

Simon Andrews
Simon Polovina
Richard Hill
Babak Akhgar (Eds.)

LNAI 6828

Conceptual Structures for Discovering Knowledge

19th International Conference
on Conceptual Structures, ICCS 2011
Derby, UK, July 2011, Proceedings

 Springer

Lecture Notes in Artificial Intelligence

6828

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Simon Andrews Simon Polovina
Richard Hill Babak Akhgar (Eds.)

Conceptual Structures for Discovering Knowledge

19th International Conference
on Conceptual Structures, ICCS 2011
Derby, UK, July 25-29, 2011
Proceedings

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Simon Andrews
Simon Polovina
Babak Akhgar
Sheffield Hallam University
153 Arundel St., Sheffield, S1 2NU, UK
E-mail: {s.andrews;s.polovina;b.akhgar}@shu.ac.uk

Richard Hill
University of Derby
Kedleston Road, Derby, DE22 1GB, UK
E-mail: r.hill@derby.ac.uk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-22687-8 e-ISBN 978-3-642-22688-5
DOI 10.1007/978-3-642-22688-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011932224

CR Subject Classification (1998): I.2, H.2, I.5, I.2.7, I.2.4, F.4.3

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2011

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.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the proceedings of the 19th International Conference on Conceptual Structures (ICCS 2011), the latest in a series of annual conferences that have been held in Europe, Asia, Australia, and North America since 1993. Details of these events are available at www.conceptualstructures.org, and www.iccs.info points to the latest conference in this prestigious series. ICCS focuses on the useful representation and analysis of conceptual knowledge with research and business applications. It brings together some of the world's best minds in information technology, arts, humanities, and social science to explore novel ways that information and communication technologies can leverage tangible business or social benefits. This is because conceptual structures (CS) harmonize the creativity of humans with the productivity of computers. CS recognizes that organizations work with concepts; machines like structures.

ICCS advances the theory and practice in connecting the user's conceptual approach to problem solving with the formal structures that computer applications need to bring their productivity to bear. Arising originally out of the work of IBM in conceptual graphs, over the years ICCS has broadened its scope to include a wider range of theories and practices, among them formal concept analysis, description logics, the Semantic Web, the Pragmatic Web, ontologies, multi-agent systems, concept mapping, and more. Accordingly CS represent a family of approaches that builds on the successes of artificial intelligence, business intelligence, computational linguistics, conceptual modelling, information and Web technologies, user modelling, and knowledge management.

The theme for this year's conference was "Conceptual Structures for Discovering Knowledge." More and more data is being captured in electronic format (particularly through the Web and social media) and it is emerging that this data is reaching such a critical mass that it is becoming the most recorded form of the world around us. It now represents our business, economic, artistic, social, and scientific endeavors to such an extent that we require smart applications that can discover the hitherto hidden knowledge that this mass of data is busily capturing. By bringing together the way computers work with the way humans think, CS align the productivity of computer processing with the ingenuity of individuals and organizations in a meaningful digital future.

The ICCS papers that appear in this volume represent the rich variety of CS. Submitted papers were rigorously reviewed anonymously by members of the Program Committee and the Editorial Board who oversaw the process together with the organizers. About 60% of submitted papers deemed relevant to the conference were accepted, plus a few as short papers. There were also three invited papers. As this volume will evidence, it is pleasing that the number of accepted full papers reflects the high quality of submissions that ICCS continues to attract as the conference approaches its 20th anniversary.

In addition to ICCS, there were four workshops at the conference. Three of these workshops' papers appear under their own sections in this volume. Two of these workshops cover CS and knowledge discovery in under-traversed domains and in task-specific information retrieval. The third addresses "CS in Learning, Teaching and Assessment;" a workshop that had its inauguration at last year's ICCS (2010) in Kuching, Malaysia. The papers of the fourth workshop, "The First CUBIST Workshop", appear in their own proceedings. ICCS 2011 represented a key dissemination event for the CUBIST project (www.cubist-project.eu), which is funded by the European Commission under the 7th Framework Programme of ICT, topic 4.3: Intelligent Information Management.

We wish to express our thanks to all the authors of the submitted papers, the speakers, workshop organizers, and the members of the ICCS Editorial Board and Program Committee. We would like to thank Uta Priss, who organized the anonymous reviewers of papers submitted by the ICCS Chairs. We also extend our thanks to the Local Organizing Chair Ashiq Anjum, and to our Sheffield Hallam and CUBIST colleague Constantinos Orphanides for managing the production of the proceedings, ready for the helpful people at Springer to whom we also owe our gratitude.

July 2011

Simon Andrews
Simon Polovina
Richard Hill
Babak Akhgar

Heather D. Pfeiffer	New Mexico State University, USA
Simon Polovina	Sheffield Hallam University, UK
Uta Priss	Edinburgh Napier University, UK
Sebastian Rudolph	University of Karlsruhe, Germany
Henrik Schärfe	Aalborg University, Denmark
John F. Sowa	VivoMind Intelligence, Inc., USA
Gerd Stumme	University of Kassel, Germany
Rudolf Wille	Technische Universität Darmstadt, Germany
Karl Erich Wolff	University of Applied Sciences Darmstadt, Germany

Program Committee

Jean-François Baget	LIRMM-RCR and INRIA Rhône-Alpes, France
Radim Bělohlávek	Palacky University of Olomouc, Czech Republic
Tru Cao	Ho Chi Minh City University of Technology, Vietnam
Peggy Cellier	INSA of Rennes, France
Dan Corbett	DARPA, Washington DC, USA
Juliette Dibie-Barthèlemy	AgroParisTech, France
Pavlin Dobrev	ProSyst Labs EOOD, Bulgaria
Jerome Fortin	Iate, France
Udo Hebisch	Technische Universität Freiberg, Germany
Jan Hladik	SAP Research Dresden, Germany
John Howse	University of Brighton, UK
Adil Kabbaj	INSEA, Morocco
Markus Kröttsch	University of Oxford, UK
Leonard Kwuida	Zurich University of Applied Sciences, Switzerland
Ivan Launders	BT Global Services, UK
Michel Leclère	LIRMM, France
Robert Levinson	UC Santa Cruz, USA
Philippe Martin	Eurécom, France
Boris Motik	University of Oxford, UK
Daniel Oberle	SAP Research Karlsruhe, Germany
Sergei Obiedkov	State University Higher School of Economics, Russia
Jonas Poelmans	Katholieke Universiteit Leuven, Belgium
Anne-Marie Rassinoux	HCUGE, Switzerland
Eric Salvat	IMERIR, France
Ulrik Sandborg-Petersen	Aalborg University, Denmark
Jeffrey Schiffel	The Boeing Company, USA
Iain Stalker	University of Manchester, UK
Martin Watmough	CIBER, UK

Further Reviewers

Peter Chapman
Andrew Fish

Workshop Organizers - CFEUTD

Azita Bahrami	IT Consultation, USA
Ray Hashemi	Armstrong Atlantic State University, USA
Hamid Arabnia	University of Georgia, USA
John Talburt	University of Arkansas at Little Rock, USA

