed.
2. *Construct term chains:* For each frequent 1-word term, a list of concept chains was constructed by linking it with other multi-word terms in which the single word occurred as a head noun. Each chain was constructed top-down by linking the short terms first followed by the longer terms containing the short term. The root node of each chain is the 1-word term (main concept), and the leaf node is the full term (full concept). The length of the chains can be different but the maximum length is six nodes – the 1, 2, 3, 4, 5-word terms and the full terms.
3. *Build cluster tree:* All chains sharing the same root node were combined to form a cluster tree. Each cluster tree uses the root node (1-word term) as its cluster label. At the root node, two types of sublevel concepts exist – subclass concepts and facet concepts. Subclass concepts are subclasses of the main concept which are restricted or narrowed down by one or more qualifiers, while facet concepts specify various aspects (characteristics or facets) of the main concept.

In this way, similar concepts at different levels are clustered automatically. In Figure 2, the concepts in round boxes represent subclass concepts, whereas the concepts in rectangular boxes represent facet concepts. The full concepts (more specific concepts occurring in the text) are at the bottom of the cluster. In the

hierarchical cluster, concepts at the lower levels can be summarized by broader concepts at the higher levels.

The relationships between variables can be grouped according to the variable concepts they refer to. Next, the same types of relationships are converted to a uniform representation. The concepts relating to research methods and contextual relations are grouped and summarized using the indicator phrases used for identifying them. Synonyms are replaced by uniform words or phrases.

2.5 Summary Presentation

We generated interactive summaries to allow users to explore details of interest by clicking on hyperlinks rather than viewing traditional plain text summaries. Hence, the final summaries are presented in an HTML format viewable on a Web browser. It includes three hierarchies – the main summary, lists of summarized single documents sharing the same concepts, and the original dissertation abstracts. The main summary is displayed in one main window while the other two hierarchies are displayed separately in pop-up windows. In the main window, the grouped and summarized research methods, contextual relations, research variables and their relationships extracted from different documents, are integrated based on the variable-based framework (see Figure 3). For each concept, the number of documents is given in parenthesis. This is clickable and links to a list of summarized single documents sharing the given concept are displayed in a pop-up window. For each document, the title, research variables (full concepts), research methods and contextual relations are displayed. The title of the document is also clickable and links to the original dissertation abstract in a separate pop-up window.

Topic 2: summary 1 - Microsoft Internet Explorer

Address http://www.sted.com/hyper/Topic2/Topic2_summary1.html

In these 64 dissertation abstracts, the following contextual relations were found:

theory(22), context(18), perspective(18), model(15), perception(15), framework(13), view(8), insight(7), attitude(6), assumption(5), hypothesis(5), thought(2)

In these 64 dissertation abstracts, the following research methods were found:

interview(22), observation(13), qualitative research(9), survey(8), content analysis(5), case study(4), experiment(4), fieldwork(4), scale(4), comparative research(3), descriptive research(3), ethnographic research(2), interviewing(2), phenomenological research(2), quantitative research(2), regression analysis(1), archival research(1), contextual analysis(1), correlational research(1), cross-cultural research(1), discourse analysis(1), grounded theory research(1), multi-method study(1), textual analysis(1), sampling(1), secondary analysis(1), statistical analysis(1), text(1), text(1)

One of main concepts

These 64 dissertation abstracts were mainly about:

- **communication(30)**, including [intercultural communication\(30\)](#), [cross-cultural communication\(19\)](#), [inter-cultural communication\(2\)](#), [cultural communication\(2\)](#), [Hst communication\(2\)](#), [communication and identity\(2\)](#), and [more ...](#)

Different aspects were investigated, including [communication competence\(5\)](#), [communication model\(4\)](#), [communication literature\(3\)](#), [communication theory\(3\)](#), [communication behavior\(2\)](#), [communication problem\(2\)](#), [communication skill\(2\)](#), [communication strategy\(2\)](#), and [more ...](#)

The following relations were investigated:

- o There was an effect on [adaptation](#), [individual well-being among racially/ethnically diverse workgroups](#) .
- o It was affected by [values](#), [lack of intention of stay in the United States](#), [Australia](#), [Straight Talk intervention](#), [racial/ethnic identity](#), [TOEFL](#) .
- o There was a relation with [parents with high school students in special education](#), [psychological health of the workers](#), [Americans](#), [psychological health](#), [power motive](#), [general themes](#), [behaviors](#) .
- o There may be an relation with [leadership](#), [managerial control mechanisms](#), [American companies](#), [Russia](#), [respondents' level of tolerance of ambiguity](#) .
- o There was no relation with [individual's level of psychological health in an American-German workplace](#), [managerial control mechanisms](#) .

One of subclass concepts

One of facet concepts

Fig. 3. The main summary on the topic of “intercultural communication”

3 Evaluation

The purpose of the evaluation was to assess the accuracy and effectiveness of two important summarization steps – information extraction and information integration, since they influence the final output of the summarization system. In this study, we assumed that the human-generated output was the “gold standard” to be measured against. The human coders were social science graduate students at the Nanyang Technological University, Singapore.

For the evaluation of information extraction, 50 PhD sociology dissertation abstracts were selected systematically. Three human coders were asked to extract all the *important concepts* from each abstract, and from among these to identify the *more important concepts* and then the *most important concepts*, according to the focus of the dissertation research. The human-extracted concepts were used as the “*ideal*” concepts to compare against the concepts extracted automatically in three situations – concepts from *research objectives* (section 2) only, from both *research objectives* and *research results* (section 2 & 4), and from all sections.

In the evaluation of information integration, 15 topics contributed by sociology researchers were used. For each topic, a set of PhD sociology dissertation abstracts were retrieved from the Dissertation Abstracts International database by using the topic as the query and five abstracts were selected to form a document set. Moreover, another five abstracts were selected for each of five topics, including the previously chosen five abstracts, to construct a second bigger document set. From each abstract, the important concepts were extracted from the *research objectives* and *research results* automatically using our system. The human coders were asked to identify similar concepts, cluster and summarize them by assigning a category label to each cluster. Each document set was coded by two human coders. The concept clusters generated by human coders were used as the “*ideal*” clusters to compare against those generated automatically by our system.

3.1 Evaluation Results for Information Extraction

Three human coders had extracted concepts at three levels of importance. Table 3 shows the average precision, recall and F-values among the three coders. Note that the

Table 3. Average precision, recall and F-value among the three coders

Importance level		All sections	Section 2	Section 2 & 4
The <i>most important</i> concepts	Precision (%)	20.36	31.62	23.60
	Recall (%)	92.26	76.06	87.37
	F-value (%)	33.15	43.91	36.80
The <i>more important</i> concepts	Precision (%)	31.02	44.51	34.28
	Recall (%)	90.93	59.31	78.81
	F-value (%)	45.94	50.27	47.35
The <i>important</i> concepts	Precision (%)	46.12	58.84	49.68
	Recall (%)	89.99	46.63	75.66
	F-value (%)	60.40	51.53	59.34

* Section 2 refers to *research objectives*, whereas section 4 refers to *research results*.

set of *important* concepts include the set of *more important* and *most important* concepts. Similarly, the set of *more important* concepts include the set of *most important* concepts.

For all *important concepts*, the F-value obtained for *all sections* (60%) and for *section 2 & 4* (59%) were similar, and both were much better than the F-value obtained for *section 2* (52%). This suggests that important concepts are not focused only in *section 2*, but scattered in all sections. Therefore, our macro-level discourse parsing for identifying different sections of the dissertation abstract may not be helpful for identifying the important concepts.

For the *more important concepts*, the F-value obtained for *section 2* (50%) was a little higher than those for *section 2 & 4* (47%) and for *all sections* (46%). This suggests that *section 2* places a bit more emphasis on the more important concepts.

However, for the *most important concepts*, the F-value (44%) obtained for *section 2* was much higher than for *section 2 & 4* (37%) and for *all sections* (33%). This suggests that *section 2* focuses on the most important concepts, and *section 2 & 4* also can contribute to identifying the most important concepts to a less extent.

In conclusion, our macro-level discourse parsing should be helpful in identifying the more important and most important concepts. Concepts are the main elements of our system-generated summaries and accurate information extraction can result in more concise summaries which focus on the more important or most important concepts.

3.2 Evaluation Results for Information Integration

For each document set, we calculated the inter-coder similarity among the two sets of clusters created by the two human coders using a similarity measure employed by Macskassy et al. [3] as follows:

Overall similarity between coding 1 and coding 2

$$= \frac{\text{number of common same-cluster-pairs between coding 1 and coding 2}}{\text{total number of unique same-cluster-pairs obtained from coding 1 and coding 2}}$$

Similarity calculation involves first identifying all the possible pairs of terms within the same cluster (same-cluster-pairs). If the two human coders created the same clusters, then the pairs of terms obtained for both codings will be the same, and the similarity value obtained will be 1. The average inter-coder similarity obtained for the 20 document sets was a low 0.19. The value ranged from 0.04 to 0.43 across the document sets. This means that clustering is a very subjective operation.

We calculated the similarity between the sets of clusters created by our system and each of the two human coders, and obtained a higher average similarity of 0.26. This indicates that the system's clustering is more similar to each of the human coders than between the two human coders! The accurate clustering can result in a clear identification of similarities across documents in the final summaries.

4 Conclusion

This paper has outlined the design, implementation and evaluation of a multi-document summarization method for sociology dissertation abstracts. Our system focuses on extracting variables and their relationships, integrating the extracted information, and presenting the integrated information using a variable-based framework with an interactive Web interface. The summarization method employs term extraction, pattern matching, discourse parsing, and a kind of concept clustering.

Two important summarization steps – information extraction and information integration were evaluated by comparing our system's generated output against human-generated output. The results indicate that the macro-level discourse parsing was helpful in identifying the more important concepts, and the accuracy of the automatic information extraction was acceptable (46% precision and 90% recall). Information integration using the hierarchical concept clustering method generated reasonably good clusters compared to human clustering. User evaluation of the summarization method is in progress.

References

1. Boros, E., Kanto, P.B., & Neu, D.J.: A clustering based approach to creating multi-document summaries. *Document Understanding Conferences* (2002). Available at http://www-nlpir.nist.gov/projects/duc/pubs/2001papers/rutgers_final.pdf
2. Harabagiu, S.M., & Lacatusu, F.: Generating single and multi-document summaries with GISTEXTER. *Document Understanding Conferences* (2002). Available at http://www-nlpir.nist.gov/projects/duc/pubs/2002papers/utdallas_sanda.pdf
3. Macskassy, S.A., Banerjee, A., Davison, B.D., & Hirsh, H.: Human performance on clustering Web pages: A preliminary study. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press (1998) 264-268
4. Mani, I., & Bloedorn, E.: Summarization similarities and differences among related documents. *Information Retrieval*, 1(1) (1999) 1-23
5. Mckeown, K., & Radev, D.: Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM Conference on Research and Development in Information Retrieval (ACM SIGIR)*. Seattle, WA (1995) 74-82
6. National Institute of Standards and Technology. : *Document Understanding Conferences* (2002). Available at <http://www-nlpir.nist.gov/projects/duc/index.html>
7. Otterbacher, J.C., Winkel, A.J., & Radev, D.R.: The Michigan single and multi-document summarizer for DUC 2002. *Document Understanding Conferences* (2002) Available at http://www-nlpir.nist.gov/projects/duc/pubs/2002papers/umich_otter.pdf
8. Ou, S., Khoo, C., & Goh, D.: Multi-document summarization of dissertation abstracts using a variable-based framework. In *Proceedings of the 66th Annual Meeting of the American Society for Information Science and Technology (ASIST)*. Long Beach, CA, 19-23 October (2003) 230-239
9. Ou, S., Khoo, C., Goh, D., & Heng, Hui-Hing. : Automatic discourse parsing of sociology dissertation abstracts as sentence categorization. In *Proceedings of the 8th International ISKO Conference*. London, UK, 13-16 July (2004) 345-350

10. Pasi, J. & Timo, J.: A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*. Washington, DD: Association for Computational Linguistics(1997) 64-71
11. Radev, D.: A common theory of information fusion from multiple text sources step one: cross-document structure. In *Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue* (2000). Available at <http://www.sigdial.org/sigdialworkshop/proceedings/radev.pdf>
12. Radev, D., Jing, H., & Budzikowska, M.: Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation and user studies. In *Workshop held with Applied Natural Language Processing Conference / Conference of the North American Chapter of the Association for Computational Linguistics (ANLP/ANNCL)* (2000) 21-29
13. Trochim, W.: *The research methods knowledge base*. Cincinnati, OH: Atomic Dog Publishing (1999)
14. White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., & Wagstaff, K.: Multi-document summarization via information extraction. In *Proceedings of the 1st International Conference on Human Language Technology Research (HLT-01)* (2001)
15. Zhang, Z., Blair-Goldensohn, S., & Radev, D.: Towards CST-enhanced summarization. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002)*. Edmonton , Canada, August (2002)

Compressing Dynamic Text Collections via Phrase-Based Coding*

Nieves R. Brisaboa¹, Antonio Fariña¹, Gonzalo Navarro², and José R. Paramá¹

¹ Database Lab., Univ. da Coruña, Facultade de Informática,
Campus de Elviña s/n, 15071, A Coruña, Spain
{brisaboa, fari, parama}@udc.es

² Dept. of Computer Science, Univ. de Chile,
Blanco Encalada 2120, Santiago, Chile
gnavarro@dcc.uchile.cl

Abstract. We present a new statistical compression method, which we call *Phrase Based Dense Code (PBDC)*, aimed at compressing large digital libraries. PBDC compresses the text collection to 30–32% of its original size, permits maintaining the text compressed all the time, and offers efficient on-line information retrieval services. The novelty of PBDC is that it supports continuous growing of the compressed text collection, by automatically adapting the vocabulary both to new words and to changes in the word frequency distribution, without degrading the compression ratio. Text compressed with PBDC can be searched directly without decompression, using fast Boyer-Moore algorithms. It is also possible to decompress arbitrary portions of the collection. Alternative compression methods oriented to information retrieval focus on static collections and thus are less well suited to digital libraries.

Keywords: Text Compression, Text Databases, Digital Libraries.

1 Introduction

Digital libraries can be thought of as a text collection plus a set of online text search and retrieval functionalities. In many cases of interest the collection grows over time, while in others it is static. On the other hand, text compression [2] aims at representing text using less space. Storing the text collection of a digital library in compressed form saves not only space, but more importantly, disk and network transmission time. In the last decades, CPU performance has been doubling every 18 months while disk access times have stayed basically unchanged. Thus it is worthwhile to pay compression and decompression overheads in exchange for reduced disk times.

There are several challenges, however, to offer digital library services over a compressed text collection. It should be possible to carry out efficiently the

* This work is partially supported by CYTED VII.19 RIBIDI Project. It is also funded in part (for the Spanish group) by MCyT (PGE and FEDER) grant(TIC2003-06593) and (for G. Navarro) by Fondecyt Grant 1-050493, Chile.

following tasks: (1) uncompress any portion of the collection; (2) accommodate new text that is added to the collection; (3) scan text portions searching for keywords. Task (1) is related to the need to display text documents in plain form to the user. Task (2) has to do with handling growing collections. Task (3) appears when searching the text collection for words or phrases of interest. This is usually faced by means of inverted indexes that permit locating the documents of interest without traversing the text. However, indexes take additional space that must be maintained at a minimum [16,1,12]. Hence it is customary that indexes do not have all the information necessary to solve all queries (in particular phrase and proximity queries) without accessing the text. Therefore, scanning text portions is usually necessary.

Two main approaches exist to text compression. *Adaptive* compression methods, such as the well-known Ziv-Lempel family [18,19], learn the data distribution as they compress the text, continuously adapting their model of the distribution. The same model has to be learned and updated during decompression. *Semistatic* methods, such as Huffman [8], make a first pass over the text collection to build a model of the data, and compress the whole collection with the same model in a second pass.

The need of task (1) prevents the use of adaptive compression methods, as these require decompression to start at the beginning of the collection, which would make impossible to carry out local decompression efficiently. It would be possible to compress documents individually, but then compression ratios are poor because there is not enough time for the model to converge to the data distribution. This is unfortunate because adaptive compression methods would deal best with task (2), by simply appending the new text to the end of the collection and going on with the compression.

Task (3) can be faced by just decompressing the text and then searching it. A much more attractive alternative is to search the compressed text directly, without decompressing it. This saves CPU time because only a small fraction of the searched text will be finally shown to the user. Albeit there exist several techniques to search text compressed with adaptive methods (e.g. [14]), much more efficient methods have been devised for semistatic techniques. The essential reason is that one can compress the pattern and search the text for it, as its compressed form does not vary across the text. Adaptive methods, on the other hand, require keeping track of the model updates. In particular, it has been shown [5] that variants of Huffman permit searching the text up to 8 times faster than over the plain text (not even counting the time to decompress).

Classical Huffman is an unpopular choice for text compression because of its poor compression ratios. However, if the source symbols are taken to be the whole words rather than the characters [9], compression ratios (the size of the compressed text as a fraction of the original text) improve to 25%-30%, which is rather competitive. The reason is that the distribution of words is much more biased than that of characters, thanks to Zipf's Law [17]. Moreover, the source alphabet and the collection vocabulary are the same, which simplifies integration with inverted indexes [16,12]. The Huffman variants that yield those good results

in search times [5] are actually word-based. Moreover, the output is a sequence of bytes rather than bits. This worsens compression ratio a bit (it gets close to 30%), but in exchange decompression is much faster. Some variants of the format, such as *Tagged Huffman*, get compression ratios closer to 35% but ensure that the text is efficiently searchable with any text search algorithm.

The above comprises a good solution for a *static* digital library to maintain the text in compressed form, by using a semistatic compression method like word-based Huffman or a variant thereof. The compressed text takes 25%-35% of the original size, the index adds 5%-15% to this, and the search is *faster* than without compression. Hence space *and* time are saved simultaneously.

The situation is more complicated when growing collections are to be handled. Semistatic methods do not work because they rely on a global model, so in principle they need to recompress the whole text collection again. The only proposal to handle this problem [10,11] has been to build a semistatic model on the current text collection and then use it as is for the new text that arrives (new words are handled somehow), with no or very sporadic global re-compressions. This works reasonably well when the growing collection stays homogeneous, but this is not the case of most digital libraries (see next section).

In this paper, we present a modification of a statistical semistatic method, ETDC [4] (see next section), adapting it to deal with growing collections. The idea is to combine statistical compression (where variable-length codewords are assigned to fixed-length source symbols) with dictionary-based compression (where varying-length source symbols are concatenated and assigned fixed-length codewords). The resulting new method, called *Phrase Based Dense Code (PBDC)*, satisfies all the requirements of growing digital libraries and maintains the same efficiency obtained by semistatic methods. In particular, PBDC: 1) obtains good compression ratios over natural language text; 2) uses a unique vocabulary for the whole collection; 3) permits continuous increments of the compressed collection by automatically adapting the vocabulary both to new words and to changes in the word frequency distribution without degrading the compression ratio; 4) supports direct search without decompressing the text using any string matching algorithm; 5) is easily and efficiently decompressible at any arbitrary section of the text without need of performing the decompression of the whole document; and 6) uses structures easy to assemble to those of the classical inverted indexes that any digital library needs.

We present empirical data measuring the efficiency of PBDC in compression ratio, compression and decompression speed and direct search capabilities.

2 Related Work

2.1 Compressing Growing Collections

Some authors have proposed the use of semistatic statistical compression techniques such as Plain Huffman or Tagged Huffman over a first part of the collection and use the obtained vocabulary to compress the remaining text of the

collection [10,11]. That is, they propose to use the same old codewords for that words in the new text that already exist in the vocabulary, and compute new codewords for the new ones. To manage the new words that can appear, different alternatives were proposed. For example, in [10] new words are inserted at the end of the vocabulary, and new codewords are generated for them. However, changes in the word frequencies are not taken into account. In [11], new words are not inserted into the vocabulary. When one appears, it is introduced in the compressed text and marked with a special previously defined codeword. To save space those new words are compressed with a static character-based Huffman code. Again, changes in the word frequencies are not taken into account.

In both cases [10,11], authors argue that the loss of compression ratio is not significant. For example in [11], some experiments were performed over the AP archive of the TREC collection. This archive occupies 200MB. Compressing the whole file with Tagged Huffman a 31.16% compression ratio is achieved. When only a 10% of the file was compressed with Tagged Huffman and then the obtained vocabulary was used to compress the rest of the file, compressing new words with a static character oriented Huffman, the compression ratio raised to 32%. When the initial compression was performed over the 40% of the file the compression ratio became 31.5%.

These experiments were done over experimental corpora that, in our opinion, do not reproduce the situation that can be found in real digital libraries. In digital libraries, the amount of digitized text grows year after year and therefore the initial portion of the corpus, used to compute the initial vocabulary, becomes smaller and smaller. On the other hand, the vocabulary is expected not to be so homogeneous as it is inside one specific collection such as AP. In real life, new words appear and become very frequent. For example, if we think in a digital library of journals, names of politicians, artists, etc. appear when they get a noticeable position and later, after some years of having high frequency, they may disappear. The same applies to other words, names of places, events, or new technologies. For example, words such as *Web* or *Internet* have not a significant frequency in journals some years ago. This constant appearance and/or changes in frequency of words in real life could produce larger loss in compression ratio than those found in [11,10].

2.2 End Tagged Dense Code

End-Tagged Dense Code (ETDC) [4] is a semistatic compression technique, and it is the basis of the *Phrase Based Dense Code (PBDC)* we present in this paper. ETDC is an improvement upon Tagged Huffman Code [5].

Let us call Plain Huffman Code the word-based Huffman code that assigns a sequence of bytes (rather than bits) to each word. In Tagged Huffman, the first bit of each byte is reserved to flag whether the byte is the first of its codeword. Hence, only 7 bits of each byte are used for the Huffman code. Note that the use of a Huffman code over the remaining 7 bits is mandatory, as the flag bit

is not useful by itself to make the code a prefix code¹. The tag bit permits direct searching on the compressed text by just compressing the pattern and then running any classical string matching algorithm like Boyer-Moore [13]. On Plain Huffman this does not work, as the pattern could occur in the text not aligned to any codeword [5].

Instead of using a flag bit to signal the *beginning* of a codeword, ETDC signals the *end* of the codeword. That is, the highest bit of any codeword byte is 0 except for the last byte, where it is 1.

This change has surprising consequences. Now the flag bit is enough to ensure that the code is a prefix code regardless of the contents of the other 7 bits of each byte. To see this, consider two codewords X and Y , being X shorter than Y ($|X| < |Y|$). X cannot be a prefix of Y because the last byte of X has its flag bit in 1, while the $|X|$ -th byte of Y has its flag bit in 0. Thanks to this change, there is no need at all to use Huffman coding in order to maintain a prefix code. Therefore, all possible combinations of bits can be used over the remaining 7 bits of each byte, producing a *dense* encoding. This yields a better compression ratio than Tagged Huffman while keeping all its good searching and decompression capabilities. On the other hand, ETDC is easier to build and faster in both compression and decompression.

In general, ETDC can be defined over symbols of b bits, although in this paper we focus on the byte-oriented version where $b = 8$.

Definition 1. *Given source symbols with decreasing probabilities $\{p_i\}_{0 \leq i < n}$ the corresponding codeword using the End-Tagged Dense Code is formed by a sequence of symbols of b bits, all of them representing digits in base 2^{b-1} (that is, from 0 to $2^{b-1} - 1$), except the last one which has a value between 2^{b-1} and $2^b - 1$, and the assignment is done in a sequential fashion.*

That is, the first word is encoded as $\underline{1}0000000$, the second as $\underline{1}0000001$, until the 128^{th} as $\underline{1}1111111$. The 129^{th} word is coded as $\underline{0}0000000;\underline{1}0000000$, 130^{th} as $\underline{0}0000000;\underline{1}0000001$ and so on until the $(128^2 + 128)^{th}$ word $\underline{0}1111111;\underline{1}1111111$. Note that the code depends on the rank of the words, not on their actual frequency. As a result, only the sorted vocabulary must be stored with the compressed text for the decompressor to rebuild the model.

It is clear that the number of words encoded with 1, 2, 3 etc, bytes is fixed (specifically $128, 128^2, 128^3$ and so on) and does not depend on the word frequency distribution. Generalizing, being k the number of bytes in each codeword ($k \geq 1$) words at positions i :

$$2^{b-1} \frac{2^{(b-1)(k-1)} - 1}{2^{b-1} - 1} \leq i < 2^{b-1} \frac{2^{(b-1)k} - 1}{2^{b-1} - 1}$$

will be encoded with k bytes. These clear limits mark the change points in the codeword lengths and will be relevant in the PBDC that we present in this paper.

¹ In a prefix code, no codeword is a prefix of another, a property that ensures that the compressed text can be decoded as it is processed.

But not only the sequential procedure is available to assign codewords to the words. There are simple *encode* and *decode* procedures that can be efficiently implemented, because the codeword corresponding to symbol in position i is obtained as the number x written in base 2^{b-1} , where $x = i - \frac{2^{(b-1)k} - 2^{b-1}}{2^{b-1} - 1}$ and $k = \left\lceil \frac{\log_2(2^{b-1} + (2^{b-1} - 1)i)}{b-1} \right\rceil$, and adding 2^{b-1} to the last digit.

Function *encode* obtains the codeword $C_i = \text{encode}(i)$ for a word at the i -th position in the ranked vocabulary. Function *decode* gets the position $i = \text{decode}(C_i)$ in the rank for a codeword C_i . Both functions take just $O(l)$ time, where $l = O(\log(i)/b)$ is the length in digits of codeword C_i . Those functions are efficiently implemented through just bit shifts and masking.

End-Tagged Dense Code is simpler, faster, and compresses 7% better than Tagged Huffman codes. In fact ETDC only produces an overhead of about 2% over Plain Huffman. On the other hand, since the last bytes of codewords are distinguished, ETDC has all the search capabilities of Tagged Huffman code. Empirical comparisons between ETDC and Huffman can be found in [4].

3 The Phrase Based Dense Code

PBDC is a hybrid approach that requires two phases. In the first phase, the initial corpus is compressed using ETDC, which produces the initial vocabulary. Then the PBDC algorithm is used to dynamically add each new document. An important property of PBDC is that the codeword \leftrightarrow word mapping will never change once it has been defined. That is, no matter what happens later, each word in the vocabulary will be always associated with the same and original codeword it was assigned.

From now on, we will call *phrases* to our input symbols. These phrases can be either just one word (as those in the initial vocabulary) or the concatenation of two or more words.

During the addition of new documents (second phase), we look for the longest known phrase that starts at the current position. For instance, if we read the word X , and X is already in the vocabulary, we will read the next word Y to create the phrase XY and we will check if phrase XY is also in the vocabulary. If it is, the next word Z will be read and concatenated to form phrase XYZ and so on. On the other hand, if XY is not in the vocabulary then X will be the longest known phrase starting at the current position.

When the longest known phrase at the current position is found, we compress it according to three different cases. Let us call αW the sequence of words starting at the current text position, so that α is the longest known phrase we found and W is the word that follows it (note that α might be the empty string ε).

New Phrase Case ($\alpha = \varepsilon$). If the next input word W is not in the known vocabulary, then such one-word phrase W will be inserted in the vocabulary, its frequency will be set to one, and the next free codeword will be assigned to

the new phrase W from now on. In addition, this new codeword will be used to compress this first occurrence of phrase W and compression will continue with the text that follows W .

No Change Case. Phrase $\alpha \neq \varepsilon$ is already in the vocabulary. The algorithm then increases its frequency by 1. If that new frequency corresponds to a codeword of the same length as that already assigned to α , then phrase α will be compressed with its usual codeword. Compression will continue from word W , which has not yet been dealt with.

Concatenation Case. The interesting case arises when phrase $\alpha \neq \varepsilon$ is already in the vocabulary, and after increasing its frequency, it turns out that the new frequency corresponds to a codeword shorter than the one already assigned to α . In this case, α and W are concatenated to form a new longer phrase αW . This new phrase will be dealt with exactly as in the *New Phrase Case* (creating a new codeword for it, and so on), and compression will continue with the text that follows W .

Note that the idea is that, since we cannot assign a shorter codeword to a word or phrase that has increased its frequency, we opt for concatenating it with the word that follows it so that, instead of using shorter codewords, we compress more words with the codewords. This resembles the way Ziv-Lempel compression takes advantage of frequently occurring phrases in the input. Actually the algorithm has some similarity with LZ78 parsing [19].

3.1 Data Structures and Compression Procedure

The data structures used to compress, uncompress, and search, along with their functionality, are sketched in Figure 1, where the three cases are illustrated. The *vocabulary array* keeps each distinct word in the source text in a compact way (marking the end of each word with a terminator character).

The *Hash Table* is used during the compression and search process. This table keeps the source phrases αW by using two pointers in vector *phrases*. The first points to the slot in the hash table that keeps phrase α , while the second points to the position in the vocabulary vector where word W is stored. For instance, in Figure 1, phrase B is represented in slot 5, therefore the first pointer in vector *phrases* in the slot representing phrase BE (slot 7) points to slot 5, and the second pointer points to word E in the *vocabulary array*.

The hash table keeps in *freq* the phrase frequency and in *codeword* its codeword. It also maintains an array *codewordlist*, which is used only for searching and is explained in Section 3.3.

In the *Codewords Array*, each slot $i = decode(C_i)$ corresponds to codeword C_i . This array is used to decompress a document. Each slot stores a pointer to the slot in the *Hash Table* corresponding to the phrase encoded by C_i .

Let us assume that we process the first n words in the collection with ETDC, and then m new words that arrive using PBDC. The complexity of the first phase (ETDC) corresponds to: computing the vocabulary frequencies ($O(n)$), sorting the vocabulary of v distinct words ($O(v \log v)$), assigning a codeword to each

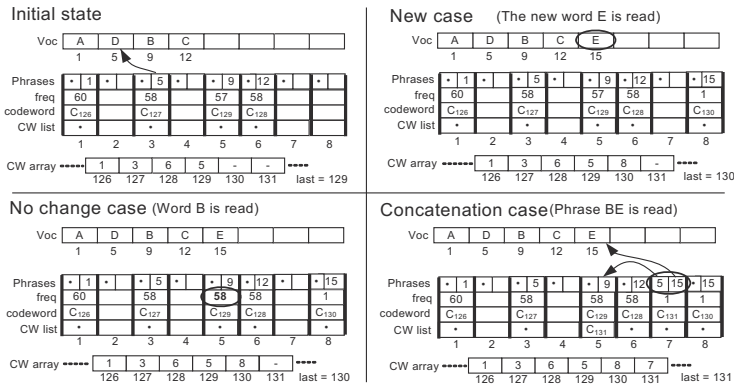


Fig. 1. Basic structures used and typical cases

word in the vocabulary ($O(v)$) and finally, compressing the text ($O(n)$). Since, empirically, $v = O(n^\beta)$ for $0 < \beta < 1$ [6], all the complexities add up $O(n)$.

The second phase (PBDC) costs $O(m)$, as we read each word at most twice (once to detect that it is not part of the next phrase, and once as the first word of the following phrase). Each new phrase requires $O(1)$ time to be dealt with. Thus, the complexity of the whole process is $O(n + m)$, linear in the whole text.

3.2 Deciding When to Create a New Phrase

There are different ways to know when a phrase α deserves a shorter codeword. For example, it is possible to keep the phrases sorted by frequency as in [3]. However, here we followed a statistical approach based on Chebyshev’s inequality.

Let us call *group* i to the set of phrases with i -byte codewords. For each such group, we maintain the average $\hat{\mu}_i$ and standard deviation $\hat{\sigma}_i$ of the frequencies in the group. If we take the frequencies of group i as samples of a random variable X_i , then $\hat{\mu}_i$ and $\hat{\sigma}_i$ are the unbiased estimators of the mean μ_i and standard deviation σ_i of X_i . Chebyshev’s inequality, which holds for any probability distribution of X_i , establishes that $Pr(|X_i - \mu_i| \leq k\sigma_i) \geq 1 - 1/k^2$ for any $k \geq 1$. We use this rule to bound the probability that a given frequency x of a phrase α belongs to group i . That is, we require that the bound tells that frequency x belongs to group i with probability at least p , for some p close to 1. By setting $k = 1/\sqrt{1-p}$, we have that $Pr(|x - \hat{\mu}_i| \leq \hat{\sigma}_i/\sqrt{1-p}) \geq p$. Therefore, only when the frequency of a phrase α becomes $x \geq \hat{\mu}_i - \hat{\sigma}_i/\sqrt{1-p}$, we will assume that α deserves belonging to group i .

Estimators $\hat{\mu}_i$ and $\hat{\sigma}_i$ are easily maintained when new phrases enter group i or when their frequencies increase. By using p values closer to 1, we are more conservative at the time of creating new phrases.

A possible problem with the compression method is that we could produce too many irrelevant phrases αW because α deserves a shorter codeword at the

time αW is read, but phrase αW will not appear again. One possible way to address this is to ensure that the frequency of α significantly exceeds what is necessary to deserve a shorter codeword. In particular, we can use a different p_i for each group i . It makes sense to be more conservative for larger i , where more useless phrases are likely to be generated.

3.3 Searching and Decompressing PBDC Compressed Text

Using ETDC, exact search of a word implies just searching for it in the vocabulary, getting its codeword, and then seeking the codeword in the compressed text using any Boyer-Moore family algorithm. However, using PBDC, a word can correspond to more than one slot of the hash table, because it can appear in one slot alone but there may also be slots storing phrases containing that word.

For this sake we maintain vector *codewordlist*. This is maintained only for slots that correspond to one-word phrases. For word X , the vector contains the list of codewords of all those phrases that include word X , for example phrases XY , YX , YXZ , etc. This list is easily updated during the compression process because each time a new phrase αW is added to the vocabulary, each of its words has just been individually read. Then, using the same hash function, the slots for all those words are efficiently found in order to update their *codewordlist* vector. The new codeword for αW is added to each of those *codewordlists*. This could be easily extended to keep track of all phrases that contain each existing pair of words, and so on.

To search the compressed corpus for a word X , a trie structure is built from the *codewordlist* vector of the slot of X . Then we apply a Boyer-Moore family algorithm such as *Set Horspool* [7,13]. More complex searches such as approximate or regular expression searching can be easily carried out by scanning the vocabulary and building the trie with all the *codewordlists* of all the matching vocabulary words.

Decompressing a PBDC is a very efficient and straightforward process. Each codeword is easily parsed due to the flag bit of each byte marking the end of each codeword. Then the codeword is transformed to a position by the decode procedure ($i = \text{decode}(C_i)$). This position is used to index the *codewords array*, and then the slot where the encoded phrase is kept in the hash table is retrieved. Finally, the *phrases* pointers are used to retrieve the phrase words one by one (right to left). Notice that the fact that a word can be encoded with different codewords in different (composed) phrases does not affect this process at all.

4 Empirical Results

We used some large text collections from TREC-2, namely AP Newswire 1988 (AP) and Ziff Data 1989-1990 (ZIFF), as well as from TREC-4, namely Congressional Record 1993 and Financial Times 1991 to 1994. We also concatenated them all, creating a corpus we called ALL, with more than one gigabyte and 885,630 different words. We used the spaceless word model [15] to create the

Table 1. Trade-off among compression ratio and number of concatenations

prob.	Phase 1 uses 10% of AP					Phase 1 uses 5% of AP				
	group 2	group 3	group 4	tot.	Ratio%	group 2	group 3	group 4	tot. conc	Ratio%
0.995	2	14,592	0	14,594	32.229	67	26,457	0	26,524	32.226
0.990	3,803	117,079	0	120,882	31.603	2,441	135,812	0	138,253	31.584
0.950	138,745	1,497,842	0	1,636,587	27.974	97,502	1,668,667	0	1,766,169	27.617
0.900	276,858	2,118,248	76,616	2,471,722	26.862	211,888	2,284,394	94,001	2,590,283	26.489

vocabulary; that is, if a word was followed by a space, we just encoded the word, otherwise both the word and the separator were encoded.

Our first experiment has to do with the number of phrases generated depending on parameter p of Section 3.2. We used AP corpus, of 238 megabytes and 269,141 words. The two sets of results were obtained using 10% and 5% of AP for the first phase (ETDC), respectively. The number of phrases produced in groups 2, 3, and 4, depends on the probabilities p_i we use. Table 1 shows the results. Group 4 does not have concatenations because there are no phrases encoded with 4-byte codewords except in the least conservative case. Notice how the compression ratio improves as the number of concatenations grows when we are less conservative and use $p = 0.9$ instead of $p = 0.99$. Of course this must be weighted against the larger vocabulary of phrases we must store.

We focus now on comparing PBDC against alternative approaches, where small portions of the ALL corpus are used to initialize the model and then all the rest of the corpus is added. In our case, we compress the first small part with ETDC and the rest with PBDC. We test the latter with two groups of p_i parameters: $p_2 = 0.9$, $p_3 = 0.99$ and $p_4 = 0.999$, and the more conservative $p_2 = 0.99$, $p_3 = 0.999$ and $p_4 = 0.9999$.

We also use ETDC in a *No-Concatenating mode*. This is exactly what is proposed in previous work [10,11] to handle growing collections, just using ETDC instead of Huffman. That is, during the second phase, new words are added to the vocabulary, but changes in the word frequency distribution are not considered.

Table 2 presents the results. Each row shows the portion (1%, 5% and 10%) of the corpus compressed during the first phase. The last row is a special case. It shows the resulting data when the semi-static ETDC approach was used over the whole corpus.

Columns 2 and 3 give the size in kilobytes and the number of words of the initial text. The fourth column shows the compression ratio achieved with ETDC using the *No-Concatenating mode* (previous work). Columns 5 to 7 and 8 to 10 show compression ratio, number of concatenations and number of phrases, respectively, with the two settings for p_i . The number of phrases is that of concatenations plus the number of single words. This latter number can be smaller than the vocabulary size because in PBDC some words may appear only as part of phrases.

The table shows that the compression ratio is always better using PBDC. We must notice immediately, however, that this result is misleading as it is not considering the size of the vocabulary, which is much larger with PBDC. Although vocabularies are usually kept in main memory, a version of them (maybe

Table 2. Compression of ALL Corpus (1,080,720,304 bytes). We do not count the vocabulary sizes.

phase 1			No conct.	Phs 2 (0.9;0.99;0.999)			Phs 2 (0.99;0.999;0.9999)		
% ALL	K bytes	#vocab	ratio	PBDC %	#concat	#phrases	PBDC %	#concat	#phrases
1%	10,807	67,559	34.831%	28.577%	2,916,577	3,752,456	30.957%	941,131	1,803,325
5%	54,036	130,585	34.636%	28.949%	2,844,572	3,684,929	31.033%	916,273	1,782,660
10%	108,072	178,050	34.607%	29.117%	2,717,952	3,558,747	31.020%	878,028	1,740,197
100%	1,080,720	885,630	32.877%	32.877%	0	885,630	32.877%	0	885,630

Table 3. Compression of ALL Corpus (1,080,720,304 bytes). We count the size of the vocabularies.

phase 1			No conct.	Phs 2 (0.9;0.99;0.999)			Phs 2 (0.99;0.999;0.9999)		
% ALL	K bytes	#vocab	ratio	PBDC %	#concat	#phrases	PBDC %	#concat	#phrases
1%	10,807	67,559	35.611%	31.246%	2,916,577	3,752,456	32.346%	941,131	1,803,325
5%	54,036	130,585	35.416%	31.571%	2,844,572	3,684,929	32.406%	916,273	1,782,660
10%	108,072	178,050	35.387%	31.657%	2,717,952	3,558,747	32.369%	878,028	1,740,197
100%	1,080,720	885,630	33.657%	33.657%	0	885,630	33.657%	0	885,630

compressed in some form) must be stored in secondary memory and accounted for in the final space requirement of the method.

A possible storage method for vocabularies is as follows. The *No-Concatenating* mode using ETDC only needs to store a vocabulary array where words must be sorted in codeword order. The vocabulary needs 8,428,001 bytes (a 0.780% of the text). In the PBDC approach, an array with as many entries as phrases must be stored. The entries must be sorted in codeword order. The first byte of each entry is used to encode whether the entry is a single word or a multi-word phrase. If it is a word entry, the byte will give its length, but if it is a phrase entry that first byte will be 0. In the phrase type, after this first byte, the entry has 2 pointers using 3 bytes each, corresponding to the *phrases* entries. Therefore an overhead of 7 bytes for each concatenation must be paid for the PBDC approach in addition to the vocabulary of the ETDC.

The space needed to store frequency information can be reduced by using an appropriate compressor, as most phrases with long codewords will share low frequencies. Just applying classic Huffman entails an overhead of less than 1 byte per phrase. A more sophisticated approach, encoding frequencies (as if they were word ranks) with ETDC and then applying classic Huffman to the output bytes reduces the overhead to 0.25 bytes per phrase.

If the vocabulary is stored with the compressed text and we want to restart the system for the digital library, all the structures shown in Figure 1 will need to be rebuilt. Notice that the codeword list of each slot can be easily obtained because each phrase is linked with all the former ones in a way that all the single words can be reached.

Table 3 shows the same results, now considering the vocabulary sizes. It can be seen that PBDC significantly outperforms previous approaches (the *No-Concatenating* mode) by around 12%. It can also be seen that the least conservative approach works better even when it has to pay for the larger vocabulary

generated. Finally, we note that PBDC works *better* when it uses a smaller initial ETDC phase, which suggests that it could be used as a replacement of ETDC in general, not only with the aim of updating the text collection.

5 Conclusions and Future Work

We have presented *Phrase Based Dense Code (PBDC)*, a new compression method that is useful for digital libraries because it reaches good compression ratios that do not degrade when the text collection grows. On the other hand, PBDC allows direct search over the text and efficient decompression of arbitrary portions of the collection.

PBDC is a hybrid method because it takes advantage not only of the word frequency distribution but also of the co-occurrence of words. Although more tuning and experiments have to be performed in real digital library scenarios, and better implementations of PBDC have to be developed, the promising empirical results show that this hybrid approach can lead to new compression methods. We emphasize that the idea of inserting phrases as source symbols of a statistical encoder has independent interest and can be used in broader scenarios.

It is clear that more experimentation is necessary to optimize the probability parameters that obtain the best compression keeping a relatively low number of concatenations, and more research must be done to have better criteria about when a new concatenation is going to be useful. One possibility is to look for the probability of the following word, because if that is not a common word, the new phrase will probably not appear ever again. It would also be interesting to explore the use of linguistic techniques to choose phrases with higher probability of future occurrence.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. AW, 1999.
2. T. C. Bell, J. G. Cleary, and I. H. Witten. *Text Compression*. P.Hall, 1990.
3. N. Brisaboa, A. Fariña, G. Navarro, and José R. Paramá. Simple, fast, and efficient natural language adaptive compression. In *Proceedings of the 11th SPIRE*, LNCS 3246, pages 230–241, 2004.
4. N.R. Brisaboa, E.L. Iglesias, G. Navarro, and José R. Paramá. An efficient compression code for text databases. In *25th ECIR*, LNCS 2633, pages 468–481, 2003.
5. E. Silva de Moura, G. Navarro, N. Ziviani, and R. Baeza-Yates. Fast and flexible word searching on compressed text. *ACM TOIS*, 18(2):113–139, 2000.
6. H. S. Heaps. *Information Retrieval: Computational and Theoretical Aspects*. Acad. Press, 1978.
7. R. N. Horspool. Practical fast searching in strings. *SPE*, 10(6):501–506, 1980.
8. D. A. Huffman. A method for the construction of minimum-redundancy codes. *Proc. Inst. Radio Eng.*, 40(9):1098–1101, 1952.
9. A. Moffat. Word-based text compression. *SPE*, 19(2):185–198, 1989.
10. A. Moffat, J. Zobel, and N. Sharman. Text compression for dynamic document databases. *KDE*, 9(2):302–313, 1997.

11. E. Moura. *Compressao de Dados Aplicada a Sistemas de Recuperacao de Informacao*. PhD thesis, Universidade Federal de Minas Gerais, Brazil, 1999.
12. G. Navarro, E.S. de Moura, M. Neubert, N. Ziviani, and R. Baeza-Yates. Adding compression to block addressing inverted indexes. *IR*, 3(1):49–77, 2000.
13. G. Navarro and M. Raffinot. *Flexible Pattern Matching in Strings – Practical on-line search algorithms for texts and biological sequences*. CUP, 2002.
14. G. Navarro and J. Tarhio. Boyer-Moore string matching over Ziv-Lempel compressed text. In *Proc. 11th CPM*, LNCS 1848, pages 166–180, 2000.
15. E. Silva de Moura, G. Navarro, N. Ziviani, and R. Baeza-Yates. Fast searching on compressed text allowing errors. In *Proc. 21st SIGIR*, pages 298–306, 1998.
16. I.H. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kauffman, 1999.
17. G.K. Zipf. *Human Behavior and the Principle of Least Effort*. AW, 1949.
18. J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE TIT*, 23(3):337–343, 1977.
19. J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE TIT*, 24(5):530–536, 1978.

Does eScience Need Digital Libraries?

Tamara Sumner¹, Rachel Heery², Jane Hunter³, Norbert Lossau⁴,
and Michael Wright⁵

¹ Dept of Computer Science, University of Colorado, Campus Box 430
Boulder, CO, USA 80309-0430
summer@colorado.edu

² UKOLN, University of Bath, Bath BA2 7AY, UK
r.heery@ukoln.ac.uk

³ DSTC, Brisbane, Australia 4072,
jane@dstc.edu.au

⁴ Bielefeld University / Library, Germany
norbert.lossau@uni-bielefeld.de

⁵ DLESE Program Center, University Corporation for Atmospheric Research, PO Box 3000,
Boulder, CO, 80307-3000, USA
mwright@ucar.edu

Abstract. eScience has emerged as an important framework for dramatically rethinking the conduct of scientific research using information technology. There is an unparalleled opportunity for the international eScience and digital library communities to create shared infrastructure to support the conduct of science from end-to-end; i.e., from hypothesis generation, to collecting and analyzing scientific data, to the reporting of research outcomes, and the inclusion of scientific data and models in teaching and learning processes. For this vision to be realized, the two communities must establish a shared vision and research agenda encompassing several critical dimensions, including differences in theoretical and methodological approaches, and collaboration goals. Additionally, for the benefits of eScience and digital libraries to be fully realized, it is vital to establish a shared vision of the broader impact of this work for educators, learners, and the general public.

1 Introduction

There are an increasing number of initiatives in several countries targeted at supporting research into new forms of computational infrastructure intended to transform the conduct of scientific research in areas such as chemistry, atmospheric science, and earth science. These initiatives, which go under a variety of names including eScience, eResearch, and cyberinfrastructure, are a response to the changing nature of scientific research, particularly in the natural and physical sciences, which is increasingly dependent upon large data sets and high-end analysis and visualization tools. Research issues being addressed in these initiatives include information retrieval, information modeling, ontologies, systems interoperability, and policy issues associated with providing transparent access to complex data sets. As such, these initiatives are concerned with many of the same research issues that the international digital library community has been grappling with for the past decade.

It is timely to seriously consider the role that digital libraries can and should play in this emerging eScience computational infrastructure. Bringing the digital library and the emerging scientific infrastructure worlds together can lay the foundation for providing truly integrated support for the entire process of science, from formulation of research questions to the publication of the outcomes. Specifically, the eScience and digital libraries research communities need to work together to identify the potential contributions of each of these communities for supporting the conduct of science and to articulate a shared research agenda. Critical questions to consider include:

- What will the ‘knowledge of science’ look like in 20 years? Will publications still be the coin of the realm or will annotated data sets, scientific models, visualizations, and other new forms of intellectual product, become the predominant mode of knowledge sharing? What is the future role of digital libraries in the support of science? What are the success stories we want tell about the influence of digital libraries on the conduct of science in 20 years?
- What are the critical areas – theoretical, methodological, social or technical – where collaboration across the two communities (Digital Libraries and eScience) is needed to support the entire process of science from inception, to implementation, and publication? What are the core elements of a long-term shared research agenda across digital libraries and eScience? What are the roadblocks, or core differences between these two communities’ approaches, that may hinder progress and collaboration? How will having digital library collections and services integrated with advanced real-time collaborative systems, such as those being created in eScience initiatives, change the scientific process?
- Can we really build significant international, sustained collaborations around a shared agenda? After years of international workshops, conferences, etc., we still can’t even agree what to call it (eScience, Cyberinfrastructure, eResearch), much less work together effectively. What should we do, can we name specific activities, to promote the necessary international dialog between researchers?
- What are the broader impacts of this agenda for science educators, learners, and the public? How can eScience and digital libraries transform teaching and learning? How can the needs and desires of science education and public science literacy inform this research agenda?