Workshop Organizers - TSIR

Rahat Iqbal	Coventry University, UK
Adam Grzywaczewski	Trinity Expert Systems Limited, UK

Workshop Organizers - CS-LTA

Meena Kharatmal	Homi Bhabha Centre for Science Education, Mumbai, India
G. Nagarjuna	Homi Bhabha Centre for Science Education, Mumbai, India

Sponsoring Institutions

School of Computing and Mathematics, University of Derby, UK
Communication and Computing Research Centre (CCRC) and the Department
of Computing, Sheffield Hallam University, UK

Table of Contents

Invited Papers

Semantic Technologies for Enterprises	1
<i>Frithjof Dau</i>	
Utility and Feasibility of Reasoning beyond Decidability in Semantic Technologies	19
<i>Sebastian Rudolph and Michael Schneider</i>	
Cognitive Architectures for Conceptual Structures.....	35
<i>John F. Sowa</i>	

Accepted Papers

In-Close2, a High Performance Formal Concept Miner	50
<i>Simon Andrews</i>	
A Mapping from Conceptual Graphs to Formal Concept Analysis	63
<i>Simon Andrews and Simon Polovina</i>	
Partial Orders and Logical Concept Analysis to Explore Patterns Extracted by Data Mining	77
<i>Peggy Cellier, Sébastien Ferré, Mireille Ducassé, and Thierry Charnois</i>	
A Buzz and E-Reputation Monitoring Tool for Twitter Based on Galois Lattices.....	91
<i>Etienne Cuvelier and Marie-Aude Aufaure</i>	
Using Generalization of Syntactic Parse Trees for Taxonomy Capture on the Web	104
<i>Boris A. Galitsky, Gábor Dobrocsi, Josep Lluís de la Rosa, and Sergei O. Kuznetsov</i>	
A.N. Prior's Ideas on Tensed Ontology.....	118
<i>David Jakobsen, Peter Øhrstrøm, and Henrik Schärfe</i>	
Crowdsourced Knowledge: Peril and Promise for Conceptual Structures Research	131
<i>Mary Keeler</i>	

Evaluating the Transaction Graph through a Financial Trading Case Study	145
<i>Ivan Launders</i>	
Integration of the Controlled Language ACE to the Amine Platform	159
<i>Mohammed Nasri, Adil Kabbaj, and Karim Bouzoubaa</i>	
Identifying Relations between Medical Concepts by Parsing UMLS® Definitions	173
<i>Ivelina Nikolova and Galia Angelova</i>	
Topicality in Logic-Based Ontologies	187
<i>Chiara Del Vescovo, Bijan Parsia, and Ulrike Sattler</i>	
A Concept Discovery Approach for Fighting Human Trafficking and Forced Prostitution	201
<i>Jonas Poelmans, Paul Elzinga, Guido Dedene, Stijn Viaene, and Sergei O. Kuznetsov</i>	
A Modeling Method and Declarative Language for Temporal Reasoning Based on Fluid Qualities	215
<i>Matei Popovici, Mihnea Muraru, Alexandru Agache, Cristian Giumale, Lorina Negreanu, and Ciprian Dobre</i>	
Expressing Conceptual Graph Queries from Patterns: How to Take into Account the Relations	229
<i>Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez</i>	
Unix Systems Monitoring with FCA	243
<i>Uta Priss</i>	
Supporting Ontology Design through Large-Scale FCA-Based Ontology Restructuring	257
<i>Mohamed Rouane-Hacene, Petko Valtchev, and Roger Nkambou</i>	
Towards a Formalization of Individual Work Execution at Computer Workplaces	270
<i>Benedikt Schmidt, Heiko Paulheim, Todor Stoitsev, and Max Mühlhäuser</i>	
Semi-supervised Learning for Mixed-Type Data via Formal Concept Analysis	284
<i>Mahito Sugiyama and Akihiro Yamamoto</i>	
Short Papers	
Towards Structuring Episodes in Patient History	298
<i>Galia Angelova, Svetla Boytcheva, and Dimitar Tcharaktchiev</i>	

Rigorous, and Informal?	304
<i>David Love</i>	

OpenSEA – Using Common Logic to Provide a Semantic Enterprise Architecture Framework	309
<i>Jeffrey A. Schiffel and Shaun Bridges</i>	

International Workshop on the Concept Formation and Extraction in Under-Traversed Domains

An Android Based Medication Reminder System: A Concept Analysis Approach	315
<i>Ray Hashemi, Les Sears, and Azita Bahrami</i>	

System Decomposition for Temporal Concept Analysis.....	323
<i>David Luper, Caner Kazanci, John Schramski, and Hamid R. Arabnia</i>	

Modeling UAS Swarm System Using Conceptual and Dynamic Architectural Modeling Concepts	331
<i>Hassan Reza and Kirk Ogaard</i>	

Name Extraction and Formal Concept Analysis	339
<i>Kazem Taghva, Russell Beckley, and Jeffrey Coombs</i>	

International Workshop on Task Specific Information Retrieval

Towards the Development of an Integrated Framework for Enhancing Enterprise Search Using Latent Semantic Indexing	346
<i>Obada Alhabashneh, Rahat Iqbal, Nazaraf Shah, Saad Amin, and Anne James</i>	

Trace of Objects to Retrieve Prediction Patterns of Activities in Smart Homes.....	353
<i>Farzad Amirjavid, Abdenour Bouzouane, and Bruno Bouchard</i>	

Distributed Context Aware Collaborative Filtering Approach for Service Selection in Wireless Mesh Networks	357
<i>Neeraj Kumar and Kashif Iqbal</i>	

A Framework for the Evaluation of Adaptive IR Systems through Implicit Recommendation	366
<i>Catherine Mulwa, Seamus Lawless, M. Rami Ghorab, Eileen O'Donnell, Mary Sharp, and Vincent Wade</i>	

MedMatch – Towards Domain Specific Semantic Matching	375
<i>Jetendr Shamdasani, Peter Bloodsworth, Kamran Munir, Hanene Boussi Rahmouni, and Richard McClatchey</i>	

Application Identification of Semantic Web Techniques in KM
Systems 383
Mohammad Reza Shahmoradi and Babak Akhgar

**Conceptual Structures – Learning, Teaching and
Assessment Workshop**

Aligning the Teaching of FCA with Existing Module Learning
Outcomes 394
Simon Andrews

A Proposal for Developing a Primer for Constructing and Analyzing
Conceptual Structures 402
Nagarjuna G. and Meena Kharatmal

Internationalising the Computing Curricula: A Peircian Approach 406
Richard Hill and Dharmendra Shadija

Broadening the Ontological Perspectives in Science Learning:
Implications for Research and Practice in Science Teaching 414
Nancy R. Romance and Michael R. Vitale

Author Index 423

Semantic Technologies for Enterprises

Frithjof Dau

SAP Research Dresden

Abstract. After being mainly a research topic, semantic technologies (ST) have reached an inflection point in the market. This paper discusses the benefits (data integration and federation, agile schema development, semantic and collaborative / social computing search capabilities) and costs (namely technical, modeling, measuring and educational challenges) of Semantic Technologies with respect to their utilization in enterprises.