This panel debate will address the contribution digital library technologies can make to the changing nature of science and scientific data curation. The panel offers an opportunity for digital librarians and computer scientists, particularly those working on data management and archiving, to consider their response to the challenges of developing infrastructure to support increasing volumes of scientific data. Closer collaboration with research scientists and educators offers potential for new approaches and services. This panel will progress formulation of a research agenda to support these developments.

Digital Libraries over the Grid: Heaven or Hell? (Panel Description)

Donatella Castelli¹ and Yannis Ioannidis²

¹ ISTI - CNR, Pisa, Italy

`Donatella.castelli@isti.cnr.it`

² Dept. of Informatics and Telecommunications,
University of Athens, Hellas (Greece)

`yannis@di.uoa.gr`

The last decade has seen unprecedented advances in network and distributed-system technologies, which have opened up the way for the construction of global-scale systems based on completely new conceptions of computation and sharing of resources. The dream of integrating unlimited levels of processing power, unlimited amounts of information, and an unlimited variety of services, and offering the entire package in a reliable and seamless fashion to widely distributed users is quickly becoming reality. As Digital Libraries move towards more user-centric, pro-active, collaborative functionality and application diversity, they should be among the first to take advantage of such environments. The long-term vision of the field for creating Dynamic Universal Knowledge Environments calls for intensive computation and processing of very large amounts of information, hence, the needs for the appropriate distributed architecture are pressing.

Grid technologies are at the forefront of these developments. While much has been written about computation in the Grid environment, information and service management of the kind required by Digital Libraries has received very limited attention in the literature. Nevertheless, the Grid offers tremendous opportunities in that direction and at the same time poses major technical challenges in the area as well. The goal of this panel discussion is to identify these opportunities and challenges and examine whether the positive aspects of the Grid outweigh the negative ones or vice versa. In this direction, the panelists are called to answer some of the following questions:

- Is there any benefit in using the Grid technologies for supporting Digital Libraries? Are there new key Digital Library functionalities that can be enabled by the use of these technologies? Are there application areas that may profit from Digital Libraries on the Grid?
- Is there new research to be done for Digital Libraries over the Grid? Are there any new problems that arise from managing general documents over the Grid? For example, are there new problems with respect to security, heterogeneous information integration, document search, or workflow management?

- Do classical problems require new solutions or do conventional approaches work well in the Grid environment? For example, how does one address issues of document indexing, information retrieval, or document composition?
- How does the Grid compare with other architectures, e.g., peer-to-peer or service-oriented architectures for Digital Libraries?
- Is the Grid middleware already developed or under development (e.g., Condor, Globus, Unicore) adequate for supporting the required Digital Library functionality? What about the currently forthcoming distributed infrastructures, e.g., EGEE?
- Are there any particular difficulties when dealing with management of any particular information form expected to be found in Digital Libraries over the Grid, e.g., differences between relational and XML data or free text, or differences between cultural information and health information?

Management and Sharing of Bibliographies

Erik Wilde, Sai Anand, and Petra Zimmermann

Swiss Federal Institute of Technology, Zürich (ETHZ),
ETH-Zentrum, 8092 Zürich, Switzerland

Abstract. Managing bibliographic data is a requirement for many researchers. The ShaRef system has been designed to fill the gap between public libraries and personal bibliographies, and provides an open platform for sharing bibliographic data among user groups.

1 Introduction

Digital libraries and access technologies have made it very easy for researchers to find relevant information resources. However, once these resources have been found, it is still a challenge to store and manage them in a structured way, and even more so in a group setting with diverse requirements in terms of operating systems, document processing software, and other information management tools. The *Shared References (ShaRef)* [1,2] system provides a solution to these problems. Using ShaRef, research groups can improve the internal information flow, and they can also improve the reuse of information, for example for reading lists for lectures, or for yearly publication lists of a research group.

Merging personal information management with collaboration features means that the role of annotations becomes more important, because annotations made by one user can be helpful to other users, and a collaborative system supporting this kind of information exchange enables users to share information in an easily understandable and useful way. The ShaRef system is designed to support this informal way of collaboration, where the exact workflow of collaboration is left open, but the data model is sufficiently rich to capture the data necessary for sharing information.

2 Design and Architecture

The focus of ShaRef is the collaborative management of reference information. ShaRef's data model has been designed with the end user in mind, which means that it primarily focuses on end users rather than library needs. One key part in the data model is the question of how to model bibliographic references, and in addition to that, individual and collaborative features have to be covered such as annotations, group management, and access control.

The data model is defined as an XML Schema, which makes it easy to import and export data because of the wide adoption of XML. This schema defines bibliographies to be collections of *references*, *shadows*, and *keyword* and *association*

definitions. A *shadow* is a reference to a reference, thus, it is a way how users can mirror other references without actually copying them (this is important for keeping references consistent). Keywords and associations are used for describing references and relationships between references.

ShaRef is a Java-based system. The system architecture has been designed to support different usage scenarios, with the goal to make the system as flexible as possible. In most cases, ShaRef is used as an online client with a Java GUI, where parts of the application logic are located within the client. In offline mode, all data is stored locally and handled by a Java DBMS. The online and offline modes are based on the assumption that the ShaRef client is installed on the local system. The thin client is an alternative which on the client side only requires a Web browser, but — due to the limitations of Web interfaces — offers a less comfortable interface than the Java client.

One of the main goals of ShaRef is to avoid lock-in and to keep the platform as open as possible. In the user study before the project start, one of the most frequent comments was that it was unacceptable if a system for bibliography management was designed in a way which made it hard or impossible to import and export data. Many researchers tend to keep their bibliographies over the course of several years, often throughout the whole career. Thus, a system forcing onto them a model into which they cannot import their existing data, and from which they cannot detach when they want to do so, would not be acceptable.

3 Related Work

When comparing work on personal bibliography management tools with work on improving access technologies to digital libraries, it can be seen that surprisingly little effort goes into this area of research. However, apart from personalization features of library access systems (which is a separate field of research), there are also some systems which are more specifically geared towards personal tools for reference management.

The *Kepler* system chooses a different approach than ShaRef, because it focuses on making the personal bibliographies available as publicly accessible catalogs through OAI-PMH. Kepler enables users to collaborate through aggregating data in a *Group Digital Library (GDL)*. CDS and DSpace are other examples for systems with collaboration features, and *Reference Manager* is a commercial product offering collaboration features.

References

1. Wilde, E.: References as Knowledge Management. Issues in Science & Technology Librarianship (2004)
2. Wilde, E.: Shared Bibliographies as Hypertext. Technical Report TIK Report No. 224, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zürich, Switzerland (2005)

Legislative Digital Library: Online and Off-line Database of Laws

Viorel Dumitru, Adrian Colomitchi, Eduard Budulea, and Stefan Diaconescu

SOFTWIN, Str. Fabrica de Glucoza Nr. 5, Sect.2, 020331 Bucharest, Romania
{vdumitru, acolomitchi, ebudulea, sdiaconescu}@softwin.ro

Abstract. The paper presents the main issues that usually appear in the development of a legislative digital library. The great number of legislative documents which accumulates over the time raises the need for electronic management of this content and the meta-information associated with it. The preparation, the management and the distribution to end users are explained in detail in this paper, offering in the same time an architectural solution for the development of a similar library. A big emphasis was putted on the legislative documents automatic reference linking mechanisms.

1 Introduction

Based on the definition from [1] - "a focused collection of digital objects, including text, video and audio, along with methods for access and retrieval, and for selection, organization and maintenance of the collection" - it can be imagined all the problems which may be encountered during the building of a legislative digital library. Taking into account that the laws are published especially on periodicals, this library must be able to import a digitized form of the laws and to be quickly updatable, at least with the same frequency as the printing version.

The need for quick and selective access to the legislation requires the digital library to offer end users features as search and retrieval the laws, benefit of the links between laws and other meta information added by the jurist user of the library.

The paper presents the features a digital library must have in order to achieve these objectives: a content production flow, a search and retrieval mechanism, some methods for accessing and linking the references and a content distribution system.

We exemplify the digital library features by the digital library that was built for the Romanian legislation and that includes all the laws and other legislative documents published by the Official Gazette of Romania between 1989 and 2003 (more than 90.000 printed pages, A4 format, and more than 460.000 links).

2 The Application Modules

From the functional point of view the software application of the digital library must have four major modules:

1. The content production flow which feeds the entire system: we suppose it has QuarkXPress files for example (QXD - see [?]) as input and XML files (eXtended Mark-up Language - see [?]) as output with custom structure described by a DTD file (Document Type Definition);
2. The content management module: includes all the functions of thematic classification and juridical linking of the laws;
3. The content distribution system: it selects, prepares and distributes the content to end users;
4. The system administration module.

These modules must be integrated into the application framework. Except the content production flow which has specific subsystems, all the other modules have the same graphical user interface.

From end users point of view, the software application must include:

1. A public web site (online access) for the anonymous access and for the registered users and subscribers;
2. A web site for administration and content management which covers the jurists, the sales men and the administrator activities;
3. A windows based software solution to produce the legislative XML files (a XML editor) for the production flow operators (XML structuring professionals).

3 Conclusions

This paper has presented the main needs and possible solutions for architecture of a digital legislative library.

This experience can be extended further on by considering also other type of legislative content such as jurisprudence, law dictionaries and even additional value added functionalities (examples: creating a virtual edition of a law document by inserting the new modifications to the already officially published document, virtual collaborative workplace for end users etc.).

The main outcome resulted from a digital library remains the fast, accurate and selective access to the legislative content plus the unique possibility for references based navigation into the complex environments created by the laws.

References

1. Witten, I.H., Bainbridge, D.: How to Build a Digital Library, Morgan Haulmann Publishers, (2003)
2. Constantopoulos, P., Solvberg, I.: Research and Advanced Technology for Digital Libraries, Springer, (2001)
3. Bergmark, D.: Automatic extraction of reference linking information from online documents Tech. Rep. TR 2000-1821, Cornell Computer Science Department, (2000)

DIRECT: A System for Evaluating Information Access Components of Digital Libraries*

Giorgio Maria Di Nunzio and Nicola Ferro

Department of Information Engineering – University of Padua,
Via Gradenigo, 6/b – 35131 Padova – Italy
{dinunzio, ferro}@dei.unipd.it

Digital Library Management Systems (DLMSs) generally manage collections of multi-media digitalized data and include components that perform the storage, access, retrieval, and analysis of the collections of data. Recently, the new trend of DLMS applications is pushing towards a components/services technology which is becoming more and more standardized [1,2]. The results of this new orientation are ad-hoc solutions for different components and services of DLMS: the data repository, the data manager, the search and retrieval components, etc. We are particularly interested in the evaluation aspects that range from measuring and quantifying the performances of the information access and extraction components of a DLMS to designing and developing an architecture for a system capable of supporting this kind of evaluation in the context of DLMSs [3,4].

An innovative system named *Distributed Information Retrieval Evaluation Campaign Tool (DIRECT)* is being designed and developed with the aim of integrating the activities among the different entities involved in the evaluation of the information access components of a DLMS. Today, this type of evaluation is carried out in important international evaluation forums which bring research groups together, provide them with the means for measuring the performances of their systems, discuss and compare their work [5,6,7,8]. The main aim of DIRECT is to create a unified view of these kind of evaluation forums especially for the management of the results and the analysis of data. We consider this last point (data analysis) a key issue for DIRECT; in fact, since generally evaluation forums do not further the systematical employment of statistical analysis from participants, the important innovative aspect of DIRECT is to provide participants with a uniform way of performing statistical analysis on their results. In this way, not only participants benefit from standard experimental collections, that make the experimental results comparable, but also they may exploit standard tools for the analysis of the experimental results, that make the analysis and assessment of experimental results comparable too. Moreover, giving to participants the possibility to interact with other participants' performances would certainly improve the quality of research.

DIRECT will mainly support the storage and preservation of data together with its retrieval and management. The concept of data in evaluation forums

* The authors wish to thank Luca Pretto and Franco Crivellari for the time spent discussing the entity-relationship schema. The work is partially supported by the DELOS Network of Excellence on Digital Libraries, as part of the Information Society Technologies (IST) Program of the European Commission (Contract G038-507618).

may vary from textual documents to statistical analysis of performances, and, in this work, we concentrate mainly on the following aspects:

- the management of an evaluation campaign from the point of view of organizers. The edition, the publication of the act of the workshop, the track set-up, the harvesting of documents;
- the management of users/participants of an evaluation campaign. The registration of the groups, the permission settings;
- the management of pooling assessment. The choice of the assessors, the selection of documents to be assessed, the gathering of assessed documents;
- the management of submission of runs and the validation of submitted runs;
- the management of statistical analysis of participants performances. The choice of the statistical test;
- the management of logs.

The architecture of DIRECT consists of three layers – data, application and interface logic layers – which allow one to achieve a better modularity and to properly describe the behavior of DIRECT by isolating specific functionalities at the proper layer. Moreover, this decomposition makes it possible to clearly define the functioning of DIRECT by means of communication paths that connect the different components. In this way, the behavior of the system is designed in a modular and extensible way.

Since we are going to provide and manage the technical infrastructure, both hardware and software, for the *Cross-Language Evaluation Forum (CLEF) 2005* ongoing evaluation campaign, the possibility of testing and evaluating the DIRECT system in real settings will be exploited.

References

1. DELOS: Newsletter – Issue 2. <http://www.delos.info/newsletter/issue2/> [last visited 2004, November 22] (2004)
2. Rauber, A.: DELOS and the Future of Digital Libraries. *D-Lib Magazine*, 10(10). <http://www.dlib.org/dlib/october04/10contents.html> [last visited 2004, November 22] (2004)
3. Agosti, M., Di Nunzio, G.M., Ferro, N.: Evaluation of a Digital Library System. In Agosti, M., Fuhr, N., eds.: Notes of the DELOS WP7 Workshop on the Evaluation of Digital Libraries, http://dlib.ionio.gr/wp7/workshop2004_program.html (2004) 73–78
4. Di Nunzio, G.M., Ferro, N.: DIRECT: a Distributed Tool for Information Retrieval Evaluation Campaigns. In: Proc. 8th International Workshop of the DELOS Network of Excellence on Digital Libraries on Future Digital Library Management Systems (System Architecture & Information Access). (2005) (in print)
5. Text REtrieval Conference (TREC). (<http://trec.nist.gov>)
6. Cross Language Evaluation Forum (CLEF). (<http://clef.isti.cnr.it>)
7. NII-NACSIS Test Collection for IR Systems (NTCIR). (<http://research.nii.ac.jp/ntcir/index-en.html>)
8. Initiative for the Evaluation of XML Retrieval (INEX). (<http://inex.is.informatik.uni-duisburg.de>)

Modular Emulation as a Viable Preservation Strategy

Jeffrey van der Hoeven and Hilde van Wijngaarden

Koninklijke Bibliotheek, The National Library of the Netherlands,
The Hague, The Netherlands
Jeffrey.vanderhoeven@kb.nl, Hilde.vanwijngaarden@kb.nl

Abstract. Emulation is the only strategy to ensure long-term access to digital objects in their original environment. The National Library of the Netherlands (KB) and the Nationaal Archief of the Netherlands believe that emulation-based preservation is worth developing and has to be tested. This short paper proposes a new model for emulation called modular emulation that will allow us to develop a working prototype for the rendering of digital objects in the future.

1 Why Emulation?

Digital preservation does not end with the careful storage of digital objects. With its e-Depot [1] in place, the Koninklijke Bibliotheek (KB) ensures that digital objects are safely stored for the long term. However, keeping them accessible requires a continuous effort working out strategies for permanent access. The usability of digital objects is threatened by rapid innovations in computer technology. Therefore strategies have to be developed to ensure access to digital objects for the long term. Preservation strategies can be divided into two groups: migration and emulation. The problem with migration is that it often creates small errors and that it is not a one-time event, but has to be applied for all objects periodically with an increasing risk of error-propagation. Moreover, migration may cause loss of functionality. Emulation recreates the original environment in which digital objects can be rendered in their authentic form, without the need for periodic conversions. This way the original object is preserved as well as the functionality offered by the environment. When functionality matters, emulation is the only effective strategy. Existing emulators - created for different purposes - prove that emulation is not too complex and can be cost-effective.

2 Modular Emulation

Based on the results of emulation research at the KB and inspired by existing ideas of both Jeff Rothenberg's virtual machine approach [2] and Raymond Lorie's Universal Virtual Computer (UVC) [3], a new model on emulation has been developed: modular emulation. This model (see figure 1) focuses on the recreation of the original hardware environment in which digital objects can be executed together with their system and application software. This way, PDF documents, databases and interactive multi-media applications can be rendered authentically.

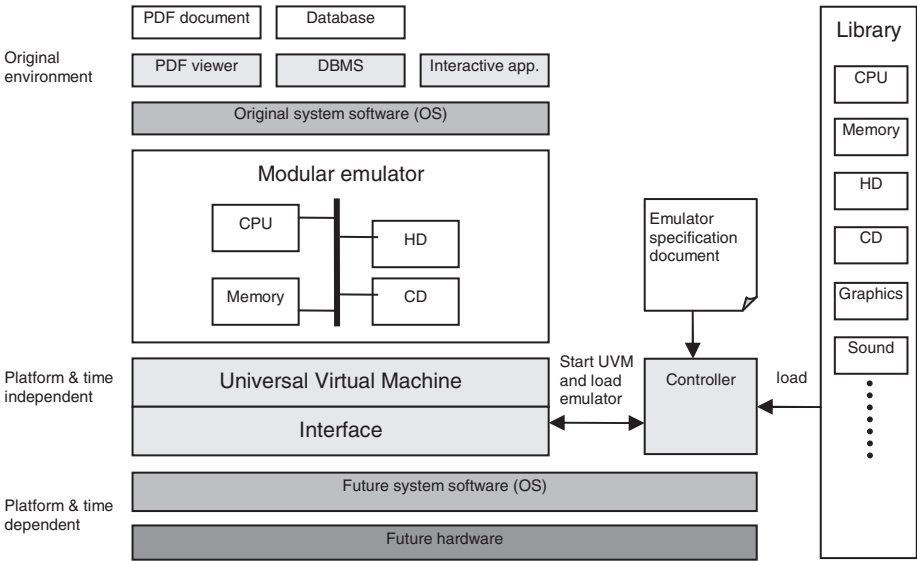


Fig. 1. Conceptual model of modular emulation

The core of this model is defined by the modular emulator that is able to run the original system and application software. The emulator consists of distinct modules, each of them emulating specific hardware functionalities, like a CPU, memory or storage device. Each module can be reused and rearranged to create different emulators. All modules are preserved in a module library. Based on an emulator specification document that defines which modules should be used, the controller loads the required modules and creates a new emulator on the fly. On top of future hard- and software, a Universal Virtual Machine (UVM) will create the desired platform independency. In turn, the UVM will execute the modular emulator.

In 2005 and 2006, an actual modular emulator will be developed incrementally by the KB and the Nationaal Archief of the Netherlands. This emulator will be platform dependent at first, but will later be extended to run on a UVM.

References

1. Oltmans, E., van Wijngaarden, H.: Digital Preservation in practice: the e-Depot at the Koninklijke Bibliotheek. In: VINE, vol. 34 (1), (2004), 21-26
2. Rothenberg, J.: An Experiment in Using Emulation to Preserve Digital Publications. Koninklijke Bibliotheek, The Hague, The Netherlands, 2000. Available at: <http://www.kb.nl/coop/nedlib/results/emulationpreservationreport.pdf> (accessed 9 June 2005)
3. Lorie, R.A.: Long-term archiving of digital information. IBM Research report, IBM Almaden Research Center, San Jose, Almaden, 2000.

Retrieving Amateur Video from a Small Collection

Daniela Petrelli¹, Dan Auld¹, Cathal Gurrin², and Alan Smeaton³

¹ Information Studies, University of Sheffield, Regent Court, 211 Portobello St.
Sheffield S1 4DP, UK
{d.petrelli@shef.ac.uk}

² Centre for Digital Video Processing, Dublin City University,
Glasnevin, Dublin 9, Ireland

1 Amateur and Historical Videos

Research on digital video libraries has been done in extensive and expensive projects (e.g. Open Video Project [1], Físchlár [2], Informedia [3]). Small video collections have small budgets and cannot afford sophisticated techniques to put their material on-line. Though very basic digital video library features can be good enough for enlarging the access to rarely seen material, e.g. folklore films from the 1920's to the 1990's owned by the National Centre for English Cultural Tradition (NATCECT). This material is unique but rarely used as the archive opens few hours a week: digital access would make it widely available to scholars, students, and enthusiasts.

Part of the NATCECT video collection was digitized (56 videos, 37 hours); movies were often of poor quality (recorded by amateurs and copied many times) and advanced video retrieval techniques could not be used; e.g. audio indexing via speech recognition was useless for silent or musical videos. Boundary detection was used to automatically generate summaries of the visual content in the form of sequences of images; the poor quality of the video source produced a high number of corrupted images requiring an initial selection. A short description of the video content and a set of archival information were manually added as metadata to support textual search.

2 User Evaluation and Focus Group

Two interfaces were compared in a counterbalanced within-subject user test with 24 participants: *browsing* listed videos grouped by archival categories (e.g. childlore, fixed dates); *searching* supported free text queries and filters selection. Tasks were topical search (retrieve relevant videos) and keyframe detection (find shown images). The interaction was recorded to extract objective data (e.g. task completion time); questionnaires and interview were used to collect subjective data (e.g. personal view).

In average search was more efficient and effective, however failures in browsing were due to the experimental time expiring, while in searching users gave up after a sequence of empty set returned. Despite the failures, 67% users preferred searching over browsing (20%), with 13% neutral. Participants “feel more in control” when searching but also found the browsing having “unclear categories” or “don’t like to scroll up and down”. Who preferred browsing (5) prized its exhaustiveness (“when searching you never know if you have actually retrieved all”) and transparency (“easier to pull out”). The advantage of the digital format was evident in users’ comments:

“to jump in on a clip and start the video at any point is brilliant”, “you get a very good idea of what’s on the system just by skimming through”.

Despite the initial removal of the corrupted keyframes, participants were frustrated by the high number of images in the visual summary, i.e. a video of 72 min had 367 frames. Many, when scrolling through several screens of images, gave up and looked at other material instead thus missing relevant video shots. This behavior occurred most with the browsing (relevance of the video was uncertain) contrary to the search.

Eight NATCECT users were involved in a focus group. They showed a fairly different behaviour from the evaluation participants: they had no problem with titles and categories as they were familiar with the archive terminology; they searched for highly conceptual content; the keyframes (visual surrogate) were used much more than the text; and much time was spent watching videos and commenting the content.

Comments on the system usefulness were enthusiastic and emphasised having the collection accessible anytime and more flexibly organized. The current configuration requires the video files to be stored on the client PC, and only the keyframes are displayed remotely but this was considered enough to get an idea of the archive content.

3 Implication for Design

Our experience shows a minimalist approach can offer enough functions for broadening access to rare video archives: accessibility is much more valued than visual quality. However some effort from the curator/archivist is required to improve the result of automatic visual indexing as historical and amateur videos are challenging, e.g. the keyframes used to summarize the video must be carefully selected to properly represent the content. A better integration of textual description and visual content is also needed, e.g. more semantic and links would allow to access single clips than to the video as a whole.

Even if searching is the most desirable feature, browsing is essential to support serendipitous discovery of unique videos and the two modalities need to be properly blended for an effective interaction. A browsing interface should offer multiple facets derived from the data and not only from the archive in order to support naïve users in familiarising with the collection. Watching movies is highly entertaining and is likely to occur often, as happened during the focus group: an easy access is crucial.

References

1. Open Video Project [The] <http://www.open-video.org/index.php> (accessed 4.3.2005)
2. Smeaton A.F. et al. The Físchlár-News-Stories System: Personalised Access to an Archive of TV News. RIAO 2004, Avignon, France, 26-28 April 2004.
3. Wactlar H., Christel M., Gong Y., Hauptmann A. Lessons Learned from the Creation and Development of a Terabyte Digital Video Library. *IEEE Computer*, 32(2), (1999) 66-73.

A Flexible Framework for Content-Based Access Management for Federated Digital Libraries

K. Bhoopalam, K. Maly, F. McCown, R. Mukkamala, and M. Zubair

Computer Science Department, Old Dominion University, Norfolk, VA – 23529, USA
{kbhoopal, maly, fmccown, mukka, zubair}@cs.odu.edu

Abstract. Recent advances in digital library technologies are making it possible to build federated discovery services which aggregate metadata from different digital libraries (data providers) and provide a unified search interface to users. In this work we develop a framework that enables data providers to control access to their content in the federation. We have built and tested such a framework based on XACML and Shibboleth.

1 Introduction

One of the primary obstacles that keep data providers from contributing to digital library federations is the lack of an infrastructure to support dynamic access policies. We extend our earlier work [2] to focus on *content-based access restrictions* for digital libraries using the XACML standard [3]. For example, we can restrict any material containing word *nuclear* from being accessed by a specific user group.

Some of the earlier work on content based authorization models [1] adds considerable administrative overhead on pre-existing digital archives. Our approach has no such overhead and allows for the specification of content-based restrictions on the values of meta-data fields. Some systems such as [4] wrap a digital object (e.g., multimedia objects) with authorization information. Although such an association allows for fine-grained access to parts of a digital object, the association remains static.

2 Implementation

We have used XACML's *obligation* feature (element) for specifying content restrictions. Figure 1 shows an obligation element encoded to be fulfilled on a 'permit' decision and is placed in a *Policy* element that encodes rules for a specific role (not shown in the snippet). Multiple phrases are separated by a colon (e.g., nuclear:anthrax). The policy engine returns the entire obligation *as is* to the enforcer (PEP). In the example, the obligation with obligationId "content-restrictions" states the following: "Whenever a user's (e.g., student role) request is permitted, the user may not see records that contain *anthrax or nuclear* in the description. The enforcer translates the obligation received in the response into a SQL statement to ensure that only the required information is fetched from the database.

Figure 2 is a scaled-down version of the policy editor we have developed for our system. The content restrictions are applied to the description and subject metadata

```

<Policy>
...
<Obligations>
  <Obligation ObligationId="content_restrictions"
    FulfillOn="Permit">
    <AttributeAssignment AttributeId="description"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      nuclear:anthrax</AttributeAssignment></Obligation>
  </Obligations>
</Policy>

```

Fig. 1. XACML Code Snippet

fields. The content administrator manages XACML access policies through this simple point-and-click editor. The changes that the content administrator makes are translated into XACML and become effective immediately. Content-based access restrictions did not introduce additional noticeable delay when tested against 50,000 records. We will pursue tests on larger test beds.

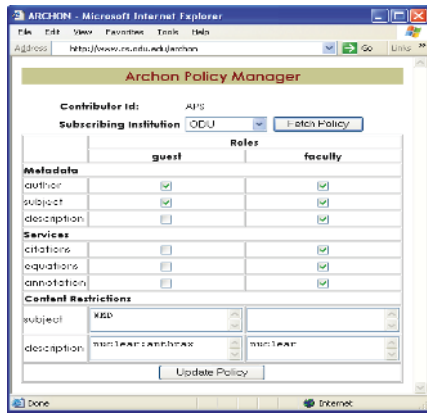


Fig. 2. Access Policy Editor

References

1. Adam N.R., Atluri V., Bertino E., and E. Ferrari. A content-based authorization model for digital libraries. *IEEE Trans. on Knowledge and Data Engineering*, 14(2):296–315, March 2002.
2. Bhoopalam, K., Maly, K., Mukkamala, R., Zubair, M. Access Management in Federated Digital Libraries. *Proceedings of WWW/Internet, IADIS, Madrid*, October 6-9, 2004.
3. Godik, S. and Moses, T. (eds.). OASIS eXtensible Access Control Management Language (XACML). Committee Specification 1.0, <http://www.oasis-open.org/committees/xacml/repository/cs-xacml-core-01.pdf> (21 April 2002).
4. Kodali N., Farkas C., Wijesekera D. An Authorization Model for Multimedia Digital Libraries. *The Int. Journal of Digital Libraries*, Vol 4, 139 -155., 2004

The OAI Data-Provider Registration and Validation Service

Simeon Warner

Cornell Information Science, Ithaca, NY 14850, USA
simeon@cs.cornell.edu

Abstract. I present a summary of recent use of the Open Archives Initiative (OAI) registration and validation services for data-providers. The registration service has seen a steady stream of registrations since its launch in 2002, and there are now over 220 registered repositories. I examine the validation logs to produce a breakdown of reasons why repositories fail validation. This breakdown highlights some common problems and will be used to guide work to improve the validation service.

1 Introduction

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [1] was released in 2001. Alongside the OAI-PMH, the validation service was launched to allow data-providers to check compliance with the protocol, and has helped identify errors in popular software packages (e.g. DSpace and eprints.org) and in particular OAI-PMH deployments. I summarize an analysis of registration and validation requests received during 2004, and discuss these results in the context of new work to improve the validation facilities (see [2] for details).

2 Registration and Validation

The registration and validation services differ only in that, if validation is successful, registration requests result in the `baseURL` being entered in the central registry. Henceforth I shall consider all requests together and refer to them as ‘validation requests’. During 2004 there were 1797 validation requests from which a valid OAI-PMH `baseURL` could be extracted. A number of error conditions cause the validator to abort, including fundamental errors such as the wrong protocol version being reported, and errors where it is not possible to extract data required for subsequent tests. In 40% of aborted validations there was no response to the `Identify` request, usually because the `baseURL` was entered incorrectly. In 21% of cases, bad XML was returned resulting in failure to parse the `Identify` response. As the response to the `Identify` request is particularly important, several other checks are made on this response. Three other errors causing the validation to abort are more indicative of problems with the repository implementation. 1% of the aborted validations were because of more than 5 successive HTTP 503 `Retry-After` responses. Two final reasons relate to failure to extract the timestamp of a sample record, without which timestamp-based

incremental harvesting cannot be tested. In 3.3% of cases there were no items in the repository, and in a further 2.5% of cases, no datestamp could be extracted from the sample record.

There were 927 completed validation requests, of which 318 were successful, 198 had errors only in the handling of exception conditions and 411 had other errors. Failures occurred in all conditions tested although certain failures were particularly common. The 5 most common errors are shown in table 1.

Table 1. Common errors in cases where validation was completed

Error	Number
Schema validation errors in standard verb responses	168
Empty response when <code>from</code> and <code>until</code> set to known <code>datestamp</code>	57
Empty <code>resumptionToken</code> in response to request without <code>resumptionToken</code>	42
Malformed response to request with identifier <code>invalid" id</code>	40
Granularity of <code>earliestDatestamp</code> doesn't match <code>granularity</code> value	35

There were 152 repositories that successfully passed robust validation in 2004. For 38% of these, often deployments of standard software, validation was successful on the first attempt. A further 30% achieved validation excluding exception conditions before robust validation, and it took about 3 further attempts on average to correct responses to exception conditions. 33 repositories managed validation excluding exceptions but never passed robust validation and were thus not eligible to register.

3 Discussion and Future Work

The continued use of the validation facility and the registration of new repositories attests to the value of these services. It is reassuring to see that most repositories managed to correct errors and pass validation in just a few attempts. However, personal assistance was provided in a number of cases and some sites tried to validate several times but never succeeded, suggesting room for improvement in the protocol documentation and the helpfulness of the validation suite.

This analysis is the first step in a project to produce improved OAI validation tools for the NSDL and the broader OAI community. Future work will include refinement of the existing validation suite, and development of validation and testing software for harvesters.

References

1. Lagoze, C., de Sompel, H.V., Nelson, M., Warner, S.: The Open Archives Initiative Protocol for Metadata Harvesting, version 2.0 (2002) <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>
2. Warner, S.: The OAI Data-Provider Registration and Validation Service (2005) <http://arxiv.org/abs/cs.DL/0506010>

An Effective Access Mechanism to Digital Interview Archives

Atsuhiro Takasu and Kenro Aihara

National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
{takasu, kenro.aihara}@nii.ac.jp

1 Introduction

Skill and knowledge of master workmen and artists are important information for digital libraries. Usually disciples acquire the skill and knowledge by conversation with masters and watching the master's works. Therefor they can be conveyed to limited number of disciples and they are sometimes lost when masters and artists pass away. A digital library for skill and knowledge plays an important role to preserve and convey them to large number of people. Since skill and knowledge inherent in masters and artists, first we need to externalize and represent them in an appropriate form.

Interview to masters and artists are effective way to record their skill and knowledge. It can record various kinds of information such as emotional behavior, procedure of creative activity as well as verbal information in conversation. Furthermore interview enables us to obtain the information from masters and artists without heavy mental load. This characteristics is effective not only for gathering information of the skill and knowledge of masters and artists but also for externalizing knowledge of human beings in many fields.

2 System Overview

We have been developing a system named MONO to archive skill and knowledge of artists [1]. In this project we adopted interview videos as the main medium of information. This system consists of data collection and three modules as shown in figure 1.

The system provides two kinds of data: interview videos and a supplementary collection of texts. Interview videos are the main part of the collection. In the interview, masters and artists explain their creative works using their artifacts. Although an interview is a very effective way to record the skill and knowledge, it is sometimes too specific for users who are not familiar with the field. To supplement interview, the system provides a collection of texts such as dictionary, articles, books and catalogs of exhibitions.

The system has three functional modules: a video editing support module (VESM), a text information capturing module (TICM) and a user interface module (UIM). VESM and TICM provide information archiving functions and

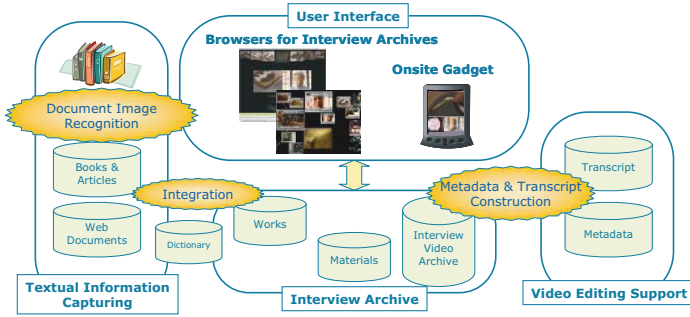


Fig. 1. An Overview of MONO project

UIM provides an information utilization function. VESM helps editors to segment videos into scenes and to make transcript of interview. On the other hand, TICM supports to obtain text information from related books and articles using document image analysis technology. UIM is a user interface for users to access the interview video archive.

3 Speech Processing for Effective Access to Interview Archives

Speech processing can be used in various phases in interview video archiving system. Its typical application is interview retrieval based on the occurrence of a keyword in the interview or a similarity between the conversation and query words in document information retrieval. It can be also used to associate supplementary text information to the scenes of interviews.

In these applications, we first need to detect the occurrence of words in interviews. This problem is called spoken document retrieval (e.g., [2]). When applying spoken document retrieval, we need to define a similarity between spoken signal and words in interviews. In case of interview videos, they contain the conversation between many interviewers and interviewees. Therefore, the similarity should be tunable to each person.

We proposed a statistical model for defining a similarity function called Dual and Variable Hidden Markov Model (DVHMM) [3]. It is an extension of edit distance which is a basic function for measuring string similarity. DVHMM has the following features:

- It can represent various editing operations including insertion, deletion, and replacement used by the edit distance,
- It can assign weight to each editing operation,
- The weights can be estimated from training data.

Because of these features, DVHMM can help the construction and retrieval of interview archive efficiently.

References

1. K. Aihara and A. Takasu. “Reciprocal Platform for Archiving Interview Video about Arts and Crafts”. In *Joint Conference on Digital Libraries*, page to appear, 2005.
2. S. Srinivasan and D. Petkovic. “Phonetic confusion matrix based spoken document retrieval”. In *23rd ACM SIGIR*, pages 81–87, 2000.
3. A. Takasu and K. Aihara. “DVHMM: Variable Length Text Recognition Error Model”. In *submit to 15th International Conference on Pattern Recognition*, pages Vol.III, 110–114, 2002.

A Semantic Structure for Digital Theses Collection Based on Domain Annotations

Rocío Abascal, Béatrice Rumpler, Suela Berisha-Bohé, and Jean Marie Pinon

INSA of Lyon – LIRIS,
7 Avenue J. Capelle Bâtiment Blaise Pascal,
F69621 Villeurbanne cedex, France
{Rocio.Abascal, Beatrice.Rumpler, Suela.Bohe}@insa-lyon.fr
<http://liris.cnrs.fr>

Abstract. Search performance can be greatly improved by describing data using Natural Language Processing (NLP) tools to create new metadata for digital libraries. In this paper, a methodology is presented to use a specific domain knowledge to improve user request. This domain knowledge is based on concepts, extracted from the document itself, used as “*semantic metadata tags*” in order to annotate XML documents. We present the process followed to define and to add new XML semantic metadata into the digital library of scientific theses. Using these new metadata, an ontology is also built to complete the annotation process. Effective retrieval information is obtained by using an intelligent system based on our XML semantic metadata and a domain ontology.

1 Introduction

Internet has developed digital libraries that make available a great amount of digital information. Nowadays, users must evaluate the pertinence of documents presented by the web. Generally, to evaluate the pertinence, users read several fragments of the documents rather than the complete documents.

The project of INSA of Lyon called CITHER (consultation of digital entire text versions of theses) concerns the online publishing of scientific theses. We encountered the same difficulties to find pertinent information in the CITHER system as in other digital libraries. During a search session, it is impossible to extract the pertinent contents of several theses. To evaluate the pertinence of a thesis, users must read the entire document. Furthermore, a document is often too long for a quick evaluation.

A promising way to solve this problem is to use metadata to “*annotate*” the documents and to describe their content in a better way. In our proposal, we have decided to extract the concepts that best describe the theses and to use them as metadata like “*semantic annotations*”. Of course, manual extraction of concepts is a long time-consuming, so to overcome these limitations we use a Natural Processing Language (NLP) tool to automate the extraction of concepts

2 Methodology Used to Annotate Digital Theses

In this paper we present an approach to improve document retrieval by using NLP tools based on the semantic content of digital theses. Our approach has a double ad-

vantage: first, it can entirely exploit the content of digital theses by using semantic annotations and second, it can provide new alternatives to the users' requests. To be able to find pertinent information using information retrieval tools, it is important to define a specific structure of the digital document during its creation. According to this point of view, we have defined a semantic structure of the document by integrating new metadata in significant parts of the corpus. This makes possible to identify semantic segments in the scientific theses stored in our digital library: CITHER. In a search session based on keywords, the system will compare them with the semantic metadata (delimiting the semantic segments) and with the keywords describing the thesis.

To use the concepts extracted by the index of the tool named Nomino [2], we have proposed a tool to “*annotate*” documents [1]. The task consists in adding new metadata into the thesis during the PhD student writing session. The student adds the new metadata by using (1) the base of concepts, (2) the Nomino evaluation and (3) his personal tags. So, after the student has inserted the metadata into the thesis, the tool allows the identification of the semantic markups. When the paragraph containing the inserted symbol (which contains the concept) is identified, it becomes embedded by a simple tag such as “*<concept-name>*” and “*</concept-name>*” at the end. This annotation scheme [3] allows us the management of Nomino concept’s as well as the indexation and extraction of pertinent paragraphs from the document according to specific search criteria. During a search session, the system focuses on semantic markups, the XML tags, in order to retrieve the pertinent paragraph(s).

3 Conclusion and Further Work

Thanks to this approach the user can get pertinent fragments from one or several theses corresponding to the semantic segment. We are currently working on the design of an “*advanced*” system based on this ontology to find more pertinent information.

Further research should investigate the model of the users profile to personalize the search sessions.

References

1. Abascal R., Rumpler B., Evaluación de herramientas de extracción automática de conceptos dentro de un ambiente de biblioteca digital. In Colombian Journal of Computation, Vol. 6 No. 1, ISSN 1657–2831, June 2005.
2. Plante P., Dumas L., and Plante A., Nomino version 4.2.22.
3. Thomasson J-J., Schémas XML, Ed. Eyrolles, ISBN: 2-212-11195-9, 2002, 466 p.

Towards Evaluating the Impact of Ontologies on the Quality of a Digital Library Alerting System

Alfons Huhn, Peter Höfner, and Werner Kießling

Institut für Informatik, Universität Augsburg,
86159 Augsburg, Germany
{huhn, hoefner, kiessling}@informatik.uni-augsburg.de

Abstract. Advanced personalization techniques are required to cope with novel challenges posed by attribute-rich digital libraries. At the heart of our deeply personalized alerting system is one extensible preference model that serves all purposes in conjunction with our search technology Preference XPath and XML-based semantic annotations of digital library objects. In this paper we focus on the impact of automatic query expansion by ontologies. First results indicate that use of ontologies improves the quality of the result set and generates further results of higher quality.

1 Introduction

P-News [4] is an experimental system that alerts users, when user-relevant documents newly arrive at the digital library. Its main feature consists in a deep personalization which is achieved by a highly flexible preference methodology [3] with powerful query capabilities. This methodology enables a consistent description framework for the users' wishes, the defaults assigned to user groups, and the domains of discourse modeled by ontologies. We evaluate the impact of query expansion by ontologies on the quality of the result set by analyzing the set of best matching objects (BMO-set) [3], when ontologies are used or are ignored. The full version of this poster is available at [1].

2 Impact of Ontologies on the Quality of the Result

In ontologies any agent easily finds *synonyms*, *hyponyms*, *hypernyms* as well as other complex semantic contexts. Depending on the involved attributes of the query, P-News decides which ontologies have to be used for the query expansion. This attribute-aware procedure sharpens the focus by just applying those ontologies which semantically model the domain of the attribute. Not surprisingly an increase of quality arises only in those test cases, where we did not have perfect matches and where the user's terms are also be found in the involved ontology. As stated in [2] recall for the ontology-based query expansion tops recall for keyword-based techniques. To reduce loss of focus, we exploit the semantic context for sense disambiguation.

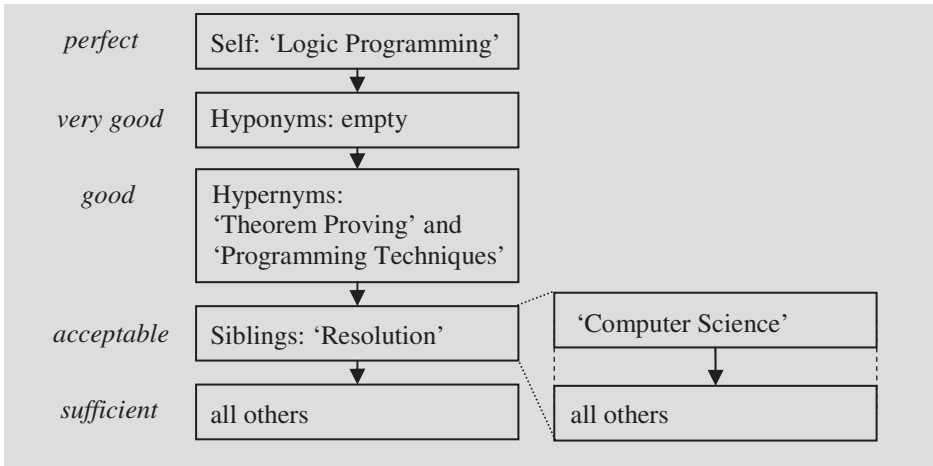


Fig. 1. Quality refinement reducing a loss of focus

Let's look at Fig.1: Suppose that a user's query for 'Logic programming' (*self*) achieves no perfect hit and no hyponyms exist. Assuming that a *sibling* (e.g. 'Resolution') has multiple hypernyms (e.g. 'Programming Techniques', 'Screen'), loss of focus may arise. To cope with this challenge, we exploit the context. We identify the common hypernyms of self and this sibling. All the hypernyms of the common hypernyms (e.g. 'Computer Science') are used as the POS-set of a POS preference [3] applied to the BMO-set of the sibling, i.e., they are preferred in comparison with all others (e.g. especially 'Screen'). The common hypernyms (e.g. 'Theorem Proving') must not be considered, because they already have a better quality (see Fig. 1).

With our case studies performed so far we could find strong evidence for the following conjectures: Use of ontologies can improve the quality of the BMO query result, and it can be controlled to reduce a loss of focus during query expansion.

Acknowledgement. P-News is funded within the German Research Foundation's strategic research initiative 'Distributed Processing and Delivery of Digital Documents (V^3D^2)'.

References

1. A. Huhn, P. Höfner, W. Kießling: Towards Evaluating the Impact of Ontologies on the Quality of a Digital Library Alerting System. Technical Report 2005-07, Institute of Computer Science, University of Augsburg
2. L. Khan, D. McLeod, E. Hovy: Retrieval effectiveness of an ontology-based model for information selection. In *The VDLB Journal* (2004) 13-1, 71-85, Springer New York
3. W. Kießling: Foundations of Preferences in Database Systems. In *Proc. Int. Conf. on Very Large Databases (VLDB 2002)*, Hong Kong, China, 311-322
4. Q. Wang, W.-T. Balke, W. Kießling, A. Huhn: P-News: Deeply Personalized News Dissemination for MPEG-7 based Digital Libraries. In R. Heevy, L. Lyon (eds.): *Research and Advanced Technology for Digital Libraries (ECDL2004)*, Bath, UK, 256-268

Building Semantic Digital Libraries: Automated Ontology Linking by Associative Naïve Bayes Classifier

Hyunki Kim¹, Myung-Gil Jang¹, and Su-Shing Chen²

¹ Computer and Information Science and Engineering Department,
Electronics and Telecommunications Research Institute,
Taejon 305-700, Republic of Korea
{hkk, mgjang}@etri.re.kr

² Computer and Information Science and Engineering Department,
University of Florida, Gainesville, Florida 32611, USA
suchen@cise.ufl.edu

Abstract. In this paper, we present a new classification method, called *Associative Naïve Bayes (ANB)*, to associate MEDLINE citations with Gene Ontology (GO) terms. We define the concept of class-support to find frequent itemsets and the concept of class-all-confidence to find interesting itemsets. Empirical test results on three MEDLINE datasets show that ANB is superior to naïve Bayesian classifier. The results also show that ANB outperforms the state of the art Large Bayes classifier.

1 Introduction

Our goal is to build a semantic digital library using the Open Archives Initiative Protocol for Metadata Harvesting [1] and data mining techniques. We define the semantic digital libraries as the digital libraries that can discover hidden, useful information from large amounts of data stored in digital libraries using data mining techniques. In this paper, we concentrate on the development of a new classification method, called *Associative Naïve Bayes (ANB)*, that will be used as a part of data mining component in our semantic digital libraries and our task is to automatically classify MEDLINE citations with Gene Ontology (GO) for building a biomedical semantic digital library.

2 Associative Naïve Bayes Classifier

To employ a frequent pattern mining method for classification, each itemset should have an associated statistical significance for each class label c_i . For this, we formally define the concept of *class-support* of an itemset [3], how many times an itemset occurs given a class c_i in database D . The *class-support* of an itemset X , $class_support(X, c_i)$, is the ratio of the number of transactions that contain an itemset X and are labeled with class c_i in D and the total number of transactions in D . We also define *class-all-confidence* to find interesting itemsets from frequent itemsets. The *class-all-confidence* of an itemset X , $class_all_conf(X, c_i)$, is the ratio of the number of transactions containing an itemset X and are labeled with class c_i in D and the maximum cardinality of transactions containing the power set of X .

Associative Naïve Bayes classifier consists of two parts: a frequent and interesting itemset generator (ANB-IG) and a classifier (ANB-CL). The main task of ANB-IG is to generate the frequent and interesting itemsets that have both *class-support* above *min_class_support* and *class-all-confidence* above *min_class_all_conf*. After the frequent and interesting itemsets are found, new unlabeled documents are classified by the algorithm ANB-CL. A feature vector of a new document $d = \langle l_1, l_2, \dots, l_n \rangle$ is used to generate a set of n discrete variables $\{l | l \in F \wedge l \in P(d)\}$ and the generated set is used to generate a product approximation.