Keywords: Semantic Technologies, Semantic Web, Enterprise Applications.

1 Introduction

In its Spring 2009 Technology Forecast [1], PwC (PricewaterhouseCoopers) predicts that “during the next three to five years, we [PwC] forecast a transformation of the enterprise data management function driven by explicit engagement with data semantics.” A recent (spring 2010) report [2] in which 50 high-level decision makers have been interviewed states that “the next generation of IT will be structured around unified information management, Enterprise-level, semantically aware search capabilities, and intelligent collaboration environments - all delivered through dynamic, personalized interfaces that are aware of context”. Taking these two quotations and their sources into account, there is a clear indication that after being mainly a research topic in Academia, now semantic technologies (ST) have reached an inflection point in the market. This paper will discuss the pros and cons of ST with respect to their utilization in enterprises.

The herein presented discussions and insights particularly stem from the author’s experience in the research project Aletheia, lead by SAP, where he has been responsible for the ST layer of Aletheia. The next paragraph summarizes Aletheia and is taken from an SAP-internal whitepaper [3].

The Aletheia research project investigated how current semantic technologies can be applied in enterprise environments to semantically integrate information from heterogeneous data sources and provide unified information access to end users. Often, related product information is spread across different **heterogeneous data sources** (e.g. product information in a database is related to a PDF manual for that product or an entry on the producing company in a spreadsheet). **Semantic integration** in this context essentially means transforming the information into a graph model of typed nodes (e.g. for products, companies) and typed edges (e.g. for the relationship “company-produces-product”). **Providing unified access** means, letting users in search, explore, visualize and augment the information as if it was from one single integrated system. The user interface can profit from the semantic

relationships of the integrated graph to support the user’s search as naturally and intelligently¹ as possible. **Semantic technologies** investigated were light-weight graph models (e.g. RDF), ontologies for capturing aspects of the information that can be reasoned with (e.g. RDFS, OWL, F-logic), as well as text analysis technology for detecting content in unstructured text on a higher level of meaning (e.g. named entity recognition).

Mentioning Aletheia is important for two reasons. First, there are quite a number of whitepapers, e.g. from companies or research institutions specialized in ST, which make (sometimes) bold claims about the benefits of ST for enterprises without further substantiating them. This paper targets at evidencing or exemplifying pros and cons of ST based on Aletheia. Second, the author’s experiences in Aletheia, particularly Aletheia’s focus on information integration, have of course shaped his understanding of ST. Thus the discussion in this paper is quite subjective and elides some aspects of ST, e.g. using ST for service description and consumption, which other people might find very relevant. Having said this, this paper does not claim to provide a complete and comprehensive list of the pros and cons of ST: Instead, it should be considered as a subjective compilation of considerations, as “food for thought”, so-to-speak.

The paper is organized as follows. First a (again subjective) definition of ST is provided. After this, one section discusses the benefits and the (not insignificant) costs of ST in enterprise settings are discussed. This is followed by a section dedicated to the ICCS community, before we come to the conclusion.

2 What Are Semantic Technologies?

Different communities have different notions of “Semantics” or “Semantic Technologies”. Different communities have a different understanding. For example, people from the NLP (natural language processing) might think of thesauri and taxonomies, database experts might have deductive databases in mind, and software engineers think of UML, the object-oriented paradigm, or model-driven architectures. For this reason, we have to clarify first our understanding of ST.

2.1 Core Semantic Technologies

Under core ST we understand technologies like

- Ontology languages, like RDFS, OWL, or FLogic
- Ontology Editors like Protégé or OntoStudio
- Triple stores and semantic repositories, like OWLIM
- Semantic Middleware, like OntoBroker
- Semantic Frameworks, like Sesame
- Reasoners, like pellet

It should be noted that we do not restrict ourselves to Semantic Web technologies, but include for example FLogic as language and the corresponding applications from Ontoprise like OntoStudio and OntoBroker as well.

2.2 Enablers for Semantic Technologies

The vast majority of information processed by ST is not created from scratch. This applies both to the ST schema (aka ontologies) as well as to the data. Instead, the schema and data in semantic repositories is often based on existing data. Thus an ecosystem of methods and tools is needed which turns existing data or existing documents into semantical information. Such methods and tools can be considered as key enablers for ST.¹

There are first of all approaches which are so-to-speak directly connected to core ST. Prominent examples are approaches which map relational databases to RDF. A W3c Incubator Group² has published in 2009 an overview over such tools in [4]; a very recent report has been published by the research project LOD2 (see [6]). The incubator group has meanwhile turned into a working group which aims to “standardize a language for mapping relational data and relational database schemas into RDF and OWL, tentatively called the RDB2RDF Mapping Language, R2RML”, which evidences the importance of these approaches.

There are moreover approaches which have not developed with a dedicated support of core ST into mind, but which are of outstanding importance for ST, namely text mining, information extraction (IE) and NLP approaches. Already in 2006, Timo Kouwenhoven named the following applications:³

- information and meaning extraction,
- autorecognition of topics and concepts, and
- categorization.

It is in the nature of IE and NLP approaches that they do not work with 100% accuracy. The success of ST in the long run will (partly) depend on the maturity and accuracy of these tools.

2.3 Semantic Web vs. Semantic Enterprise

It has to be stressed that ST for enterprises is not the same as the Semantic Web, or Semantic Web technologies simply put in place in enterprises. To name (and sometimes overstress) some differences:

- **Data:** The data in the web is mainly unstructured data (like txt, doc, pdfs) and semistructured data (like html pages or xml files), whereas in enterprises, besides unstructured data, structured data from databases (e.g. ERP systems) is of high importance.
- **Domain:** The web domain is topic-wise unrestricted, whereas the domain for a given enterprise is restricted to the enterprise business. In the enterprise, sometimes existing business vocabulary can (and should) be reused for ST applications. In the web, we have no unique name assumption and the open

¹ This point of view is disputable: The herein mentioned technologies are sometimes understood as genuine ST, e.g. in [5]

² <http://www.w3.org/2005/Incubator/rdb2rdf/>

³ See http://www.timokouwenhoven.nl/2006_02_01_archive.html.

world assumption, whereas in enterprises, entities and documents should have only one identifier (thus the unique name assumption can be assumed), and the closed world assumption holds.⁴

- **User:** In contrast to the web, users in enterprises have specific well-known roles and work in specific well-known contexts. Depending on role and context, the user-access to data and information is controlled.
- **Governance:** Content-wise, the web is not governed, whereas in enterprises, authorities can govern the vocabularies, content, or the development of ST applications as such.