3 Empirical Results

We constructed three kinds of datasets (small, medium, and large datasets) from MEDLINE database. We obtained 12,585 citations, 111,871 citations, and 228,876 citations for small, medium, and large datasets, respectively. We then randomly assigned a 70% of data to the training set and the remaining 30% of data to the test set for each dataset. Each dataset contains 10 categories. For evaluating the performance of ANB, we use the standard recall (r), precision (p), and F_1 measure. Classification accuracy results for vocabulary sizes of 100, 500, and 1000 words, are depicted in Table 1. In most cases ANB is superior to naïve Bayes [2] and large Bayes [3].

Table 1. Classification results of NB, LB, and ANB algorithms

Dataset		#Words								
		100			500			1000		
		p	r	F_1	p	r	F_1	p	r	F_1
Small	NB	91.57	91.87	81.72	94.90	95.02	94.96	96.11	96.23	96.17
	LB	93.53	94.07	93.80	96.24	96.38	96.32	96.24	96.36	96.30
	ANB	93.80	94.17	93.99	96.65	96.74	96.70	96.91	97.03	96.97
Medium	NB	80.04	80.98	80.50	85.54	86.39	85.96	87.54	88.44	87.99
	LB	83.06	84.55	83.80	87.30	88.37	87.83	88.35	89.38	88.86
	ANB	83.25	84.59	83.92	87.70	88.64	88.16	88.87	89.82	89.35
Large	NB	71.81	74.45	73.11	77.43	79.08	78.24	80.37	81.43	80.89
	LB	73.81	79.16	76.39	79.43	86.84	82.97	81.03	86.70	83.77
	ANB	73.70	78.94	76.23	79.67	86.86	83.11	81.26	86.75	83.91

References

1. Kim, H., Choo, C., and Chen, S.: An Integrated Digital Library Server with OAI and Self-Organizing Capabilities. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003)*, Norway (2003) 164-175
2. Mitchell, T.M.: *Machine Learning*. McGraw-Hill (1997)
3. Meretakis, D. and Wuthrich, B.: Extending Naive Bayes Classifiers Using Long Itemsets. In *Proceedings of the fifth ACM SIGKDD*, San Diego, CA, USA (1999) 165-174

Evaluation of a Collaborative Querying System

Lin Fu, Dion Hoe-Lian Goh, and Schubert Shou-Boon Foo

Division of Information Studies,
School of Communication and Information,
Nanyang Technological University,
Singapore 637718
{p148934363, ashlgoh, assfoo}@ntu.edu.sg

Abstract. We report evaluation results for a collaborative querying environment. Our results show that compared with traditional information retrieval systems, collaborative querying can lead to faster information seeking when users perform unspecified tasks.

1 Introduction

Collaborative querying helps users formulate queries by sharing expert knowledge or other users' search experiences. In previous work, a collaborative query environment (CQE) was developed for a digital library as shown in Figure 1. The system operates by clustering and recommending related queries to users using a hybrid query similarity identification approach. Users can explore the query clusters using a graph-based visualization system, the Query Graph Visualizer (QGV) [2]. The purpose of this paper is to evaluate the CQE with goal of informing the usefulness and usability of such a system.

2 Evaluation Design and Results

Sixteen students from Nanyang Technological University participated in this evaluation. We created two categories of tasks: clearly specified tasks and unclearly specified tasks [1]. Each category contained two tasks. The 16 participants were randomly divided into four groups of four participants each. Groups A and B used the CQE to complete the clearly specified and the unclearly specified tasks respectively. Participants in Groups C and D used NTU's existing OPAC system to complete the clearly specified and the unclearly specified tasks respectively. The time taken to accomplish the tasks successfully was recorded and used to measure the usefulness of the CQE. Finally, all students will answer a questionnaire regarding the design features of the CQE. The time for Group A, B, C and D to finish the task is 3, 12.5, 3 and 22 minutes respectively. Our results show that CQE can speed up the unspecified tasks compared with OPAC system. The reason is that most users can not formulate a precise query to represent their information need for the unspecified tasks in the first round of search. This leads the participants in group D to spend lots of time sifting through the result listings and reformulating the queries. However, for the participants

in group B, they can formulate the proper queries by either harnessing the recommended queries or exploring the QGV which in turn reduces the time sifting through the result listings. On the other hand, there is no noticeable difference between Groups A and C in the time required to complete the clearly specified tasks accurately. This indicates that the collaborative querying has no time advantage in the process of information seeking for clearly specified tasks.

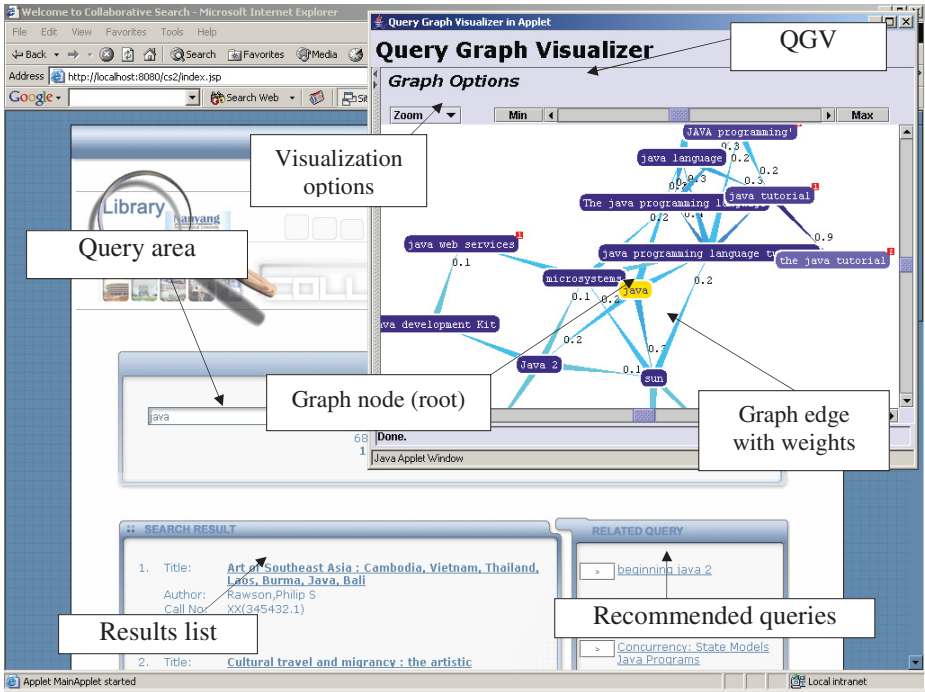


Fig. 1. The Collaborative Querying Environment

References

- [1] C. Plaisant, B. Shneiderman, K. Doan, T. Bruns. (1999) Interface and data architecture for query preview in networked information systems. *ACM Transactions on Information Systems*, 17(3), 320-341.
- [2] L. Fu, D. Goh, S. Foo & Y. Supangat (2004) Collaborative querying for enhanced information retrieval. *Proceedings of the 8th European Conference on Digital Libraries*, 378-388.

Aiding Comprehension in Electronic Books Using Contextual Information

Yixing Sun, David J. Harper, and Stuart N.K. Watt

School of Computing, The Robert Gordon University,
St Andrew Street, Aberdeen AB25 1HG, UK
miki.sun@smartweb.rgu.ac.uk, [d.harper, s.n.k.watt]@rgu.ac.uk

1 Motivation

A person reading a book needs to gain insights based on the text. In most books, stories, themes, and references are organized structurally and purposefully. In previous work, we presented the design of an e-Book user interface that reveals the multi-structural information to support reading for comprehension[1]. In this paper, we describe techniques for discovering and representing the narrative structure of e-Books, and describe the user interface components for revealing this narrative structure to readers. We chose e-Bible as our corpus and named our user interface "iSee", meaning that "I see what I read".

2 Design of the iSee User Interface

The design of the iSee interface was motivated by scenarios of use, devised for both novice and experienced readers of the Bible¹. One scenario of use involved a new reader of the Bible as follows:

A new reader is studying the story of Jacob's marriage to two sisters, Leah and Rachel (Genesis, Chapters 29 and 30). She wishes to explore similar stories, where similarity is based both on the characters that feature in the story and on the theme(s) of the story (e.g. marriage, family, etc). As a new reader, she requires support in comprehending the organizational structure of the Bible, and in exploring the narrative structure, i.e. similar or relevant stories.

Figure 1 shows a screenshot of the user interface of iSee. Part 1 is an interactive e-Bible viewer with the visualization of the organizational structure. An overview of the structure of the Bible shows parts of the Bible, the books, and relative lengths of each book. The narrative thread structure is presented in an "overview + details" interactive visualization. Each multi-verse narrative segment within a chapter is identified and displayed in the text viewing area in Part 1. An overview of all the relevant narrative

¹ The Bible is divided into two parts, the Old Testament and the New Testament. There are 66 books in the Bible, each book is divided into a number of chapters, and each chapter is divided into a number of verses.

segments is displayed separately in Part 2, according to their similarity to a selected segment. Together, the selected segment and related segments comprise a narrative thread. The reader can indicate her interest in a given story within a chapter displayed in Part 1, and explore the similar or relevant stories displayed in Part 2.

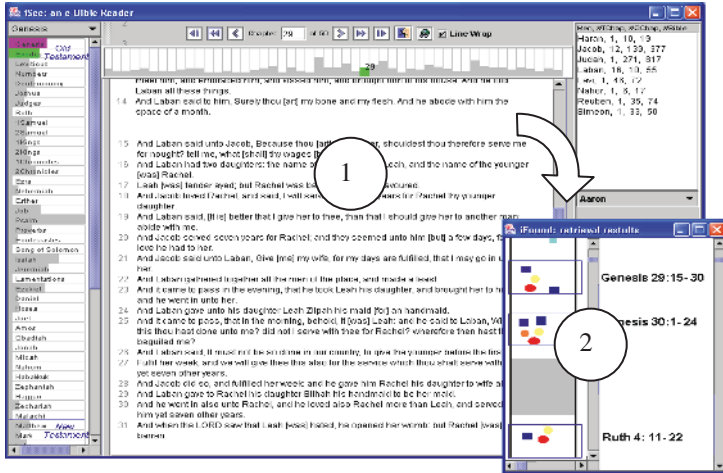


Fig. 1. iSee e-Bible Screenshot

3 Narrative Segment Detection and Linking

Two techniques underpin the user interface, namely narrative segment detection and narrative thread linking. First, we need to identify the coherent story segments within a chapter. Second, we need to link (or retrieve) similar stories given a particular story.

For segment detection, we adapt Hearst's multi-paragraph segmentation technique[2]. In that work, gaps/breaks between text segments are identified by computing the similarity between adjoining 5-sentence blocks across the text to be segmented. Our approach differs in that we model our blocks (5 verses each) using statistical language modelling, and specifically a simple "bag of words" model. Critically, we are able to give additional weight to the occurrence of biblical characters (e.g. Jacob), where these characters are identified by reference to a gazette of biblical names. The consecutive block models are compared using Jensen difference measure, a symmetric divergence measure[3] that measures the extent to which blocks differ. Initial experiments have shown that this technique is effective in partitioning a chapter into non-overlapping stories (narrative segments).

Given the individual narrative segments, we can then retrieve similar segments for any given segment (or story). As before, we model the individual narrative segments using a "bag of words" model, and retrieve similar segments by computing the divergence between a target story and all stories identified by the segmentation process. Thus, given a target story within a chapter, we can rank similar stories, and support linking between stories, and hence support the discovery of narrative threads.

Our approach opens the possibility of differentially weighting characters against non-character words, and so emphasising characters and/or thematic elements of a story. Currently, we are also investigating ways of visualising information showing what aspects of a story contributed most to its retrieval.

4 Conclusion and Future Work

Readers need support to understand the organizational and narrative thread structure of a book for better comprehension; this support could be achieved using information retrieval and visualization techniques. We have carried out initial experiments for the narrative threads detection algorithms, and will report the results shortly. Meanwhile, we plan to conduct user experiments for the e-Bible user interface in the near future.

References

1. Sun, Y., D.J. Harper, and S.N.K. Watt. *Design of an e-Book User Interface and Visualizations to Support Reading for Comprehension*. In *Proceedings of ACM SIGIR*. 2004. Sheffield, UK: ACM Press. p. 510-511.
2. Hearst, M.A., *TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages*. *Computational Linguistics*, 1997. **23**(1): p. 33-64.
3. Taneja, I.J. *Generalized Information Measures and Their Applications*. Electronic resource, <http://www.mtm.ufsc.br/~taneja/book/book.html>, Accessed on 3rd May, 2005

An Information Foraging Tool

Cathal Hoare and Humphrey Sorensen

Computer Science Department, University College Cork, Ireland
{c.hoare, sorensen}@cs.ucc.ie

Abstract. Electronic document repositories continue to expand rapidly; public collections, for instance the Google index, contain up to 8 billion individual items. Private electronic archives, maintained by companies, governments and other bodies grow at similar rates. While search techniques have scaled to manage these vast collections, most interfaces between search engines and searchers, usually based on a ranked list, are increasingly insufficient. This paper explains how Information Foraging Theory was applied to create visualisations of query resultsets which, when embedded in an application that contained tools to manipulate the visualisation, helped alleviate the deficiencies of the ranked list.

1 Foraging for Data

Most users of search engines do not use advanced features when formulating a query. Jensen [1] and other have shown that 85% of users submit simple queries consisting of just three phrases or less. Often these queries return large resultsets. Users examine relevant items from the resultset, refining their knowledge before submitting an improved query. This cycle is repeated until a manageable set of relevant results is obtained. Pirolli and Card [2] called this search/browse cycle *information foraging*; they created a comparison between the natural world and the infoscape where relevant information was compared to sources of nutrition and information seekers took the role of hunters. The analogy was developed into a gain function that described the rate at which relevant documents were retrieved by a given foraging technique. The proponents of the theory have shown, through experimentation, that the rate at which searchers find relevant documents grows if the documents are gathered together into patches or clusters, where similar documents are co-located. The authors believed that combining Information Foraging Theory and visualisation principles, such as Rao's *focus + context* concept [3] would produce a superior interface for examining query resultsets.

The ranked list does not scale well to large datasets; it is, however, simple to implement, inexpensive and intuitive to use. While many interesting visualisations have been produced, quite often, these efforts have introduced complicated interactions that have not achieved the same universality as the ranked list. The authors believe that it is important to retain the simplicity of the ranked list while leveraging the information map's properties to decrease searchers' cognitive workload.

The application retrieves a resultset, the result of a query, from which the application calculates and presents an interactive 2-dimensional proximity based visualisation to the searcher. The position of each artefact is calculated by applying a clustering algorithm [4] to inter-artefact similarity measurements and then submitting those clusters to a force-directed layout algorithm [5] to determine the position of nodes within the visualisation. Encoding information about each artefact in its representation further enhances the functionality provided by the visualisation; this includes whether or not a node has been visited. Ranking of documents is accomplished through scaling the intensity of the nodes' colour. A ranked list is also provided to create a link with the traditional approach of representing results. The visualisation is enhanced by the provision of several tools. The threshold controls manipulation of the visualisation allowing a searcher to vary the density of information patches. The proximity tool allows users to focus on particular artefacts within segments of the visualisation by changing affected nodes' colour. This allows users to maintain focus on particular parts of a visualisation without losing the context of those nodes within the overall resultset.

Evaluation of the application included both usability testing and performance testing; performance testing will remain outside of the scope of this paper. Indicative tests, involving small groups, have been conducted to gain feedback that was used to refine aspects of the application. This testing was based on users completing several standard tasks before answering a questionnaire. Users expressed satisfaction with the application; their satisfaction grew as they became familiar with the application. The authors feel that questionnaire based tests can yield subjective data. To improve the quality of the usability tests, eye-tracking tests will be employed with a larger subject group to create unbiased empirical results of how users interact with the application.

References

1. Jasen, B. J., Spink, A., Bateman, J., Saracevic, T.: Real Life Information Retrieval: A Study of User Queries on the Web. In: ACM SIGIR Forum, 32(1) (1998) 5-17
2. Pirolli, P., Card, S.: Information Foraging. In: Psychological Review, 106(4) (1999) 643-675
3. Lamping, J., Rao, R., Pirolli, P.: A Focus+Context Technique Based on Hyperbolic Geometry for Visualising Large Hierarchies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver, Colorado (1995) 401-408
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. In: ACM Computer Surveys, 31(3) (1999) 264-323
5. Fruchterman, T., Rheingold, E.: Graph Drawing by Force-directed Placement. In: Software - Practice and Experience, 21(11) (1991) 1129-1164

mod_oai: An Apache Module for Metadata Harvesting

Michael L. Nelson¹, Herbert Van de Sompel², Xiaoming Liu²,
Terry L. Harrison¹, and Nathan McFarland²

¹ Old Dominion University, Department of Computer Science, Norfolk VA 23508 USA
{mln, tharriso}@cs.odu.edu

² Los Alamos National Laboratory, Research Library, Los Alamos NM 87545 USA
{herbertv, liu_x, nmcfarl}@lanl.gov

Abstract. We describe mod_oai, an Apache 2.0 module that implements the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). OAI-PMH is the de facto standard for metadata exchange in digital libraries and allows repositories to expose their contents in a structured, application-neutral format with semantics optimized for accurate incremental harvesting. mod_oai differs from other OAI-PMH implementations in that it optimizes harvesting web content by building OAI-PMH capability into the Apache server.

1 Introduction

There has been considerable attention given to increasing the efficiency of web crawlers through more accurate estimation of updates [1]. This problem arises from the fact that http does not support semantics of the form "what resources have changed since 2004-12-27?" Although syndication formats such as RSS are widely implemented, these formats are either in the process of standardization or optimized for syndicating web ephemera and not for accurate incremental harvesting. Within the digital library community, the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is the de facto standard for metadata interchange.

We have developed an Apache module, mod_oai, that automatically responds to OAI-PMH requests on behalf of a web server. If Apache and mod_oai are installed at <http://www.foo.edu/>, then the baseURL for OAI-PMH requests is http://www.foo.edu/mod_oai. While respecting the http access controls specified in `httpd.conf`, mod_oai provides 3 metadata formats in the OAI-PMH responses. Dublin Core is provided, but only technical metadata such as file size and MIME type is included. We introduce a new metadata format, `http_header`, which contains all the http response headers that would have been returned if the resource had been obtained directly. The third metadata format, `oai_didl`, encodes the web resource with the MPEG-21 Digital Item Declaration Language (DIDL) [2]. This representation includes the metadata in the `http_header` format, as well as the web resource itself, either base64 encoded ("by-value"), as a URL ("by-reference"), or both. The `http_header` metadata, either by itself or included in the `oai_didl` metadata format, provides complete http header information about the resource as well; information that is otherwise not available in a standard OAI-PMH usage scenario. The introduction of the `oai_didl` metadata format allows for the incremental harvesting of resources while remaining within the boundaries of the OAI-PMH [3].

A number of subtle interpretations of the OAI-PMH data model are made to achieve optimal functionality of `mod_oai`. First, the URL of the resource serves as the OAI-PMH identifier. Second, the last modified date of the resource is used as the OAI-PMH timestamp of all 3 metadata formats. As a result, all metadata for a given OAI-PMH identifier will share an OAI-PMH timestamp. Lastly, the set membership of item is based on the MIME type of resource.

There are two general classes of `mod_oai` use. The first is to issue only `ListIdentifiers` as a way of identifying new URLs to be added to a regular web crawler. In the `ListIdentifiers` scenario, `mod_oai` offers incremental harvesting semantics with timestamp and sets (i.e. MIME types) as arguments. The second scenario, is to issue `ListRecords`, which causes an entire website to be transformed into OAI-PMH Archival Information Packages (AIPs) and stored for later reconstitution.

2 Conclusions

`mod_oai` currently works for static files only; we are adding support for dynamic pages in a future release. `mod_oai` is not intended to replace existing OAI-PMH repositories, but rather to bring the OAI-PMH semantics of incremental harvesting based on timestamps and sets to general web servers. A full architectural discussion and performance evaluation of `mod_oai` can be found in [4] and more information can be found at <http://www.modoi.org/>.

Acknowledgements

`mod_oai` is supported by the Andrew Mellon Foundation.

References

- [1] Cho, J., Garcia-Molina, H. Estimating Frequency of Change. *ACM Transactions on Internet Technology*, 3, 3, 2003, 256-290.
- [2] Bekaert, J. Hochstenbach, P., Van de Sompel, H. Using MPEG-21 DIDL to Represent Complex Digital Objects in the Los Alamos National Laboratory Digital Library. *D-Lib Magazine*, 9, 11, 2003.
- [3] Van de Sompel, H., Nelson, M. L., Lagoze, C. L., Warner, S. Resource Harvesting within the OAI-PMH Framework. *D-Lib Magazine*, 10, 12, 2004.
- [4] Nelson, M. L., Van de Sompel, H., Liu, X., Harrison, T. L., McFarland, N. `mod_oai`: An Apache Module for Metadata Harvesting, arXiv Technical Report cs.DL/0503069, 2005.

Using a Path-Based Thesaurus Model to Enhance a Domain-Specific Digital Library

Mathew J. Weaver, Lois Delcambre, Timothy Tolle, and Marianne Lykke Nielsen

mweaver@cse.ogi.edu, lmd@cs.pdx.edu,
timtolle@starband.net, mln@db.dk

1 Introduction

Our research¹ focuses on providing easy access to interdisciplinary information in the natural resource management domain [3] so users can more readily benefit from previous scientific findings, assessments, and decisions. Because of the widespread use of specialized terminology, our work focuses on extending a traditional thesaurus model [1] to properly represent and exploit the broad range of terms in a digital library designed for natural resource management.

Natural resource managers gather information necessary to make decisions about the environment from a wide spectrum of documents generated by various individuals for various purposes. Many of these documents focus on numerous topics about a particular location; other documents focus on a specific topic or issue – such as a wildlife survey for a particular location. The terminology of interest in this application domain spans a number of subject areas, with each one including one or more controlled vocabularies. Most of the vocabularies come from existing sources – published glossaries, terminologies, and taxonomies. As part of our work, our team researched and evaluated existing sources to determine their suitability for use in Metadata++, our digital library software system.

2 Path-Based Metadata++ Model

Instead of requiring that all users from all disciplines agree upon a single vocabulary or conceptual model – as in some ontology-based systems [2,6] – Metadata++ lets users from separate discourses use vocabularies “as-is”. Like a typical thesaurus, Metadata++ supports polyhierarchies in the form of *multiple occurrences* – where the same term appears in two or more places in the hierarchy. Figure 1 illustrates a few of the many multiple occurrences of the term *Riparian*. Multiple occurrences often describe the same general concept with slightly different connotations. For example, Figure 1 illustrates the term *Riparian* as a child term for both *Watershed Management* (referring to how the riparian area affects the encompassing watershed) and *Wetlands* (referring to a particular type of wetland that is not permanently inundated but is close to surface water).

¹ This work is supported in part by the National Science Foundation, grant number EIA 9983518. Any opinions, findings, conclusions, or recommendations expressed here are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

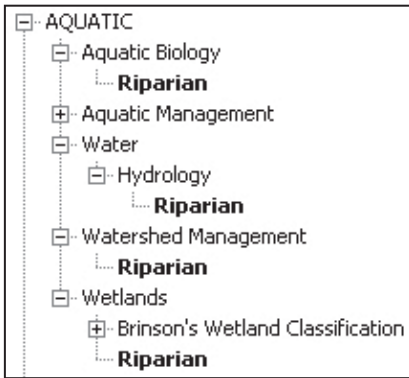


Fig. 1. Multiple Occurrences of *Riparian*

of the same term (i.e., with a different path) – one that more accurately described the concept that they wanted to express.

3 Related Work

The Metadata++ model is quite similar to a standard thesaurus [1,5] with a couple of generalizations that extend the model to support the specific needs of natural resource management. A standard thesaurus distinguishes between a single concept that belongs in multiple categories (using polyhierarchies) and homonyms that describe different concepts (using parenthetical qualifiers). Metadata++ unifies both issues using multiple occurrences, where the same term can appear in multiple hierarchies – but it may be entirely different connotations of the term (specified by the path). MeSH [4] (Medical Subject Headings) was designed to reflect a view of the literature for a user – without being confined to specific types of hierarchical relationships. The Metadata++ hierarchical structure was designed to reflect the discourses of the various domains. Both MeSH and Metadata++ support polyhierarchies – though MeSH defines Polyhierarchy as the same concept within multiple categories (e.g. *Nose* within *Sense Organs* versus *Nose* within *Face*).

References

1. ANSI/NISO Z39.19 – 2003. *Guidelines for the Construction, Format, and Management of Monolingual Thesauri*. NISO Press, 2003.
2. Bates, Marcia J. "Subject Access in Online Catalogs: A Design Model." *Journal of the American Society for Information Science*. 37 (6). 1986, pp 357-376.
3. Delcambre, L., T. Tolle, "Harvesting information to Sustain Forests", *CACM*, 2003.
4. Nelson, Stuart J.; Johnston, Douglas, Humphreys, Betsy L."Relationships in Medical Subject Headings." *Relationships in the organization of knowledge*. New York: Kluwer Academic Publishers; 2001. p.171-184.
5. Soergel, Dagobert. *Indexing Languages and Thesauri: Construction and Maintenance*. Wiley-Becker & Hayes Series Book. Los Angeles, CA, Melville Publishing, 1974.
6. Staab, S., J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.P. Schnurr, R. Studer, Y. Sure. "Semantic community web portals." *Computer Network*, 2000.

Multiple occurrences are also used to represent homographs. For example, the term *dolphin* is a marine mammal found within the *WILDLIFE* subject area and is also a term used to describe an inanimate, submerged object within *Aquatic Habitat Elements*. The path associated with these two occurrences of the term easily makes the distinct meanings clear. During our user study, participants knowingly selected the paths that were most appropriate for the particular task. In some cases, they selected a path, then later removed the path and selected a different occurrence

Generating and Evaluating Automatic Metadata for Educational Resources

Elizabeth D. Liddy, Jiangping Chen, Christina M. Finneran, Anne R. Diekema, Sarah C. Harwell, and Ozgur Yilmazel

Center for Natural Language Processing (CNLP) School of Information Studies
Syracuse University

Abstract. Metadata provides a higher-level description of digital library resources and serves as a searchable record for browsing and accessing digital library content. However, manually assigning metadata is a resource-consuming task for which Natural Language Processing (NLP) can provide a solution. This poster coalesces the findings from research and development accomplished across two multi-year digital library metadata generation and evaluation projects and suggests how the lessons learned might benefit digital libraries with the need for high-quality, but efficient metadata assignment for their resources.

1 Introduction

In 2000, we at CNLP recognized the potential of NLP methods and technology for the emerging field of digital libraries. At that time, metadata generation was a bottleneck in the process of getting sufficient resources available in the new digital libraries to ensure their success. We set forth to specialize our NLP technology to the task of assigning values to the Dublin Core metadata elements for individual items in digital educational resources collections.

2 Breaking the Metadata Generation Bottleneck

In the first project, the discourse linguistic method of sublanguage analysis was used to develop rules which identified and extracted the features of educational materials which should be captured as values for the various Dublin Core metadata elements. The resulting system, *MetaExtractTM*, compiles output from three distinct extraction modules, along with information from the collection-level configuration file as a text file metadata record.¹

We conducted a user study comparing 390 manually generated metadata records (MGMR) with those generated automatically by *MetaExtractTM*. 26 education students and teachers each examined 15 records (195 automatically generated records (AGR) and 195 MGMR). 20 individuals looked at the lesson

¹ Yilmazel, O., Finneran, C.M. and Liddy, E.D. MetaExtract: An NLP System to Automatically Assign Metadata. Proceedings of the JCDL 2004.

plan first and then judged the metadata compared to the lesson plan. The remaining 6 individuals looked at the metadata first and then the lesson plan to see if the lesson plan matched the expectation generated by the metadata. On a scale of one to five, MGMR did only slightly better than AGR. These results were promising as the difference in the scores of the two types of records was very small and this was the first attempt at automating metadata generation.

3 MetaTest

Encouraged by MetaExtract's ability to automatically assign high-quality metadata values, we conducted 3 studies to gain a fuller understanding of the quality and utility of automatically generated metadata. In the first study we tested the quality of each metadata element produced by *MetaExtract*TM. Analysis showed no statistically significant difference between manual and automatic generation methods for Description, Grade, Duration, Essential Resources, Pedagogy (Teaching Method or Group). Manually generated Title and Keyword elements had slightly higher scores.

The Metadata Information Retrieval Experiment was a comparative analysis of the retrieval effectiveness of MGMR, AGR, and free text alone (no metadata). With the fielded search, there was no statistical difference at any of the precision ranks between the manually generated and automatic metadata. With the un-fielded search, there was only a statistical difference at rank 10, with manual metadata precision slightly lower. With the baseline search, there was no statistical difference at any of the precision ranks.

A Metadata User Study investigated the effects of domain knowledge and feedback on search term selection and reformulation and explored differences between experts and novices. Those searchers who viewed both description and metadata tags were significantly faster in their search term entries. Searchers in the Metadata only condition decreased their search time by almost 10 seconds. These findings suggest that the availability of metadata enhances search time efficiency.

4 Conclusions

Having conducted two distinct digital library projects that implemented, compared, and evaluated metadata generation methods and metadata utility has enabled us to reach some unified conclusions. First, compared to manually assigned metadata, automatically assigned metadata was qualitatively evaluated as comparable for most elements. Second, automatic, manual, and no metadata performed comparably in retrieval, while automatically assigned metadata achieved better coverage of metadata elements. These key findings revealed that NLP can be utilized to automatically produce metadata records in an efficient and effective manner.

Web Service Providers: A New Role in the Open Archives Initiative?

Extended Abstract

Manuel Llavador¹, José H. Canós¹, and Marcos R.S. Borges²

¹ Dept. of Computer Science (DSIC),
Technical University of Valencia,
E-46022 Valencia, Spain,
+34 96 387 7007 Ext. 83525

{mllavador, jhcanos }@dsic.upv.es

² Núcleo de Computação Eletrônica and Instituto de Matemática,
Universidade Federal do Rio de Janeiro, Brasil
mborges@nce.ufrj.br

Service Oriented Computing [1] is consolidating as the dominant paradigm for software development of this decade. The support it has received from researchers, practitioners and —most important— the software industry demonstrates the suitability of this approach. This means that current software systems must evolve in this direction in order to keep aligned with technology, providing a set of services that can be invoked by programs instead of end users.

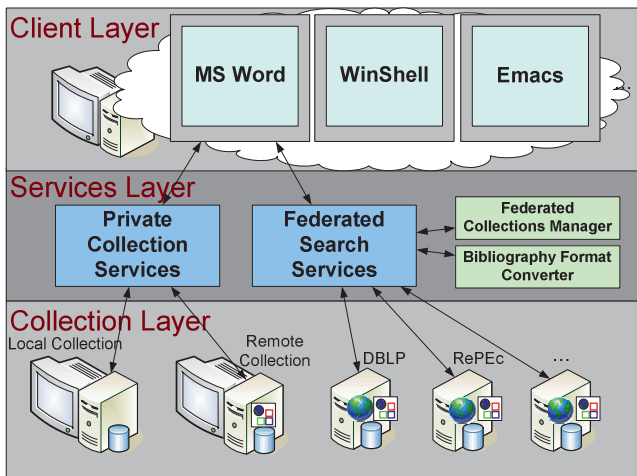


Fig. 1. Architecture of BibShare

An example of the service oriented approach is BibShare [2], a framework that aims to unify bibliography management across operating systems and word processors. BibShare has been conceived as a set of bibliography management

services that can be invoked by clients embedded into different word processors. The services include creation and management of bibliography collections, local and federated searches, and bibliographic format transformation. The BibShare functionality has been implemented following the Service-Oriented Computing principles: all the bibliography management services are implemented as XML Web Services, and can be invoked by the clients using the SOAP protocol [3].

Figure 1 shows the architecture of the framework, where three different layers are highlighted. The Client Layer is where the word processing systems are placed. Embedding a BibShare client into a word processor enables its users to retrieve bibliographic information from different collections, insert them into a document, and generate the bibliography automatically using different bibliographic styles and labelling schemas.

At the Collection Layer, a number of autonomous, heterogeneous and distributed collections of bibliographic data are registered, eventually cached and used to support federated searches. Some of them are accessible to their regular users via a web interface, whereas others are private, that is, their owners are individuals or small groups.

The Services Layer includes private and federated search Web services. Private search services allow the retrieval of bibliographic records from private collections. Federated searches are supported by the Federated Search Engine (FSE), the Federated Collections Manager and the Bibliography Format Converter. Notice that the BibShare search services are not end-user oriented, as they are called by the clients modules embedded into the word processors, as in typical service oriented computing scenarios.

The FSE is a Web service supporting searches in heterogeneous collections that offers two different search methods:

- *Query*: accepts a search criterion plus a list of collections as input and then returns the set of bibliographic records from these collections that match the query. Internally, the FSE propagates the query to the collections in the input list, collects the record lists from each collection, and joins all the records in an XML document which is returned to the caller.
- *GetRecord*: allows clients to get the full record, which corresponds to a specific reference from a federated repository. Internally, the FSE sends a request to the repository holding the record, which transforms it into the Bibshare Bibliographic Format before returning it to the client.

OAI-BibShare is the Service Provider developed to make all the Open Archives Initiative (OAI) [4] Data Providers available to BibShare users. The most remarkable characteristic of OAI-BibShare is that, to the best of our knowledge, is the first Service Provider which is not end-user oriented. Rather, it is called by the FSE every time any OAI collection is included in a *Query* request by some BibShare client. OAI-BibShare was implemented as a Web service which can be called in the same way that any other collection server in the BibShare Federation.

Figure 2 shows how the OAI-BibShare is integrated in the architecture of BibShare. *FSEWS.asmx* is the Web service implementation of the BibShare federated search service. The Format Converter and the Collection Manager are the modules that perform the record format transformation using XSL templates and the federated

collections management, respectively. *OAIWrapper.asmx*: it is the Web service that publishes the *Query* and *GetRecord* methods of the OAI harvesters. Finally, the BibShare Server is the main server of the BibShare environment. It implements the Services Layer and includes some additional components, such as the database where the OAI collections are cached.

The three layered architecture of the BibShare framework provides many features, such as the extensibility both at the client and collection layer, as discussed in [3]. There we detail how extensibility is granted in the case of the OAI repositories. At the BibShare Services Layer (see Figure 1) there is a module called Federated Collections Manager which allows adding, removal and modifying collections in the BibShare federation. It uses the information about collections held in the Federated Repositories Database (see Figure 2) and it includes the following attributes in all repositories:

- *repositoryId*: used to identify every OAI repository
- *name*: Name of the repository that will be listed in the BibShare clients.
- *WSUrl*: URL where the data access services are available.
- *XSLTPath*: Location of the XSL template.

In general, this information must be supplied by collection managers at the registration time. However, for the OAI case, we have automated the process in two steps. First, we obtain the updated list of registered data providers from the OAI website. Then, we can use the verb *Identify* to retrieve information about any repository (specifically the repository identifier, the repository name and the URL of the protocol) This allow us to periodically check the list of data providers in order to automatically include them in the BibShare federation. The OAI Manager module in Figure 2 is in charge of adding new data providers, as well as caching and updating their metadata.

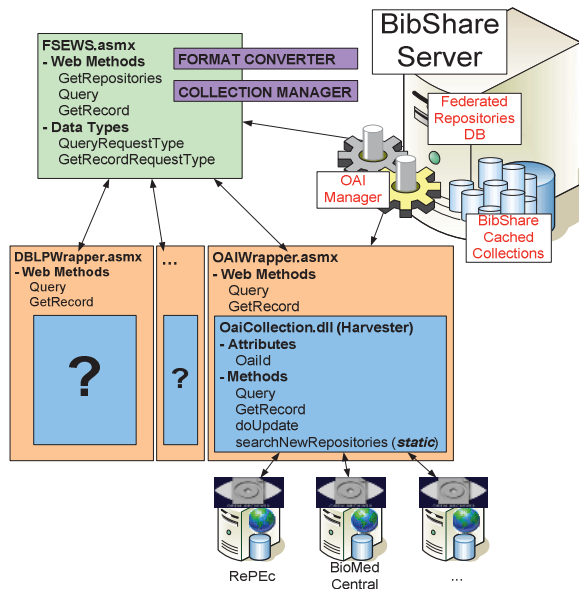


Fig. 2. OAI management in BibShare

The Service Oriented Computing has been changing the way applications are conceived. In the case of the OAI, we firmly believe that Web Service Providers will play an important role in the near future, and that this issue deserves some reflection within the OAI community in order to resolve a number of important issues such as, service description, publication, and others.

References

- [1] Tsalgatidou, A. and Pilioura, T., An Overview of Standards and Related Technology in Web Services. *Distributed and Parallel Databases*, 12, 2/3 (Sept./Nov. 2002), 135-162.
- [2] The BibShare project. <http://www.bibshare.org>
- [3] Canós, J. H., Llavador, M, Solís, C. and Ruiz, E. A Service-Oriented Framework for Bibliography Management. *D-Lib Magazine* 10, 11 (Nov. 2004) , <http://www.dlib.org/dlib/november04/canos/11canos.html>
- [4] Open Archives Initiative. <http://www.openarchives.org/>

DiCoMo: An Algorithm Based Method to Estimate Digitization Costs in Digital Libraries*

Alejandro Bia¹ and Jaime Gómez²

¹ Miguel Hernández University, Spain
abia@umh.es

² University of Alicante, Spain
jgomez@dlsi.ua.es

Abstract. The estimate of web-content production costs is a very difficult task. It is difficult to make exact predictions due to the great quantity of unknown factors. However, digitization projects need to have a precise idea of the economic costs and times involved in the development of their contents. As it happens with software development projects, incorrect estimates give way to delays and costs overdrafts. Based on methods used in Software Engineering for software development cost prediction like COCOMO [1]) and Function Points [2], and using historical data gathered during five years of work at the Miguel de Cervantes Digital Library, where more than 12.000 books were digitized, we have refined an equation for digitization cost estimates named DiCoMo (Digitization Cost Model). This method can be adapted to different production processes, like the production of digital XML or HTML texts using scanning plus OCR and human proofreading, or the production of digital facsimiles (scanning images without OCR). The estimates done a priori are improved as the project evolves by means of adjustments based on real data obtained from previous stages of the production process. Each estimate is a refinement obtained as a result of the work done so far.

The Digitization Cost Model

In the digitization cost model we propose, we use an equation similar to Intermediate COCOMO [1], but with some differences:

Size-Independent Overhead: We added a new term called *SIO (Size-Independent Overhead)*, which represents the preparation time for the task, which is independent from its size. An example of this size-independent overhead is the time needed to adjust the parameters of an image scanner and OCR before starting a scanning session. This is a fixed time which does not depend on the number of pages to be scanned later.

No Uncertainty in Digitization: One of the reasons why COCOMO often fails in estimating software costs is because its calculations are based on an estimated size (KLOC), which is highly uncertain at the initial stages of the project. When applying a similar method to estimate digitization costs, the first thing we realize is that we don't have to estimate the size of the work because we already know it. The size of

* This work has been supported by the Ministry of Education and Science of Spain through grant TIN2004-00779.

the document to digitize is measured as the number of pages P and is perfectly measurable beforehand.

The Importance of Historical Cost-Data: Inside most organizations, the estimation of production costs is usually based on past experiences. Historical data are used to identify the cost factors and to determine their relative importance within the organization. Historical data will be used to adjust cost estimation algorithms, and the detected cost factors to set estimate modifiers for the algorithms. This is the reason why it is so important to systematically store current project's costs data. Different literary works have different degrees of difficulty owed to several factors (to be discussed in the poster session), which will affect production times. We have detected the most important of these factors, and assigned weights to them. Then we added an effort adjusting factor EAF to the equation, equivalent to the one used in Intermediate-COCOMO, but based on specific digitization features. The EAF is calculated as the multiplication of all relevant feature modifiers.

Factors That Affect Digitization Costs: There are many factors that influence the production costs of digital objects. Both these factors and their effect on costs were difficult to determine and had to be carefully studied. Among them, we can highlight the individual skills and experience of the persons assigned to the project, their familiarity with the specific characteristics of the work to be digitized and with the computer tools to be used. Also important are some features of the document that affect digitization times like: foreign or ancient languages, stained paper, old font faces, high quality demands, inadequate technology used, special care required for old books, etc. An abridged list of the complexity modifiers used to calculate the EAF is shown besides.

Modifier	Low	Normal	High
encoder experience & skills	1.30	1.00	0.70
familiarity with task	1.20	1.00	0.80
familiarity with computer tools	1.20	1.00	0.80
foreign or ancient languages present	---	1.00	1.25
stained or old paper	---	1.00	1.15
old font faces	---	1.00	1.15
special care required (ancient books)	0.80	1.00	1.20
high quality demands	0.80	1.00	1.20
inadequate technology used	0.80	1.00	1.20

Now we can directly calculate the time in hours T using the DiCoMo formula. In the poster we include a visual example of this approach, where an estimation curve approaches real data spots that represent real measures of digitized documents. The following estimation equation (left), adjusted with values from real cases (right), gives us the estimated number of hours to digitize a text given the number of pages:

$$T = a \cdot P^b \cdot EAF + SIO \qquad T = 0.06 \cdot P^{1.47} \cdot EAF + 2 \qquad (1)$$

$$EAF = \prod modifier_i \qquad (2)$$

For instance, a standard-complexity book of 100 pages will take about 54 hours of scanning, correction and XML markup altogether, according to this estimation.

References

1. Barry W. Boehm. Software Engineering Economics. Prentice Hall, Englewood Cliffs, N.J., 1981.
2. A.J. Albrecht. Measuring Application Development Productivity. In Proceedings of the Joint Share/Guide/IBM Applications Development Symposium, pages 83-92, October 1979.

Adapting Kepler Framework for Enriching Institutional Repositories: An Experimental Study

A. Ramnishath, Francis Jayakanth, Filbert Minj, and T.B. Rajashekar

National Centre for Science Information, Indian Institute of Science, Bangalore, India
ramnishath@gmail.com,
{franc, filbert, raja}@ncsi.iisc.ernet.in

1 Introduction

There is growing trend towards academic and research organizations to establish OAI-compliant institutional repositories. ePrints@IISc (<http://eprints.iisc.ernet.in/>) is the institutional repository of Indian Institute of Science (IISc), Bangalore. Though the repository is growing steadily, mediated submission by the ePrints@IISc staff is the predominant mode of enriching the repository. We have been exploring viable means of getting our researchers to contribute more actively to the repository. Observations have recently been made as to why researchers might be reluctant to contribute to central repositories [1]. It has been suggested that it might be useful to provide researchers with tools to easily create and share Personal Digital Repositories (PDR) designed to organize and facilitate their research and learning agendas. The collection in the PDR is built and managed by the scholar based on individual needs. A network of such PDRs could form the basis for a bottom up, organic approach to enrich centralized institutional repositories.

2 Implementation Details

We carried out a study to explore the bottom up approach using the Kepler Communal Digital Library Framework [2, 3]. In adapting the Kepler framework, our objectives were: a) To enhance Kepler Archivelet to serve as a PDR for the individual researcher, to provide controlled web access to publications in the PDR, and allow controlled exposure of metadata and full text for harvesting by the Kepler Group Server, and b) To enhance Kepler Group Server to enforce document type and metadata policies of the ePrints@IISc repository on the network of registered PDRs and to harvest metadata and full text from individual PDRs as per the harvesting rights permitted by the PDRs and ingest them to ePrints@IISc repository. Key requirement was to enforce the collection policies of the ePrints@IISc repository at PDR level for content publishing, and at the Kepler Group Server level for harvesting permitted content from PDRs. This was achieved by introducing three key enhancements:

- A 'foreign.xml' file at the Group Server level, configurable by the ePrints@IISc repository manager reflecting the collection policies of the repository, to be conformed to by PDRs. Each PDR registering with the Group Server automatically downloads this file and is used during content publishing

- Customized metadata driver at PDR level imposing the policies encoded in 'foreign.xml' file
- Customized 'NCSI-OAI-DC' metadata scheme for harvesting by the Group Server as per the collection policies, in addition to universal OAI-DC module.

Apart from carrying out these specific enhancements, we could successfully achieve the following:

- Adapted the Kepler Archivelet and Group Server framework in a centralized institutional repository setting, to support bottom up content publishing and repository level content aggregation as per the collection policy requirements of the central repository
- Enhanced the Kepler Archivelet to develop a prototype PDR supporting key features such as: richer menu of document types (e.g. conference papers, presentations, preprints) for publishing including metadata specific to these document types; option to register with the Group Server; option to produce a PDR home page based on a configurable style sheet showing personal details of the researcher and archived publications providing an additional means to reach users on the Internet; and option to manage access rights indicating which publications to appear on the PDR home page and which publications can be harvested at metadata and/or full text level
- Enhanced the Kepler Group Server to enforce key collection policies (e.g. document types, metadata and subject categories) on registered PDRs; to harvest metadata and full text as per these policies and the access rights defined by PDRs; and to export this content in XML format as per the import requirements of ePrints@IISc repository, and
- Imported the content to ePrints@IISc repository with provision for indicating possible duplicates, for review by the repository moderator

While this study was carried out in the specific institutional repository set up at IISc, which uses the EPrints.org repository software, the enhancements we have carried out could be applicable in different repository software situations.

References

1. Gandel, Paul, B, Katz, Richard, N, Metros, Susan, E.: The weariness of the flesh: Reflection on the life of the mind in an era of abundance. *EDUCAUSE Review*, Vol. 39, no. 2 (2004) 40-51
2. Liu, X, Kurt, M, Zubair, M.: Enhanced Kepler Framework for Self-Archiving. *Proceedings of the 2002 International Conference on Parallel Processing Workshops*, p.455, August 18-21 (2002)
3. Maly, K, Nelson, M, Zubair, M, Amrou, A, Kothamasa, S, Wang, L.: Kepler – A Communal Digital Library. Technical report, Old Dominion University (2004)

The Construction of a Chinese Rubbings Digital Library: An Attempt in Preserving and Utilizing Chinese Cultural Heritage Materials

Guohui Li¹ and Michael Bailou Huang²

¹ The Graduate School Library, People's Bank of China,
Beijing, China

liguohui@gspbc.edu.cn

² Health Sciences Center Library,

Stony Brook University

Stony Brook, NY 11794-8034, USA

michael.b.huang@stonybrook.edu

China is a country with an ancient civilization going back 5,000 years. Keeping records on inscriptions is an important method of preserving the memory of Chinese history and culture. Rubbings are important components of ancient Chinese books, and are the main source for people to learn, study, and research history. The construction of a Chinese rubbings digital library is an attempt to solve the problems of preserving and utilizing cultural heritage materials. This poster will discuss the following topics: (1) technical process of constructing a Chinese rubbings digital library; (2) formulating principles and designing metadata standards for the Chinese rubbings digital library; (3) introduction of four prototype databases; and (4) analysis of existing problems in building a rubbings digital library such as data capacity, system functions, metadata standards, and international cooperation.

Rubbings are an important part of ancient Chinese book collections. They are a type of historical documents created by placing paper over an image on a raised, indented, or textured surface and rubbing the paper. More than any other civilization, the Chinese have long relied on carving inscriptions into stone as a way of preserving their history and culture. In addition to stone, materials such as brick, tile, ceramics, bronze ware, wood, and jade were also engraved to preserve writings and pictorial representations. Some rubbings have survived to date. They are one of the major sources that people use to study and research history. The rubbing itself is paper, and how to guarantee access while preserving the document has always been a dilemma. The advent of the digital library has provided a gleam of hope as a solution to this problem. Many libraries and institutions worldwide with collections of Chinese rubbings have done extensive research and have been in practice for several years. Some prominent Chinese rubbings digital collections are: (1) Resource Database of Chinese Rubbings at the National Library of China, (2) Peking University Library Ancient Documents Database, (3) Database of Bronze Image Rubbings collected in the Institute of History and Philology, Academia Sinica in Taipei, and (4) Chinese Stone Rubbings Collection at the UC Berkeley East Asian Library. At present, the construction of a Chinese rubbings digital library is still in its initial stage. Most of the work is mainly formulating metadata standards, arranging rubbings, digitizing rubbings, and building prototypes of rubbings digital library.