3 Benefits of Semantic Technologies

ST is said to have various benefits in the context of enterprises. Of course, different authors and different people name different lists of benefits, but some existing or envisioned benefits are frequently reoccurring in different sources. First off all, *data integration* is identified as a key benefit, e.g. by [1, 2, 7, 8]. *Agile schema development* is similarly often mentioned [1, 2, 7, 8]. The first two benefits mainly concern the technical backend of enterprise information systems. For users, *semantic search capabilities* is an often named benefit of STs [2, 7, 8]. Finally, though not directly a feature of ST, *collaborative / social computing* is often brought up as key feature or enabler of ST [2, 7].

In the following, we will elaborate on these four benefits in more detail.

3.1 Data Integration and Federation

Its the integration, stupid!“ We find this nice quotation in [9], a work which starts with an analysis of the overall enterprise software market (\$222.6 billion in 2009 according to Gartner), expresses that ERP (Enterprise Resource Planning) “is still what pays the bills” (ERP has a \$67 billion share of the enterprise software market), and examines that “enterprises are all about integration”. They are not alone with this estimation, for [1, 2, 7] data integration is a key asset of ST as well. So we have to dive deeper into the problem, investigate why existing solutions fall short w.r.t. integration, and discuss what ST has to offer.

The need for data integration and federation is certainly not new to enterprises, even if we look only at enterprise internal, structured data. A common problem for enterprises is the independent development of solutions for the different constituencies, which lead to data being spread across different databases. On the one hand, data is often stored in different formats in different databases, on the other hand, different departments often have a different understanding of the meaning, e.g. semantics, of the stored data. Of course, there exists approaches to cope with this problem, e.g. Master Data Management (MDM) systems which attempt to tame the

⁴ As with all points in this list, contrasting the WEB OWA and enterprise CWA is disputable. An example for a different point of view can be found in Bergman’s “seven pillars of the open semantic enterprise/”, where he argues for the “open world mindset”. See <http://www.mkbergman.com/859/seven-pillars-of-the-open-semantic-enterprise/>

diversity of data formats (for a short discussion on the shortcomings of MDM systems compared to ST see for example [10]), or Data Warehouses (DW) persist data federated from different databases based on a unified view on the federated data (which is gathered with the well-known ETL-process, including data cleansing and fusion techniques).

Anyhow, looking only at integrating data from different enterprise-internal databases is certainly not sufficient. Besides relational data, other structured data formats have to be taken into account as well, like xml data, Excel files, CAx files, etc. More importantly, more and more valuable information assets are stored in unstructured formats like (the text of) office documents, emails, or (enterprise-internal) forums and blogs. In fact, the ratio of unstructured data amongst all data is estimated to be 80% to 85%, leaving structured data far behind in second place. MDM systems and DWs cannot deal with these kinds of information. With ST, in combination with enabling technologies like text analysis, information extraction and natural language processing, it is possible to integrate such information sources as well.

The need for data federation does not stop at the borders of enterprises: public and governmental data sources become increasingly important. Initiatives like Linked Open Data [11] foster the availability of public datasets with an impressive growth rate in the last 5 years. Even governments encourage (e.g. UK⁵ and US⁶) the use and re-use of *their* data-sets as Linked Data⁷.

The central goal of open data protocols like Linked Data, OData⁸ (Microsoft) and SAP Data Protocol⁹ (SAP) is to avoid data silos and make data accessible over the web. Common to all of them is the use of URIs to name things and to provide metadata along with the data itself. While OData and the SAP Data Protocol, which builds on OData by adding a business relevant view on it, favor the relational data model and apply a schema first approach, Linked Data is better suited for the graph data models and supports the schema later approach. The tool support for OData is superior today, while on the other side Linked Data supports semantic reasoning with its web-query language SPARQL.

To summarize the discussion so far: With ST, it is possible to federate data from all relevant sources, independent of its format (databases, XML, excel, CAx, text, etc) and location (internal or external). Federating data is more than just gathering data from different sources and persisting¹⁰ it in one central repository. Instead, the mutual relationships and connections between the different data snippets have to be embraced. With the graph-based information models of ST, it is not only feasible to provide an appropriate data model for federated information in which those

⁵ <http://data.gov.uk/>

⁶ <http://www.data.gov/>

⁷ <http://www.w3.org/wiki/LinkedData>, <http://www.w3.org/DesignIssues/LinkedData.html>

⁸ <http://www.odata.org/>

⁹ <http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/00b3d41b-3aae-2d10-0d95-84510071fbb8?QuickLink=index&overridelayout=true>

¹⁰ To avoid too detailed technical discussions, the question which data has to be materialized in a central repository and which data can be retrieved on the fly from the original data sources is deliberately ignored.

relationships are made explicit. Using the reasoning facilities of ST, it is moreover possible to derive new information from the federated data which is not explicitly stored in any of the sources and which might even be obtained by combining facts from *different* data sources.

In Aetheia, indeed information from different sources like databases, xml-files, excel-sheets and text-documents is federated into a graph-based model, and it is even evidenced in Aletheia that that information is presented to the user which can only be obtained by both reasoning on facts from different data sources.

3.2 Agile Schema Development

Schemata for data and information are usually not very stable: They may evolve over time, the notion of entities and relationships change or are extended, new types of entities or relationships have to be added, whereas other types or relationships might become obsolete and thus are dropped from the enterprise information model.

Enterprises have to cope with the following problems: First off all, a too high fraction of the enterprise data models and business logic is still hard coded in the applications. For this reason, it is error prone and costly to employ changes in the model or business logic. For databases, the situation might look different, as data models and even business rules can be captured by the data model of the databases. But the rise of databases dates back to those times where the waterfall model was prominent in software development, and this is still reflected by the design and execution of relational databases. That is, when a database system is set up, first the conceptual schema of the database is to be developed, which is then translated into the relational model. Only after the relational model has been set up, it is possible to fill the database with data. Even if data models are not hard coded, changes in the data model are costly.

With ST, the situation is different. First of all, due to the high expressiveness of semantic models (i.e., ontologies), it is possible to capture a relatively high amount of the enterprise information model and business logics in the semantical model, which leads to a clearer separation of the application and knowledge model. Secondly, it is not necessary to develop a schema first: It is instead possible to store data in a semantical persistency layer (e.g., a triple store) and add later the corresponding schema information. Moreover, changes in the data model are, compared to relational databases, much easier and can be conducted at runtime. Assuming a smart user interface, this could even be done by end users; the data model of applications can even be extended at runtime by end user with new entities and relationships without breaking the application or requiring it to be re-developed.

The separation of application and business logic lowers the TCO (total cost of ownership) of applications to a large extent, suits better the state-of-the-art agile paradigm of software development and is assessed to be a key benefit by many professionals (see [1, 2, 7, 8]).

In Aletheia, filling the repository with federated data and the development of the ontologies has been carried out in parallel. Indeed, the ontologies have sometimes been adapted after (informal) evaluations of Aletheia's behavior in the frontend, which evidences the benefits of the agile schema development. Moreover, the separation of application and business logic can be shown with Aletheia as well: For

the two main use cases, namely the use cases from the partners ABB and BMW, the same Aletheia application is used. In this application, it is possible at run-time to switch between the underlying models for ABB and BMW.