For a long period, researchers have to use a card catalog to access rubbings housed in various libraries and institutions. Even after the library adopted large scale computer systems, card catalog records to access rubbings were not converted to machine-readable formats. Due to specific characteristics of rubbings, field descriptions for rubbings are quite different from those for monograph records in card catalogs. We still need to look at the original in order to digitize rubbings. For instance, each record from Jiagu Rubbings Database of the National Library of China consists of the library collection number, source number, division of history into periods, place of excavation, property of original bone, size of the original bone, rubbing source, rubbing size, and so on. Usually, there are two ways to digitize rubbings: taking pictures, or scanning the original. Both methods have advantages and disadvantages, and they will be compared in details in the poster. A metadata standard is a key issue in building a rubbings digital library. It determines the design of library managing system and large scale inputting of rubbings records directly. There has never been a standard to guide recording and cataloging of rubbings throughout history. Therefore, in building a digital library, there is no metadata standard such as USMARK or CNMARK that librarians normally use to catalog books. Presently, Dublin Core (DC) is a popular and widely supported metadata standard. Peking University Libraries formulated a rubbings metadata standard based on DC, and designed an experimental system that uses Chinese metadata standard to record Chinese rubbings.

The construction of a Chinese rubbings digital library has solved the contradiction of preservation and utilization of ancient documents to certain extent. It also provides more convenient and faster access to rubbings and related documents. Some robust digital library systems even offer study help to novice users. Nevertheless, some problems in the building of a rubbings digital library need to be addressed and solved. Chinese rubbings, by nature, are reproductions. It is important to preserve the details of the paper reproduction as well as provide information on the original material such as, stone, wood, bronze, and so on. One of the greatest challenges in providing descriptive metadata for Chinese rubbings is to transcribe the information of the "original." Some database fields will need to be added or deleted while others will need to be refined or modified. At present, different institutions wish to use their own guidelines and standards. It is urgent to set up a metadata standard for Chinese rubbings for all participating institutions to use. Ideally, all participating institutions will merge their holdings into a Chinese rubbings union catalog, and their collections would be accessible through a single web interface for distributed sites. Another problem is, compared with book and journal automation systems, the Chinese rubbings digital library management systems are not user-friendly and flavored. There has been some cooperation among participating institutions in building the Chinese rubbings digital library, but that's not enough. We should promote more international library cooperation so valuable time and resources are not wasted.

Policy Model for National and Academic Digital Collections

Alexandros Koulouris and Sarantos Kapidakis

Laboratory on Digital Libraries and Electronic Publishing, Department of Archive and Library Sciences, Ionian University, Plateia Eleftherias, Palaia Anaktora, Corfu 49100, Greece
akoul@ionio.gr, sarantos@ionio.gr

Abstract. The access and reproduction policies of the digital collections of fifteen leading academic and national digital libraries worldwide are classified according to factors such as the creation type of the material, acquisition method and copyright ownership. The relationship of these factors and policies is analyzed and quantitative remarks are extracted. We propose a policy model for the digital content of the national and academic libraries. The model consists of rules, supplemented by their exceptions, about which factors lead to specific policies. We derive new policy rules on access and reproduction when different copyright terms are applied. We conclude with findings on policies. Finally, we compare national and academic library policies, showing interesting results that arise on their similarities and differences.

1 Overview of Problem, Methodology and Results

Libraries are in a transition period from conventional to digital formats and have not yet developed common practices and traditions on policy for digital material. This can prevent cooperation and interoperability in libraries, restricting the usefulness of their services. Conventional policies do not map directly onto digital policies with differences primarily due to the easy duplication properties of the digital material. However, national and academic libraries have well established traditions of cooperation and they play a leading role in the production and dissemination of digital material and in the development of digital libraries.

The access and reproduction policies of the digital collections of thirty-five leading digital libraries worldwide, twenty academic and fifteen national, were examined. Fifteen of them, ten academic and five national, are presented in this poster, those, which are considered to have the most diversified and innovative access and reproduction policies, and are the most active in the area of digital libraries. We were interested in libraries which have collections with various creation types (digitized, born-digital) or content types of material (video, audio etc.), various copyright owners (libraries, individuals, organizations such as publishers etc.), diversified access and reproduction policies and various acquisition methods (license, purchase etc.). To collect the data, we derived information from the websites, in some cases supplemented by personal communication with the libraries and from relevant studies [1, 2]. The access and reproduction policies of the examined digital collections are classified

according to factors of creation type, acquisition method and copyright ownership; their relationship is analyzed and quantitative remarks are extracted.

From this analysis, we derive a policy model for the digital content of the national and academic libraries. The proposed policy model is not only comprised of the most common policies – practices that the libraries implement, but also, of new ones that have not been implemented so far, and can offer solutions to problems of access, reproduction and digital content management. The policy model contains rules, supplemented by their exceptions, about which factors lead to specific policies. Due to the differences on policies according to the creation type factor, we divide the policy model onto two separate ones, for the digitized and the digital material respectively, by extracting the relevant conclusions that are valid on each case.

On the two models, we make proposals and recommendations on policies when different copyright terms are applied. For example, what policies can be used when the library has the copyright to the material or when organizations, such as publishers, have it, or when multiple copyright owners exist, or when it varies from item to item? What policy rules apply for onsite (on and off-campus) and offsite access on each case? When and why is the onsite and offsite access the same or diversified? When and why on and off-campus onsite access is different? What combinations of access policies can we face on each case and why? Why is there a distinction between private and commercial reproduction? When and why is fair use doctrine used for private reproduction? Under which factors does the commercial reproduction need written permission and fees, or is prohibited and why?

At the end, we present our findings on policies. Some findings are common for national and academic libraries; for example, the on-campus onsite access is always free, independent of copyright ownership and the creation type of the digital content. In addition, when the library has the copyright of the digital content, the private reproduction is usually provided free with a credit to the source (creator, author) or otherwise mostly under fair use provisions, but the commercial reproduction needs written permission and fees are charged. However, some findings are unique. For example, academic libraries decide the provision of reproduction (private and commercial) on case-by-case basis, when there are copyright uncertainties. On the other hand, national libraries may require fee for the reproduction not for copyright but for preservation reasons; and the copyright fee, which may be required by the copyright owner, is added to the reproduction fee. Finally, national and academic policies are compared and interesting results, on similarities and differences, arise.

References

1. Meyyappan, N., 2000. A review of the status of 20 digital libraries, *Journal of Information Science*, 26 (5) pp. 337-355
2. Walters, W.H., 2003. Video media acquisitions in a college library, *Library Resources & Technical Services*, 47 (4) pp. 160-170

A Framework for Supporting Common Search Strategies in DAFFODIL

Sascha Kriewel, Claus-Peter Klas, Sven Frankmölle, and Norbert Fuhr

University of Duisburg-Essen

{kriewel, klas, frankmoelle, fuhr}@is.informatik.uni-duisburg.de

Abstract. DAFFODIL is a front-end to federated, heterogeneous digital libraries targeting at strategic support of users during the information seeking process by offering a variety of functions for searching, exploring and managing digital library objects. In the process of searching for information, common strategies and tactics emerge that can be reused in different searches and different contexts. This poster presents the framework that will be used to build a search support system that provides the possibilities to define and recognize such common strategies and tactics, to save and reuse them, to build larger search plans from these parts, and to support automatic execution of partial or complete search strategies.

1 Introduction

Based on empirical observations of the information seeking behaviour of experienced library users, Bates [1,2] identified a number of successful tactics for the information search. In [3] tactics referring to *monitoring*, *file structure*, *search formulation*, *term selection*, and *ideas* are described. Building on these tactics recurring strategems can be defined, e.g. for doing an author run over a collection of documents.

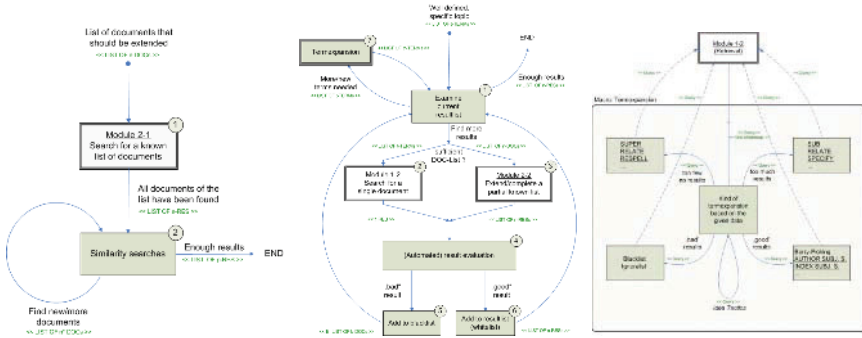
In DAFFODIL [4],[5], a virtual digital library, support for various common tactics and strategems is supported in a number of tools. Users can reuse results of one search tool as input of another, thus creating a *chain of tool uses* that compose a strategy for a specific search.

Often these strategies contain patterns of tool usage that are repeated over several different searches. Also, a number of common, simple search tasks lend themselves to common strategies. There is however no support in current digital library search systems to help the user to build, save, or execute these search plans besides going through all the steps of the strategy for every search the user does. We claim that we can raise efficiency and effectiveness, given a strategic framework and implementations for common strategies.

2 Strategic Framework

To address these problems in the context of the digital library system DAFFODIL a framework for strategic support was developed that provides modular strategies for common search tasks, as well as macros that can be used by the user as building blocks to create larger search plans [6].

For a number of common search tasks modules were developed that provide strategies which can be suggested to users in need of strategic support as possible search plans. Two dimensions of search tasks were found to be especially important in developing these basic building blocks: complexity and domain knowledge of the user. Using these two dimensions search tasks were categorized, and strategies were created for each combination.



(a) A simple strategic module (b) A complex strategic module (c) A macro for query term expansion

3 Outlook

Preliminary user questionnaires strengthen our belief that the strategic modules are adequate for representing basic strategies for common search tasks. Using this strategic framework we plan to extend the existing strategic support of DAFFODIL to include the possibilities for the user to define their own strategies by using the modules, macros, and tactics provided, and to save and reuse these strategies. We also want to enable DAFFODIL to provide suggestions of search plans appropriate to the user’s search task and current situation, and to be able to execute complete or partial search plans automatically or user-guided.

References

1. Bates, M.J.: Information search tactics. *Journal of the American Society for Information Science* **30** (1979) 205–214
2. Bates, M.J.: Idea tactics. *Journal of the American Society for Information Science* **30** (1979) 280–289
3. Bates, M.J.: Where should the person stop and the information search interface start? *Information Processing and Management* **26** (1990) 575–591
4. Klas, C.P., Fuhr, N., Schaefer, A.: Evaluating strategic support for information access in the DAFFODIL system. In: *ECDL 2004*, Springer (2004)
5. Fuhr, N., Klas, C.P., Schaefer, A., Mutschke, P.: Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In: *ECDL 2002*, Springer (2002) 597–612
6. Frankmölle, S.: Strategien zur Suche in Digitalen Bibliotheken. Master’s thesis, Universität Dortmund, FB Informatik (2004)

Searching Cross-Language Metadata with Automatically Structured Queries

Víctor Peinado, Fernando López-Ostenero, Julio Gonzalo, and Felisa Verdejo

NLP Group - UNED. ETSI. de Informática,
28040 Ciudad Universitaria, Madrid, Spain
{victor, flopez, julio, felisa}@lsi.uned.es
<http://nlp.uned.es>

1 Introduction

When searching metadata, it can be useful to detect expressions in the query that should be searched for in specific fields (for instance, person names might correspond to an “author” field). In [1], it was shown that automatically structured queries (matching title, abstract, author and publication fields) improved effectiveness when searching the ACM, CITIDEL and NDLTD Computing Digital Libraries.

In a cross-language retrieval setting, we can decide, in addition, how to translate different types of information (named entities, temporal references, quantities, etc.) once they are automatically detected in the query.

This is the approach to cross-language metadata search that we test in this paper. We experiment with a simple strategy that **i)** locates proper names, temporal references and numbers in the query; **ii)** attempts to classify them by checking whether they appear as “author”, “location” or “date” in the collection; **iii)** uses positive cases to structure the query, forcing the search engine to favour documents with the appropriate author, location or date.

2 Experimental Settings

We have used the Spanish-English version of the ImageCLEF ad hoc task testbed. In order to locate and identify named entities, temporal references and numbers appearing in the ImageCLEF queries, we have used a simple strategy that was enough for our purposes (see [2] for further details).

Then, we have compared three approaches: **i)** a **naive baseline** using a word by word translation. **ii)** a **strong baseline** following Pirkola’s proposal [3], where alternative translations for a query term are taken as synonyms, giving them equal weights, and; **iii)** our **structured query approach**, which incorporates **field search** operators in addition to Pirkola’s strategy. All three conditions have been tried with six different bilingual dictionaries.

Besides, we have evaluated three additional runs for comparison purposes: two monolingual runs (a straight run with the English version of the query, and an enhanced run with the field search strategy described in [2], and an additional

cross-language run where named entities and temporal references are annotated manually. The latter is intended to evaluate the effects of errors in the automatic location of entities.

3 Results and Discussion

The results of the experiment are shown in Table 1. For all bilingual dictionaries, our structured query approach is better than the naive and Pirkola baselines. Pirkola’s approach is, in turn, substantially better than the naive run in all cases. Only the differences between our structured query approach and the naive baselines are relevant according to a non-parametric Wilcoxon sign test (in half of the cases). Our best runs achieve an average precision of .54, which represents 91% of the best monolingual run (*monolingual+field search*). This result slightly outperforms the best official cross-language run in the ImageCLEF 2004 evaluation (which was .53, obtained by Dublin City University with the DE→EN language pair).

Remarkably, the manual annotation of named entities and temporal expressions does not improve the results obtained with our simple automatic recognition strategy. This is an indication that the field search strategy is reasonably robust: for instance, if an expression is misinterpreted as a person name, it will probably not appear in the author field and therefore precision will hardly be affected.

Table 1. Experimental results. Non-interpolated mean average precision (MAP) for different combinations of retrieval strategy and bilingual dictionary. “*” denotes a statistically significant difference with respect to its naive counterpart.

Dictionary	naive	Pirkola	field search	Additional reference runs	MAP
FreeDict	.34	.38	.42		
EWN	.36	.50	.52*	Monolingual base	.57
EWN2	.38	.51	.54*	Monolingual+field search	.59
Vox	.40	.45	.53	CL manual field search	.54
All-Vox	.34	.52	.54*	Best CL ImageCLEF run	.53
All	.37	.49	.53		

References

1. M. A. Gonçalves and E. A. Fox and A. Krowne and P. Calado and A. H. F. Laender and A. S. d. Silva and B. Ribeiro-Neto. The Effectiveness of Automatically Structured Queries in Digital Libraries. JCDL 2004
2. V. Peinado and J. Artilles and F. López-Ostenero and J. Gonzalo and F. Verdejo. UNED at Image CLEF 2004: Detecting Named Entities and Noun Phrases for Automatic Query Expansion and Structuring. Cross Language Evaluation Forum, Working Notes for the CLEF 2004 Workshop (2004)
3. A. Pirkola. The Effects of Query Structure and Dictionary Setups in Dictionary-Based Cross-Language Information Retrieval. SIGIR’98 (1998) 55–63

Similarity and Duplicate Detection System for an OAI Compliant Federated Digital Library

Haseebulla M. Khan, Kurt Maly, and Mohammad Zubair

Computer Science Department, Old Dominion University, Norfolk VA-23529, USA
{hkhan, zubair, maly}@cs.odu.edu

The Open Archives Initiative (OAI) is making feasible to build high level services such as a federated search service that harvests metadata from different data providers using the OAI protocol for metadata harvesting (OAI-PMH) and provides a unified search interface. There are numerous challenges to build and maintain a federation service, and one of them is managing duplicates. Detecting exact duplicates where two records have identical set of metadata fields is straight-forward. The problem arises when two or more records differ slightly due to data entry errors, for example. Many duplicate detection algorithms exist, but are computationally intensive for large federated digital library. In this paper, we propose an efficient duplication detection algorithm for a large federated digital library like Arc.

1 Introduction

There are numerous challenges to build and maintain a federation service, and one of them is in managing duplicates. The duplicates arise either due to publication of the same record in multiple collections or due to hierarchical harvesting that is made possible by the OAI framework. Detecting exact duplicates where two records have identical set of metadata fields is straightforward. The problem arises when two or more records differ slightly due to data entry errors, for example. Duplicate detection problem has been extensively studied in the literature, for example [1-2], and many of the algorithms proposed are application specific. Most of these algorithms, for a large federated digital library consisting of records with multiple metadata fields, have a large over-head and are computationally intensive. In this paper, we propose an efficient duplicate detection algorithm for a large federated digital library like Arc[3].

In our approach, we first sort all records in the federation by their rank, where rank of a record is defined by how similar the record is to an *anchor* record (a randomly picked record). The similarity is measured for all metadata fields in the two records, though with different weights. We use a sliding window of size w to detect duplicates with a runtime complexity of $O(n*w)$, where n is the total number of records.

2 Experimental Results

The production Arc has over 7 million records and harvest 180 collections. For duplicate testing, we created a smaller subset of Arc consisting of 73 archives with 465,440

records. We selected title and author as the only attributes for our performance study. We plotted number of duplicates found against different window sizes. The results indicate that the number of duplicates found increases with the window size up to a point and then starts leveling. In Fig 1(b) we plotted the number of duplicates found when the weights for different metadata fields in the similarity measures are changed.

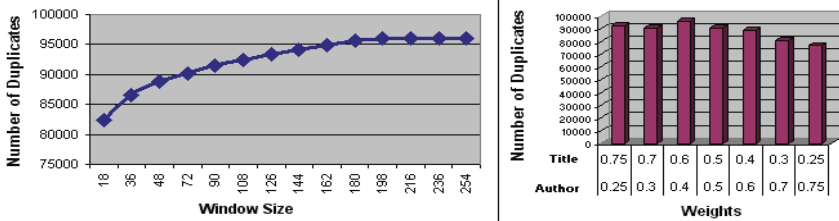


Fig. 1. (a) Window Size Vs Duplicates (b) Weights Vs Duplicates

Archive Name	Common records	Archive Name
lib.umich.edu	8992	Dlpscoll
xtcat.oclc.org	19637	Btdcat
arXiv	50487	Citebase
Citebase	801	Cogprints
lcoal.loc.gov	30850	lcoal
uiLib	5269	oai.library.uiuc.edu

Archives	Self Duplicates
lib.umich.edu	37623
xtcat.oclc.org	39242
lcoal.loc.gov	10064
lcoal	7872
uiLib	12988
RePEc	8777

Fig. 2. (a) Relationships between archives (b) Archives with self duplicates

The results show that the title’s weight is more important than that of the author’s weight. This is true as author names are not normalized and some times are represented by initials. Fig 2(a) shows selected archives that have duplicates in common. For example, the table indicates that “lcoal.loc.gov” and “lcoal” archives have around 30850 records with title-author similarity; they are both Library of Congress OAI Repositories with different versions of OAI-PMH. Fig 2(b) shows samples of archives with self duplicates.

References

- [1] Sam y. Sung Zhao li Peng Sun. “A Fast Filtering Scheme for Large Database Cleansing” *Proceedings 11th international conference on Inform & knowledge management, ACM2002*
- [2] M. Hernandez and S. Stolfo. The merge/purge problem for large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, May 1995.
- [3] ARC - A Cross Archive Search Service ODU, Digital Library Research Group

Sharing Academic Integrity Guidance: Working Towards a Digital Library Infrastructure

Samuel Leung¹, Karen Fill², David DiBiase³, and Andy Nelson⁴

¹ School of Geography, University of Southampton, Southampton, SO17 1BJ, UK
Y.Leung@soton.ac.uk

² CLT, University of Southampton, Southampton, SO17 1BJ, UK
kef@soton.ac.uk

³ John A. Dutton e-Education Institute, 2217 Earth-Engineering Sciences Building,
University Park, Penn State University, PA16802, USA
dibiase@psu.edu

⁴ IES, Joint Research Centre of EU, TP 440, Via Enrico Fermi 1, I-21020 Ispra (VA), Italy,
Previously School of Geography, University of Leeds, Leeds, LS2 9JT, UK
andrew.nelson@jrc.it

1 Introduction

This poster draws on the ‘Digital Libraries in Support of Innovative Approaches to Learning and Teaching in Geography’ (<http://www.dialogplus.org>) project under which geography teachers in two UK and two US universities are collaborating in the creation and sharing of reusable online learning activities. A specific aim of the project has been to explore the use of digital library (DL) infrastructures to enable the sharing of learning objects between the participating institutions.

This poster presents a brief overview of the broader project, but focuses on one case study drawn from our programme of work, whereby a learning activity or ‘nugget’ concerned with academic integrity, originally developed at Pennsylvania State University (PSU) in the USA for use by distance learning masters students, has subsequently been repurposed for campus based students at the Universities of Southampton and Leeds in the UK.

2 Technical Challenges

The Academic Integrity Guidelines (AIG) nugget is concerned with how to ensure that students understand the protocols of citation and referencing and thus avoid plagiarism. It has broad applicability beyond the discipline of geography, with potentially shareable content relevant to teaching and learning in all the partner institutions. The adoption of the nugget at each institution has allowed the project team to address the practical problems of mounting the same resources within our different online learning environments, and informed efforts towards an appropriate DL repository where they could also be accessed by others [1], [2]. As described in the poster, the AIG nugget has been modified from its initial form within Angel at PSU for use within Blackboard at Southampton and Bodington Common at Leeds.

3 Case Study

Following discussion between the academic collaborators based on the PSU implementation, initial agreement was reached that the common resource framework should consist of a narrative providing learners with help and instructions on academic integrity, and a formative quiz allowing self-assessment. Decisions on the exact configuration of the resources were then made locally at Southampton and Leeds, with reference to the institutional settings, target audiences, existing local expertise and resources available. The poster traces the original development of the AIG nugget at PSU through take-up, versioning [3] and repurposing firstly at Southampton and then at Leeds. It also highlights the different approaches in handling the dynamic quiz pages and the configuration of the supporting resources.

4 Interim Conclusions

The experience of versioning the AIG nugget across the three universities serves to demonstrate the importance of an identifiable need for learning materials in order to engage academics in the process of re-use and repurposing through DL infrastructures [4]. There are also genuine challenges relating to institutional practices which require even the lowest level materials to be recast before they are approved for use with students under different institutional regulations. There are future challenges in identifying whether the very mechanisms of sharing learning objects through a DL would enable greater convergence of regulations or whether this will remain an insurmountable barrier. This transatlantic, inter-institutional collaboration serves to illustrate many of the challenges which will be faced by any attempt to place general-purpose learning objects within shared digital library repositories. Our evaluation of the use of these resources in academic year 2004/05 is currently underway and the poster will incorporate some preliminary findings from this evaluation. Equally important is identifying the need to author content objects in accordance with learning technology standards [ibid], such as IMS-QTI, at the earliest possible stage to ensure interoperability and to avoid the problem of VLE lock-in.

References

1. Boyle, T.: Design principles for authoring dynamic, reusable learning objects. *Australian Journal of Educational Technology*, 19(1) (2003) 46-58
2. Weller, M. J., Pegler, C. A. and Mason, R. D.: Putting the pieces together: What working with learning objects means for the educator. Presented at ELearn International, 9-12 February, Edinburgh (2003)
3. Thorpe, M., Kubiak, C. and Thorpe, K.: Designing for reuse and versioning. In: Littlejohn, A. (ed.): *Reusing Online Resources: a sustainable approach to e-learning*. Kogan Page, London (2003)
4. Duncan, C.: *Digital Repositories: E-Learning for Everyone*. Presented at ELearn International, 9-12 February, Edinburgh (2003)

Supporting ECDL'05 Using TCeReview

Andreas Pesenhofer¹, Helmut Berger¹, and Andreas Rauber^{1,2}

¹ iSpaces Group, eCommerce Competence Center – ec3,
Donau-City-Straße 1, A-1220 Wien, Austria

² Vienna University of Technology,
Favoritenstraße 9–11/188, A-1040 Wien, Austria

Abstract. Conference Management constitutes a field in Digital Libraries including tasks such as paper to reviewer assignment and session compilation. These tasks depend on the paper to topic assignment. TCeReview addresses the automatic organization of text documents and enhances conventional conference management applications by incorporating a text classification module. This paper presents the results obtained during the empirical evaluation of the TCeReview applied at ECDL'05.

1 Introduction

One task authors have to complete when submitting a paper to a conference is to select a research topic that identifies their submission. Subsequently, this topic is used by conference organizers to determine appropriate reviewers and, in case of acceptance, compile sessions. However, authors might be uncertain about selecting an appropriate topic for the paper. Authors confusion might be even greater when research topics cannot be clearly described in a few words. TCeReview (Text Classification Enhanced Review) addresses this issue by incorporating a text classification module into the conference management application *MyReview* (<http://myreview.lri.fr/>). The classification module was trained with accepted submissions from previous conferences and automatically suggests the most likely topic to the author. TCeReview is currently being evaluated in different conference settings, starting with small and medium sized events up to a challenging medical conference with about 3,000 submissions. In this paper we report in a quasi-recursive manner first results using ECDL'05 as live example.

Data from previous ECDL conferences were downloaded from Springer Online (<http://www.springerlink.com>) in order to build the training set for TCeReview. 311 abstracts related to conference call topics were manually selected and assigned to research topics (cf. Table 1, first column). After preprocessing, the abstracts were indexed with the Rainbow library [1]. 4,141 unique terms were obtained and further reduced to 3,460 terms using Information Gain as feature selection metric. The classification task was carried out by means of a Naïve Bayes classifier [2].

At the first step of submission authors were asked to register their paper and supply meta-data including an abstract. Later, authors completed their

Table 1. Classifier confusion matrix

class name	ID	1	2	3	4	5	6	7	recall
Concepts of Digital Libraries, Concepts of Documents and Metadata	1	1	1	2	3	.	1	1	0.11
System Architectures, Open Archives, Collection Building, Integration and Interoperability	2	1	18	1	0.90
Information Retrieval, Information Organization, Search and Usage	3	1	3	28	6	.	2	.	0.70
User Studies, System Evaluation, Personalization, User Interfaces and User Centered Design	4	.	.	4	22	.	2	1	0.76
Digital Preservation, Web Archiving and Long Term Access	5	1	1	3	.	.	1	1	0.00
Digital Library Applications and Case Studies	6	.	3	1	2	.	14	1	0.67
Multimedia, Mixed Media, Audio, Video, 3D and non-traditional Objects	7	3	1.00
precision		0.25	0.68	0.72	0.67	0.00	0.70	0.43	

submission by uploading their final paper. Based on the abstract TCeReview suggested a research topic for the paper. In case of disagreement the author had the possibility to change the topic. Author’s agreement or disagreement was tracked in order to evaluate the system.

2 Results and Conclusion

While the submission site was open 129 abstracts were received. Note that abstracts being shorter than 100 characters were discarded. We performed an ex-post evaluation of the author’s disagreement with the suggested research topic. The results of the classifier are depicted in the confusion matrix (cf. Table 1). Rows give the class assignments and columns correspond to the prediction of the classifier. The obtained results showed that precision of the classes 2, 3, 4 and 6 is about 70%. These classes were represented by 40 to 67 instances per class in the training set. In case of the class “Concepts of Digital Libraries, Concepts of Documents and Metadata” 25% are correctly classified. The poor performance of this class is attributed to the few training examples (12). Overall, TCeReview achieved an accuracy of 66.67%. This indicates that the system helps the author in assigning the appropriate topic to the submitted abstract.

A similar approach might be taken in order to assign papers to reviewers. Moreover, the system assists in session compilation by applying clustering algorithm on final submissions.

References

1. Andrew McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Internet: <http://www.cs.cmu.edu/~mccallum/bow>, 1996. (last visit: 27.4.2005).
2. Andrew McCallum and Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48. AAAI Press, 1998.

ContentE: Flexible Publication of Digitised Works with METS

José Borbinha, Gilberto Pedrosa, and João Penas

INESC-ID – Instituto de Engenharia de Sistemas e Computadores, Rua Alves Redol 9,
Apartado 13069, 1000-029 Lisboa, Portugal
jlb@ist.utl.pt, gpedrosa@ext.bn.pt, penas@ext.bn.pt

Abstract. This poster addresses the problem of the publication of digitized works. It presents the solution developed at the National Library of Portugal, where nearly one million of images were created in the last year, from a wide range of original genres (manuscripts, maps, posters, books, newspapers, etc.). The solution is based on a tool named ContentE, which supports the creation, import and editing of structural descriptions of works, making it possible to record them in XML using the METS schema. The tool manages also collections of style sheets, making it possible to create multiple publication copies, as XHTML sites. This solution can be used as a standalone tool, with a graphic user interface, or embedded in a web-service, for automatic publishing.

Poster Description

ContentE is a result of the initiative BND¹, promoted by the National Library of Portugal. It is all coded in Java, and is available as open-source. It is a tool to create and edit structural descriptions of digital works, making it possible to save and reuse those results in formal structural metadata schemas, such as METS². Its initial purpose was the publishing of digitised collections, but its success made it a very practical tool to deal with a wide number of MIME types.

The structural descriptions edited in ContentE are defined as indexes, which are made of trees of nodes where each node can be an aggregator or a leaf. Aggregators represent structural concepts such as parts of books, chapters, volumes, sections, etc. A leaf node is a reference to a content file of any MIME format (such as image formats like PNG, GIF, JPEG, TIFF, etc., but also MS-Word, PDF, Postscript, ASCII, RTF, etc.). Aggregators can make reference to leaf elements, and can have also a list of descriptive and rights metadata files, such as MARCXML, Dublin Core, etc. ContentE can import those descriptions from local files, or it can retrieve them on-line from remote web-services.

ContentE manages also collections of style sheets, making it possible to create multiple publication copies of the objects, as XHTML sites, each one with its specific style. That makes it possible, from the same master object, to publish objects with different layouts (menus at the right or left, background colours, etc.), or with

¹ <<http://bnd.bn.pt>>

² <<http://www.loc.gov/standards/mets>>

different contents (showing or not descriptive metadata, showing images at different resolutions for digitised books, etc.)

ContentE can be used as a standalone tool, with a graphic user interface, or embedded in a web-service, for automatic publishing. Figure 1 shows the user interface for the standalone tool in a session to describe the structure of a specific digitized book³. Figure 2 shows two works published using different styles.

ContentE is being used at BND for the publication of one million of images, which is also a very important demonstration of the relevance of METS as simple, practical and flexible structural schema.

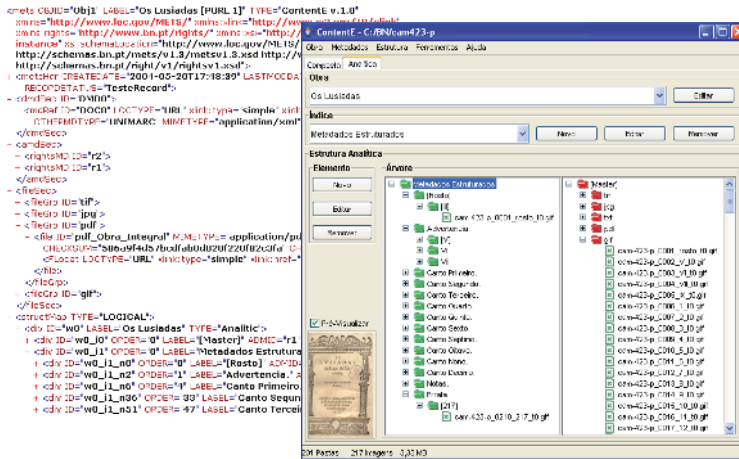


Fig. 1. ContentE as a standalone tool, in the edition of the structure of a digitised book

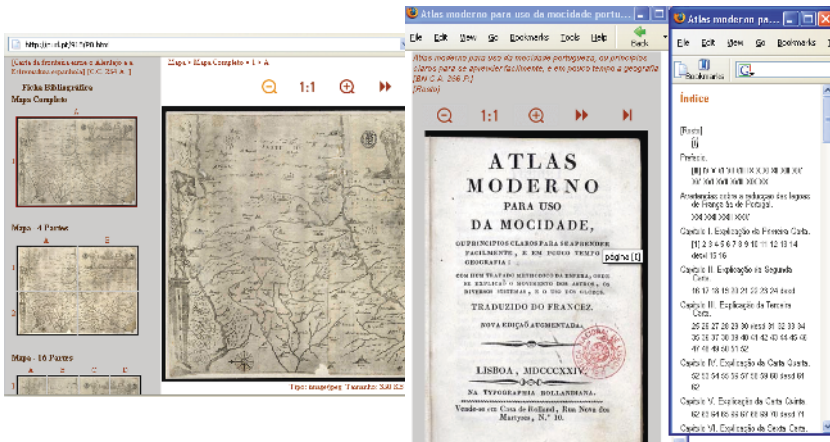


Fig. 2. Examples of two digitised works published by ContentE using very different styles

³Copies of this specific book can be seen at <http://purl.pt/1>

The UNIMARC Metadata Registry

José Borbinha and Hugo Manguinhas

INESC-ID – Instituto de Engenharia de Sistemas e Computadores, Rua Alves Redol 9,
Apartado 13069, 1000-029 Lisboa, Portugal
jlb@ist.utl.pt, mangas@bn.pt

Abstract. This poster describes the first steps in creating a metadata registry for the UNIMARC formats. This registry aims to hold formal descriptions of the structure of the formats, keeping track of their versions, as also the register of the textual descriptions in multiple languages. These structural representations are recorded in XML. This poster gives a special focus to the results already available for the bibliographic format: its on-line textual publication and the automatic validation of bibliographic records.

Description

UNIMARC is a family of metadata schemas with formats for descriptive information, classification, authorities and holdings. The UNIMARC activities are an IFLA Core Activity, the ICABS: IFLA-CDNL Alliance for Bibliographic Standards¹.

In 2004, BN, the National Library of Portugal started an activity to develop a UNIMARC Registry, to support the evolution processes of the UNIMARC formats and provide a reference point for the professionals and organisations using it.

The initial activities of this action were focused on the bibliographic format, and especially in developing a formal description of its structure and grammar in XML (Figure 1, left). The application of that result has been tested and validated in two practical cases: the on-line textual publication of the format, in this moment in Portuguese and English languages² (Figure 2), and the automatic validation of bibliographic records.

The system for the on-line publication of the format provides persistent URN identifiers for the most recent version of the elements of the format (but the URN schema makes it possible to access also to specific versions of those descriptions). In this moment those URN are resolved only for HTML pages, but in the future it'll be possible to return also the same information in a structured XML reply.

For the automatic validation of bibliographic records we developed a standalone tool, named MANGAS (Figure 1, right), which validates the integrity of a UNIMARC record and its information. This tool supports also features for correction of records, and detailed reports for record sets. The engine used in this application is going to be used also in a new on-line service to provide the same features for on-line users, as also embedded in a web-service, for usage by remote systems.

¹ <http://www.ifla.org/VI/7/icabs.htm>

² <http://www.unimarc.info>

The next steps of this activity will be, in the short-term, to revise and finish the formal Registry according to the ISO/IEC 11179³ (formal description of terms, revision of the URN space, etc.), to develop a stable web-service for machine access, add more formats, versions and languages, and formalise the descriptions of changes between versions.

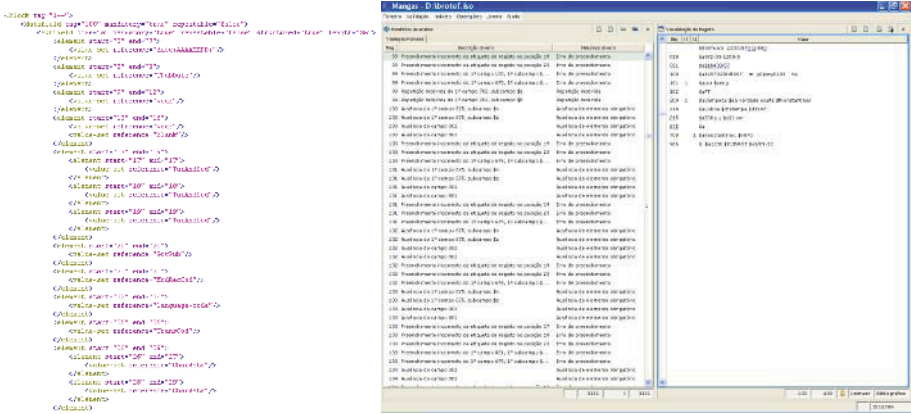


Fig. 1. The formal structure of UNIMARC Bibliographic Format in XML, and its usage in the records validation tool MANGAS

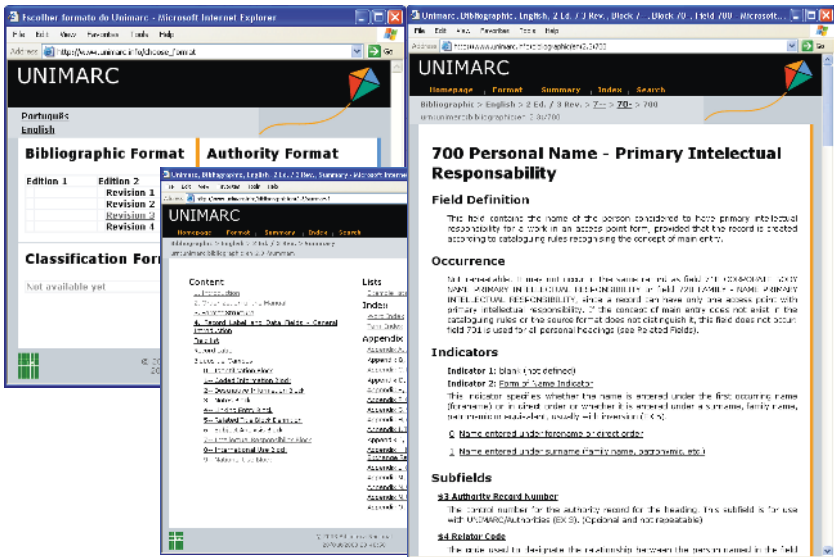


Fig. 2. The automatic publication of the UNIMARC Bibliographic Manual from the UNIMARC Registry

³ <http://metadata-standards.org/11179/>

Developing a Computational Model of “Quality” for Educational Digital Libraries

Tamara Sumner¹, Mary Marlino², and Myra Custard¹

¹ The University of Colorado at Boulder, Boulder CO, USA
{sumner, myra.custard}@colorado.edu

² DLESE Program Center, University Corp. for Atmospheric Research, Boulder CO, USA
marlino@ucar.edu

Abstract. This poster will present the results of a pilot study that examined the efficacy of a computational model to support users in determining quality in educational digital libraries. The subsequent research design of a larger follow-on project will also be presented. It is anticipated that the conceptual and computational models that will be created for scaffolding quality judgments about library resources can be empirically validated, and ultimately integrated into digital library tools and services.

1 Research Problem

Over the past decade, there has been vocal and visible public demand in the United States for improved science education across the national, state, and local levels, and a call for greater access to quality science education for all citizens. In part, this call is being answered by the emergence of operational digital libraries devoted to science education across a range of science, mathematics, engineering and technology disciplines. Two such examples funded by the US National Science Foundation are the National Science Digital Library (www.NSDL.org) and the Digital Library for Earth System Education (www.DLESE.org), which provide the context for our research. Both of these library efforts aim to make accessible an array of “high-quality” collections to serve diverse constituencies of users, from research scientists, to educators, to students, at all educational levels, in both formal and informal settings.

Our experience to date suggests that understanding ‘quality’ per se, and how to develop and manage quality collections is a critical and growing issue as educational digital libraries mature. As such, these library initiatives are devoting significant resources to establish policies and procedures to support developing, accessioning, and curating quality resources and collections.

This poster will present the results of a pilot study of our efforts to develop a computational model to assist library users in making judgments regarding quality in educational digital libraries resources and collections. Our effort builds on prior research in three main areas:

- Conceptual models of expert evaluative criteria, processes, and strategies
- Computational models that approximate expert processes
- Models of expert evaluative knowledge or processes to scaffold evaluative processes in the context of learning and design activities

2 Methods and Techniques

Based on our experiences in our pilot study, we will investigate our research questions with the following methods. We will first study expert collection developers in order to identify key markers of quality. Results from these studies will guide the creation of machine learning training data. An iterative approach of algorithm refinement and testing will be used to identify dimensions that can be reliably identified and classified. We will then develop a data model, informed by the conceptual model, the machine learning results, and best practices for data modeling and data interoperability within library networks. Finally, we will conduct laboratory studies of human experts to analyze the fidelity and utility of the information provided by the computational model for supporting human quality judgments.

We anticipate two primary outcomes from this work: a conceptual model of expert quality evaluation processes and a corresponding computational model. Secondary outcomes include the development and refinement of use cases to guide our own evaluation efforts and to help document the model's applicability and potential uses. Conceptual models are a recognized intellectual mechanism for documenting and sharing knowledge about cognitive processes within the information science and cognitive science disciplines. As such, our conceptual model of expert quality evaluation processes will be an important theoretical contribution to information science. The computational model we will create will provide a concrete implementation and formalization of this conceptual model, one that can be empirically validated and ultimately integrated into digital library tools and services.

This research lays critical groundwork for next generation digital libraries on multiple levels. The resulting computational model can be applied to develop intelligent interfaces and visualization tools to support and scale a wide range of collection development activities. While not the focus of this project, we believe that this model could ultimately be of great benefit for supporting the development of students' information literacy skills.

Author Index

- Abascal, Rocío 496
Adly, Noha 116
Ahuja, Kapil 186
Aihara, Kenro 493
Almpanidis, George 402
Amato, Giuseppe 69
Anand, Sai 479
Ardö, Anders 368
Audenaert, Neal 151
Auld, Dan 487
- Balakireva, Lyudmila 254
Balke, Wolf-Tilo 379
Berger, Helmut 535
Berisha-Bohé, Suela 496
Bhoopalam, K. 489
Bia, Alejandro 519
Bier, Eric A. 162
Blandford, Ann 218
Bollen, Johan 344
Borbinha, José 537, 539
Borges, Marcos R.S. 515
Brase, Jan 128
Brisaboa, Nieves R. 462
Buchanan, George 218
Budulea, Eduard 481
Butterworth, Richard 278
- Canós, José H. 515
Careaga, Doris 151
Carrasco, Rafael C. 81
Castelli, Donatella 477
Chang, Chew Hung 336
Chanod, Jean-Pierre 92
Chen, Jiangping 513
Chen, Su-Shing 500
Chidlovskii, Boris 92
Cole, Timothy W. 290
Colomitchi, Adrian 481
Councill, Isaac G. 438
Cox, Ingemar J. 438
Cunningham, Sally Jo 195, 218
Curzon, Paul 278
Custard, Myra 541
- Davis-Perkins, Veronica 278
Debole, Franca 69
Dejean, Hervé 92
Delcambre, Lois 511
Deng, Jie 151
Diaconescu, Stefan 481
DiBiase, David 533
Diekema, Anne R. 513
Diepenbroek, Michael 128
Di Nunzio, Giorgio Maria 483
Dumitru, Viorel 481
Duval, Erik 323
- Fambon, Olivier 92
Fan, Weiguo 1, 139
Fariña, Antonio 462
Ferro, Nicola 483
Fields, Bob 278
Fill, Karen 533
Finneran, Christina M. 513
Flexer, Arthur 37
Foo, Schubert Shou-Boon 502
Foulonneau, Muriel 290
Fox, Edward 1
Fox, Edward A. 139, 186
Frankmölle, Sven 527
Frew, James 302
Fu, Lin 502
Fuhr, Norbert 414, 527
Fülöp, Csaba 314
Furuta, Richard 151
Fuselier, Jérôme 92
- Gerken, Jens 174
Giles, C. Lee 438
Goh, Dion Hoe-Lian 336, 450, 502
Golub, Koraljka 368
Gómez, Jaime 519
Gonçalves, Marcos André 1, 139
Gonzalo, Julio 529
Good, Lance E. 162
Grün, Christian 174
Gurrin, Cathal 487

- Han, Hui 438
 Harper, David J. 504
 Harrison, Terry L. 509
 Harwell, Sarah C. 513
 Hedberg, John 336
 Heery, Rachel 475
 Hoare, Cathal 507
 Hochstenbach, Patrick 254
 Höfner, Peter 498
 Huang, Michael Bailou 523
 Huhn, Alfons 498
 Hunter, Jane 475

 Idreos, Stratos 25
 Ioannidis, Yannis 477

 Jaballah, Imene 195
 Jacquin, Thierry 92
 Janée, Greg 302
 Jang, Myung-Gil 500
 Janssen, William C. 162, 230
 Jayakanth, Francis 521
 Jetter, Hans-Christian 174
 Jordan, Matthias 414

 Kapidakis, Sarantos 356, 426, 525
 Khan, Haseebulla M. 531
 Khoo, Christopher S.G. 450
 Kießling, Werner 498
 Kim, Hyunki 500
 Kim, Seonho 186
 Kiss, Gergő 314
 Klas, Claus-Peter 414, 527
 Klatt, Manuel 104
 Klerkx, Joris 323
 König, Werner 174
 Kotropoulos, Constantine 402
 Koubarakis, Manolis 25
 Koulouris, Alexandros 356, 525
 Kovács, László 314
 Kriewel, Sascha 527

 Leung, Samuel 533
 Li, Guohui 523
 Liddy, Elizabeth D. 513
 Lim, Ee-Peng 336
 Liu, Xiaoming 254, 509
 Llavador, Manuel 515
 López-Ostenero, Fernando 529
 Lossau, Norbert 475

 Maly, Kurt 489, 531
 Manguinhas, Hugo 539
 Marlino, Mary 541
 McCown, Frank 344, 489
 McFarland, Nathan 509
 Meunier, Jean-Luc 92
 Micsik, András 314
 Minj, Filbert 521
 Monroy, Carlos 151
 Muñoz, Carlos González 81
 Mukkamala, R. 489
 Müller, Uwe 104
 Murthy, Uma 186

 Nagi, Magdy 116
 Najjar, Jehad 323
 Nanba, Hidetsugu 391
 Navarro, Gonzalo 462
 Nejdil, Wolfgang 379
 Nelson, Andy 533
 Nelson, Michael L. 344, 509
 Neve, Giovanna 49
 Nielsen, Marianne Lykke 511
 Nikolaidou, Mara 13

 Okumura, Manabu 391
 Orio, Nicola 49
 Ou, Shiyan 450

 Pampalk, Elias 37
 Paramá, José R. 462
 Pedrosa, Gilberto 537
 Peinado, Víctor 529
 Penas, João 537
 Pesenhofer, Andreas 535
 Petrelli, Daniela 487
 Petricek, Vaclav 438
 Phelps, Thomas A. 266
 Pinon, Jean Marie 496
 Pitas, Ioannis 402
 Popat, Ashok C. 162
 Pyrounakis, George 13

 Raghavan, Ananth 139
 Rajashekar, T.B. 521
 Ramnishath, A. 521
 Rauber, Andreas 535
 Reiterer, Harald 174
 Rimmer, Jon 218
 Rumpler, Béatrice 496

- Sáenz, Rosy 151
Saidis, Kostas 13
Salampasis, Michail 57
Saleh, Iman 116
Sánchez-Villamil, Enrique 81
Schaefer, André 414
Schindler, Uwe 128
Sfakakis, Michalis 426
Shen, Rao 1, 139
Siberski, Wolf 379
Smeaton, Alan 487
Smith, MacKenzie 242
Sorensen, Humphrey 507
Sumner, Tamara 475, 541
Sun, Aixin 336
Sun, Yixing 504

Tait, John 57
Takasu, Atsuhiko 493
Talja, Sanna 207
Tansley, Robert 242
Thaden, Uwe 379
Theng, Yin-Leng 336
Tolle, Timothy 511
Tryfonopoulos, Christos 25

Urbina, Eduardo 151

Vakkari, Pertti 207
Van de Sompel, Herbert 254, 509
van der Hoeven, Jeffrey 485
van Wijngaarden, Hilde 485
Vasile, Sandi 186
Vemuri, Naga Srinivas 139
Verdejo, Felisa 529
Vuorikari, Riina 323

Walker, Julie Harford 242
Warner, Simeon 491
Warwick, Claire 218
Watry, P.B. 266
Watt, Stuart N.K. 504
Weaver, Mathew J. 511
Widmer, Gerhard 37
Wilde, Erik 479
Witten, Ian H. 195
Wright, Michael 475
Wu, Dan 336

Yilmazel, Ozgur 513

Zimmermann, Petra 479
Zong, Wenbo 336
Zubair, M. 489
Zubair, Mohammad 531