3.3 Semantic Search Capabilities

When it comes to the interaction between users and applications, ST have some core benefits as well. First of all, for accessing information, coherent semantic models can be developed which are especially designed for human understanding (e.g. domain- or business-ontologies), and concepts in these models are mapped to the underlying data sources in a manner transparent to the end users. Thus the heterogeneity and complex technical models and the gap between IT and business is hidden. These models can particularly cover the business terminology of the end users, including synonyms (i.e. different terms which denote the same concept) and homonyms (i.e. terms which denote different concepts). (Syntactically) dealing with synonyms is not very complicated (here it is essentially sufficient to maintain lists of synonyms and taking them in use queries into account), dealing with homonyms is more challenging. Homonyms must be resolved. This can either be done by smart algorithms which do that automatically (i.e., in a search for “bank credit mortgage”, an algorithm could guess that “bank” refers to financial institutions and not to seating-accommodations) or manually when the user enters search terms. Aletheia deals with homonyms by an autocomplete functionality in its search box (see Fig 1).

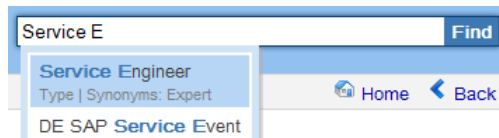


Fig. 1. Autocomplete in Aletheia

“Semantic search” might stand for searching the information space in an explorative manner, or searching *for* specific pieces of information. When it comes to exploring the information space, some user interaction paradigms are quite natural for ST. First of all, the well-known faceted search approach [12] is self-evident. Facets can directly be generated from concept hierarchy of the underlying semantic model. Moreover, as semantic models usually capture relationships between different types of entities as well, a semantically enabled faceted search can allow for navigating along these relationships. Secondly, as semantic data models are usually graphs, graph-based visualizations are similarly natural to employ.¹¹ The essential idea of such visualizations is to display some entities of the information space as nodes of a graph, and displaying the relationships between these entities as (unlabeled or labeled) edges between the corresponding nodes. In such visualizations, different

¹¹ On the web are several examples of graph-based visualizations, e.g. google image swirl (<http://image-swirl.googlelabs.com/>), rel-finder (<http://relfinder.dbpedia.org/relfinder.html>), or Microsoft academic search (<http://academic.research.microsoft.com/>).

means to interactively explore the information space can be implemented. For example, the graph can be extended, nodes can be filtered out, subgraphs of interest can be highlighted, etc. Quite interesting is on the other hand to explore for given entities the path between them in the repository.

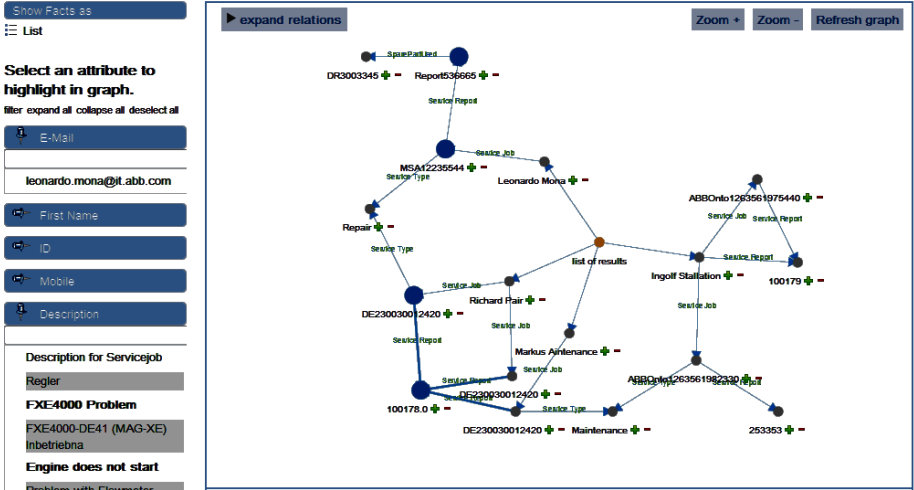


Fig. 2. Aletheia graph-based visualization

Besides (more or less) new user interaction paradigms, it is well accepted that a “classical” search box is a key feature requested by users. We have already argued semantic search can embrace synonyms and homonyms, which renders semantic search superior to standard keyword searches. If moreover the structure of the semantic model (e.g. concept hierarchies, relations between entities in the ontology) is taken into account, smart search algorithms can try to understand what the user means with entered text search queries and can therefore provide better search results. Moreover, search results can be personalized, e.g. based on past search or the context of the user.

An example for a smart search is the semantic search functionality of Aletheia. In Fig 3, a user has entered "Chemical, 800, configured" (and during his input, the search terms “Chemical” and “configured” have been disambigued). This search does not search for entities in the search space which are labeled with all three search strings, but for possibly different entities which are labeled with some of the search strings and which are meaningful connected. Indeed, in Fig 3, one search result is provided which is a specific configuration for a product whose description contains “800” (e.g. AC800M) and which belongs to the branch “Chemical”. To understand why search results are found, Aletheia offers both a textual explanation and a graph-based visualization. Both are shown in in Fig 3.

Initial query and first two search results:

Textual explanation of search result:

Graphical explanation of search result:

Fig. 3. Semantic search in Aletheia

In the long run, the goal is that ST will make the shift from search engines to answer engines by providing what users mean instead of what users say.

3.4 Collaborative/Social Computing

For enterprises, information is a valuable asset, and the benefits of ST discussed so far aim at getting the best out of this asset by integrating all information sources and providing a single point of entry to all information with sophisticated search capabilities. There is anyhow another asset of enterprises which has to be taken into account as well: The enterprises employees and their knowledge. Networking and knowledge exchange between people are becoming increasingly important, thus ST should not only support an integrated access to the data, but attempt to provide an integration of people as well. In the last decade a variety of social network tools has been developed, both in the web sphere (for example, social networking platforms) as well as enterprise internal collaboration platforms (Enterprise 2.0 tools and platforms like semantic media wikis used within companies). So it comes to no surprise that decision makers assess the support of collaboration very important: According to [2], “interoperability is the top priority for semantics; searching/linking information and

collaboration rank next in importance – all are top priority for more than half the companies”, and the report states that “as semantically enabled applications come into the Enterprise mainstream, they will bring the integration and interoperability required for next-generation systems, as well as the usability and collaborative features of social computing (“Web 2.0”).”. In fact, in November 2009 Gartner estimated the “content, communications and collaboration (CCC) market” revenue to be at \$2.6 billion in 2009¹², and in [15] Gartner states that “a major advance in the Semantic Web, the one that has pushed it along on the Hype Cycle, has been the explosion of social networking and social tagging with sites such as Facebook, YouTube, MySpace, Flickr, Wikipedia and Twitter.”

It has anyhow to be observed that to date, semantically supported collaboration tools (like semantic media wikis) are rare and the CCC market is hardly embracing ST. From an enterprise-internal view, enterprise 2.0 tools cover corporate blogging, intra-enterprise social networking tools, corporate wikis, etc. The danger is that companies might implement various mutually independent enterprise 2.0 services, which would cause information about some objects of interest is scattered over the network of the enterprise. How can ST help here? If the content of enterprise 2.0 tools, like people, objects of interests, content, comments, tags are described by agreed-upon semantics, then enterprise 2.0 tools can better interoperate. This is currently a matter of research.¹³ In the web sphere, one expects the convergence of the Web 2.0 on the one hand and semantic web on the other hand to the next web generation called web 3.0. A similar convergence is needed in the enterprise realm for applications which provide a unified view on all enterprise information on the one hand and for applications which support the collaboration amongst employees.

To summarize: in contrast to the ST benefits discussed so far, a semantic approach to collaborative/social computing is still in its infancy and has to to be considered a *prospective* benefit of ST. Promising research on semantically supported collaboration tools is currently conducted, and collaboration tools are likely to be a key enabler for ST (see section 1.3 in [2]).

4 Costs of Semantic Technologies

Semantic technologies do not come for free, there are challenges and cost factors to consider. In the following, the costs and challenges of ST are discussed. This section is based on the findings gained from the research project Aletheia, and a book chapter [13] from Oberle et al which summarizes challenges in adopting ST for software engineering.

[13] names six challenges in adopting ST, namely

- “Technical Integration”, which includes scalability and performance issues of ST applications as well as the maturity level of ST tools
- “Technical Integration $\times n$ ”, which discusses problems if different ST tools are chosen for different use cases

¹² <http://www.gartner.com/it/page.jsp?id=1223818>

¹³ See for example the SIOC-project, <http://sioc-project.org/>

- “Modeling Depends on Use Case”, which scrutinizes the cost of modeling ontologies
- “Cost-Benefit Ratio”, which shortly investigates the TCO (total cost of ownership) for employing ST
- “How to Measure the Benefits?”, which discusses problems to measure the benefits of ST, and
- “Education”, which takes the costs for training developers and users into ST into account.

Though essentially addressing cost of ST for software engineering, the findings of [13] apply to other domains as well. The diagram of Fig 4 is taken from [13] and provides an estimation in how specific the above mentioned challenges to software engineering are.

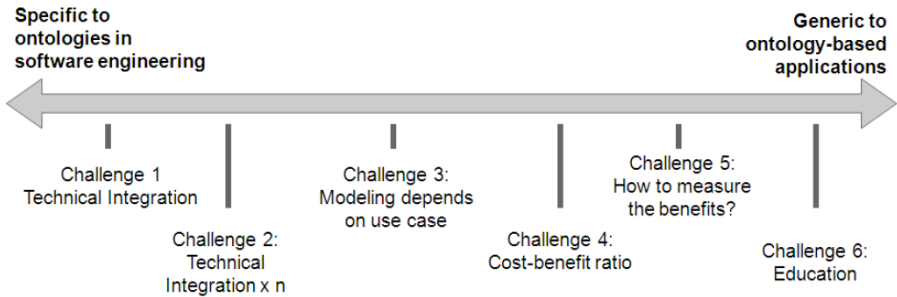


Fig. 4. Challenges of ST for software engineering, according to [13]

Essential findings from Aletheia concerning the costs of ST are:

- Performance and scalability issues of semantic middleware,
- complexity of additional technology stack, including training costs for users and maintenance cost, and
- manual operation effort of the ontologies used in Aletheia.

This findings are closely related to the technical integration and modeling challenges from [13].

In the following, we dive deeper into the challenges.

4.1 Technical Challenges

There is a variety of tools for ST available, but most of them are outcome of academic research (e.g. scientific work by PhD students, academic groups or research projects) and thus lacking documentation or ongoing maintenance and further development. For this reason, most of these tools are not mature enough to be used within an enterprise setting and cannot be taken into account when ST for enterprises are discussed.

Of course, this does not apply to all tools and frameworks from Academia (e.g. Protégé has achieved a enterprise-suitable maturity), and there are moreover

applications from professional vendors, being it dedicated ST vendors like Ontotext, Ontoprise or Franz Inc. or being it large-scale like Oracle or IBM. But even dealing with mature tools has drawbacks.

Two prominent problems are scalability and performance: ST tools still are a magnitude behind relational databases. Moreover, due to the complexity of the semantic languages (like RDFS, OWL, F-Logic) and the corresponding reasoning facilities¹⁴, it might even happen that the tools never completely catch up. These problems are both mentioned in [13] and experienced in Aletheia.

Moreover, the integration of ST into the IT landscape of an enterprise is challenging. To quote [13]: “Technical integration means the required technology needs to be embedded in the existing landscape of the adopting enterprise. Adaptors have to be written to legacy code and databases, versions of programming languages might have to be switched, specific software components might have to be replaced because of license incompatibilities, etc. The challenge typically increases with the size of the legacy code and size of the enterprise's portfolio.”

The situation might become worse if different use cases in an enterprise are taken into account. As discussed in [13], different use cases, which often target different beneficiaries, might use the use of different ontology languages, editors, stores, and reasoners. Obviously, using different ST tools for different use cases increases the complexity of integrating these tools into the IT landscape. As stated in [13], this probably yields to “the challenge of technical integration might have to be faced $\times n$ ”.

Besides these performance/scalability issues and the challenges when it comes to integration, [13] discusses furthermore the challenges

- whether enterprises which want to embrace ST should build ST tools on their own or buy them from third-party vendors, and
- how ontologies are updated in the enterprise IT landscape, which usually offers “a transport system for dealing with updates”.

4.2 Modelling Challenges

Implementing ST in an enterprise use case requires the modeling of an ST schema, i.e., an ontology. Automatic creation of high-quality ontologies is still out of realm, thus ontologies usually have to be manually designed, which is costly and thus increases the TCO (total cost of ownership) when ST are set into place. Anyway, a use case partner in Aletheia understood this effort as an investment beyond Aletheia for their company, as “the ontology can be reused across different software systems and helps the company to maintain a consistent view on their data assets” [AletheiaWP].

Apart from the costs, modeling ontologies is technically challenging as well. First of all, in contrast to the ubiquitous relational model in relational databases, there is a variety of different ontology languages (like RDFS, several OWL profiles tailored for different purposes, or -not being a semantic web language- FLogic) to choose from. Secondly, there is still a lack of mature and comprehensive CASE-tools for modeling

¹⁴ The discussion on the (depending on the languages, sometimes huge) different computational complexities of different languages is outside the scope of this paper and hence deliberately neglected in this discussion.

ontologies. Finally, there is no standard methodology for ontology modeling. A number of methodologies have been proposed, like “Ontology 101” from McGuinness, the method from Uschold and King, the method from Grüninger and Fox, On-To-Knowledge, the Cyc method, SENSUS, KACTUS, TOVE, METHONTOLOGY, etc. The sheer number of methodologies reveals that this is still work in progress, and no methodology has become accepted as standard methodology.

4.3 Measuring Challenges

The pros and cons of ST are (like in this paper) discussed in a qualitative manner. It is anyhow desirable to *quantitatively* measure the benefits and costs. There are different high-level dimensions which can serve as a basis for evaluating ST:

- Technical dimensions like performance and scalability
- User-centric dimensions like the effectiveness of semantic models for users (e.g., compared to relational models)
- Cost-centric dimensions like the TCO for employing ST

For measuring technical dimensions, particularly evaluation RDF stores, a number of benchmarks have been developed. Well known are the Lehigh University Benchmark (LUBM), which been extended to the University Ontology Benchmark (UOBM) for targeting OWL Lite and OWL DL, and the Berlin SPARQL Benchmark (BSBM).¹⁵ Measuring ST technologies is anyhow inherently complicated, as a huge variety of factors which impact the performance or scalability have to be taken into account. Basic measurements are of course loading and retrieval times of triples in RDF stores, which are impacted not only by the number of triples, but by the underlying schema as well, which might trigger several costly reasoning steps, including the recursive application of rules (e.g. to compute the transitive closure of a relation). Some benchmarks cover mapping of relational data to RDF as well.

So it comes as no surprise that there is indeed a variety of benchmarks, and no benchmark has become accepted as the standard for ST. This contrasts the situation of relational databases, where 1988 the Transaction Processing Performance Council (TPC), being a non-profit consortium of different IT enterprises, has been founded, which defines benchmarks that have become the de-facto standard for databases.

Benefits for users are less concrete, thus evaluating the benefits of ST for users is harder. Indeed, [13] states out that “the ontology and Semantic Web community has been struggling to evaluate their contributions accordingly. Indeed, one hardly finds scientific methods or measures to prove the benefits”, but they point out that “other communities share similar struggles”. Of course, there are of course quite a number of user evaluations of applications where ST are used, but the problem is the lack of comparisons of these tools to (corresponding) solutions where ST have not been employed. It can be argued that (carefully crafted) semantic models are closer to human model of the given universe of discourse (from a general design point of view, D. Norman argues in [16] for “proper conceptual models”). Apart from cognitive reasons, this argument is even be witnessed by the success of the leading BI company Business Objects (now a part of SAP): the supremacy of Business Objects is based on

¹⁵ More benchmarks can be found at <http://www.w3.org/wiki/RdfStoreBenchmarking>

their patented invention of their so-called “semantic layer”, which essentially provides a meaningful, business-user-oriented vocabulary of some domain, which is transparently mapped to SQL queries on relational databases. But still, this argument is of qualitative nature: To the best of the author’s knowledge, there are no quantitative evaluations which substantiate the claim that ST are from a user’s perspective superior to relational databases.

After discussing the challenges of evaluating the technical and the use-centric dimensions of ST, it remains to elaborate on the cost-centric dimensions. This dimension is usually measured by the total cost of ownership (TCO). In [13], the following formalization of the TCO is used:

$$TCO \sim TCO \text{ drivers} \times \# \text{stacks in the IT landscape} \times \text{Integration} \times \# \text{of technologies}$$

TCO drivers might include costs for acquiring ST experts or training users in ST, modeling costs, maintenance, and the like. The number of (existing) stacks in the landscape can be explained with the SAP landscape, which features an ABAP and a Java stack. Finally, the number of technologies refers back to the problem of “technical integration $\times n$ ”: If more than one ST editor, store or reasoner is to be used, the factor will increase. But measuring the TCO is not sufficient. Of course, ST can save money as well, compared to other technologies. From an enterprise point of view, a “business case” which captures the rationales for employing ST in an enterprise is needed. As listed in [13], a decent business case must argue for ST

- under consideration of the cost-benefit ratio
- including a deployment plan of available (human) resources
- defining quantifiable success criteria
- proposing an exit strategy
- concerning the business capabilities and impact
- specifying the required investment
- including a project plan
- in an adaptable way, meaning the proposal can be tailored to size and risk.

4.4 Educational Challenges

When ST-based applications are implemented, experts in ST are needed. Being an ST expert requires knowledge in a broad spectrum of topics, e.g.

- knowledge about different ontology languages and their respective capabilities and shortcomings. This particularly includes knowledge about the logical background of (heavy-weight) ontology languages in order to understand the reasoning techniques and capabilities, which often hard for people who lack training in mathematical logic.
- Knowledge about ontology engineering methods and methods, including knowledge about existing ontologies, approaches for re-using ontologies, and methodologies for ontology engineering
- knowledge about existing tools like editors, stores, and reasoners

Thus it takes arguably some effort to become an ST expert.

If an enterprise lacks such experts, either existing employees have to be trained in ST, or ST experts have to be acquired. Even if existing employees are willing to become familiar with ST, teaching them will create considerable training costs. The conclusion from observation can be found in [13]: "Usually, the training costs are very high and managers are not willing to expend them unless there is a compelling business case."

Of course, acquiring new experts instead of training existing employees raises costs as well. But compared to the ubiquitous relational databases, ST are still a quite new technology and neither established in academia nor industry. Thus it will not only be costly to employ ST experts, it will be harder to find them compared to experts in established technologies like relational databases. Again the conclusion can be found in [13]: "If there is no convincing business case, an enterprise might decide to realize the use case with conventional technologies, i.e., technologies where there is expertise readily available in the company."

Expenses are not limited to application developers: they are likely to occur for users as well. As discussed in the last section, ST have benefits both in the back- and in the frontend of applications. For the frontend, semantic search facilities have been named. It should be anyhow noted that only in the ideal case, such new capabilities in the frontend are that user-friendly and easy to understand that no educational costs for users have to be taken into account. Such educational costs do not necessarily refer to training courses; self-education (e.g. E-learning) raises costs as well.

5 Buy-In for the Conceptual Structures Community

It is the author's belief that the conceptual structures (CS) community exhibits significant knowledge for embracing ST, such as theoretical and philosophical background of ST, as well as practical knowledge about

- FCA (both theoretical foundations and practical applications)
- different graph-based knowledge representation and reasoning (like existential graphs, RDF, conceptual graphs in different forms –simple, with rules, based on different kind of logics, with different levels of negation and context, etc-)
- ontologies (languages, background, modeling)

This knowledge is evidenced both by a significant foundational contributions in terms of scientific papers and books ([17, 18] as well as several, sometimes quite powerful and mature, applications for FCA and CG (e.g. ToscanaJ¹⁶ for FCA, and Amine¹⁷ and Cogitant¹⁸ for CGs).

It is anyhow the author's opinion that the community has not such a significant impact in the field as it might deserve. Other communities do better in this respect. An example is the Semantic Web community, which started later¹⁹ than the CS

¹⁶ <http://toscanaj.sourceforge.net/>

¹⁷ <http://amine-platform.sourceforge.net/>

¹⁸ <http://cogitant.sourceforge.net/>

¹⁹ The first Semantic Web Working Symposium has been held in Stanford in parallel to 9th International Conference on Conceptual Structures.

community, but to this end, it is obviously way more prominent. Besides the amount of scientific work coming from this community, two other aspects are worthwhile mentioning:

- **Standards:** The SW community managed to (informally or formally) standardize important aspects of their assets. This is best witnessed by the standards set by W3C, but other de facto standards like the OWL API be mentioned as well.
- **Projects:** The SW community is involved in a huge number of research projects with tangible outcomes, ranging from pure research projects in one scientific institution to the involvement in huge applied research projects with several academic or industrial partners.

Concerning standards, achieving an ISO standard for common logic (CL)²⁰ has been an important step into the right direction. The CL standard has gained some visibility²¹, but is still four years after being established it does not have a strong lobby and is not very often quoted.²² To the author's opinion, it is not very likely that the conceptual graphs community will gain significantly higher impact or reputation through further standardization activities. A better way to achieve more impact is through conducting or participation in (applied) research projects. A good example are the activities of Gerd Stumme's Knowledge and Knowledge and Data Engineering Group²³, which covers both internal projects which meanwhile gained high reputation (e.g. bibsonomy, which is since 2008 used within SAP as well, which indicates the usefulness and maturity of bibsonomy), or publicly funded projects with partners, like Nepomuk.²⁴ Another example is the recently started research project CUBIST, lead by SAP Research with partners from the CS community.²⁵ It is the author's belief that the the CS community has relevant expertise concerning ST for enterprises, and a higher engagement in projects would better unleash these valuable assets.

6 Conclusion

“Beware of the hype!” This is a quote found on quite a number of presentations about Semantic Web technologies. Though SW and ST, as discussed, is not the same, this quotation should be applied to ST for Enterprises as well. A well-known approach to describe the maturity, adoption and social application of a given technology are Gartner's hype cycles. According to [15], “since its unveiling, the Semantic Web has been full of promise, but largely unfulfilled. In the last few years this has changed

²⁰ http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39175

²¹ For example, Pat Hayes keynote at ISWC 09 refers to common logic. Interestingly, additionally he introduces Peirce's ideas of graph surfaces and negation to RDF. See http://videlectures.net/iswc09_hayes_blogic/

²² To some extent, this situation can be compared to topic maps, which gained ISO standardization as well.

²³ <http://www.kde.cs.uni-kassel.de>

²⁴ A list of projects is provided on their webpage, see <http://www.kde.cs.uni-kassel.de/projekte>

²⁵ <http://www.cubist-project.eu>

[...]”, and Gartner refers to the interest of enterprises in ST which caused that change. Gartner rates the benefits of SW high, and it still sees SW at the peak of interests.

As discussed in this paper, ST can indeed fulfill some of its promises. Anyhow, with the turn from academic research to real-world applications in enterprises, a new set of challenges arises. Some of these challenges, like scalability and performance issues or educational challenges are rather general and somewhat ST-agnostic, but for enterprises, it is of great significance to cope with them.

The maturity of ST tools is still not sufficient for enterprises, but it is emerging. Anyhow, even very mature ST tools are likely to fail in meeting some expectations (particularly if these expectations stem from the bold promises made at the dawn of ST), and moreover, new technologies usually do not only solve existing problems, but raise new problems as well. But this will not stop the “semantic wave”, the emergence of ST for consumer and enterprise applications. Instead, in the long run, the author expects ST to become one of many mainstream and ubiquitous technologies, both the benefits and the costs will become widely demonstrated and accepted (this is Gartner’s “plateau of productivity”, the end of the hype cycle of a technology). It is still time to shape ST on its path to become of the bricks in future enterprise IT environments.

Disclaimer: Parts of this work have been carried out in the Aletheia project and in the CUBIST project (“Combining and Uniting Business Intelligence with Semantic Technologies”). Aletheia is sponsored by the German Federal Ministry of Education and Research. CUBIST is funded by the European Commission under the 7th Framework Programme of ICT, topic 4.3: Intelligent Information Management.

References

- [1] Horowitz, P. (ed.): PwC Technology Forecast (Spring 2009), <http://www.pwc.com/us/en/technology-forecast/spring2009/index.jhtml>. 2009 (retrieved September 2010)
- [2] Final Demand driven Mapping Report. Public report D3.2 from the research project value-it, <http://www.value-it.eu> (retrieved September 02, 2010)
- [3] SAP Research Dresden: Lessons on Semantic Information Integration and Access in the Aletheia Research Project. SAP internal whitepaper (2011)
- [4] W3C RDB2RDF Incubator Group: A Survey of Current Approaches for Mapping of Relational Databases to RDF. Report (2009), http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf
- [5] Moulton, L.: Semantic Software Technologies: Landscape of High Value Applications for the Enterprise. Gilbane Group Report (2010), http://www.expertsystem.net/documenti/pdf_eng/technology/semanticsoftwaretechnologies_gilbane2010.pdf (retrieved October 08, 2010)
- [6] Deliverable 3.1.1: Report on Knowledge Extraction from Structured Sources. Public deliverable from the research project LOD2 - Creating Knowledge out of Interlinked Data (2011), <http://static.LOD2.eu/Deliverables/deliverable-3.1.1.pdf>

- [7] Stark, A., Schroll, M., Hafkesbrink, J.: Die Zukunft des Semantic Web. Think!innowise Trend Report (2009), <http://www.innowise.eu/Dokumente/Trendreport.pdf> (retrieved October 2010)
- [8] West, D.: What Semantic Technology Means to Application Development Professionals. Forrester Research, Inc Report (October 2009)
- [9] Lunn, B.: Creative Destruction 7 Act Play. Series on semanticweb.com (2010), http://semanticweb.com/index-to-the-creative-destruction-7-act-play_b624 (retrieved October 2010)
- [10] Axelrod, S.: MDM is Not Enough - Semantic Enterprise is Needed. Information Management Special Report (March 2008), <http://www.information-management.com/> (retrieved September 2010)
- [11] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
- [12] Yee, K.P., et al.: Faceted Metadata for Image Search and Browsing. In: Proceedings of the Conference on Human Factors in Computing Systems. ACM Press, New York (2003)
- [13] Oberle, D., et al.: Challenges in Adopting Semantic Technologies. In: Pan, Z. (ed.) Ontology-Driven Software Engineering, ch. 11. Springer, Heidelberg (to appear)
- [14] Kreis, M.: Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und Medien Internationale Delphi-Studie 2030 (2010)
- [15] Valdes, R., Phifer, G., Murphy, J., Knipp, E., Mitchell Smith, D., Cearley, D.W.: Hype Cycle for Web and User Interaction Technologies, Gartner Report (2010)
- [16] Norman, D.: Cognitive engineering. In: Norman, D., Daper, S. (eds.) User Centered System Design, pp. 31–61. Lawrence Erlbaum Associates, Hills-dale (1988)
- [17] Hitzler, P., Scharfe, H. (eds.): Conceptual Structures in Practice. CRC Press, Boca Raton (2009)
- [18] Chein, M., Mugnier, M.-L.: Graph-based Knowledge Representation. In: Computational Foundations of Conceptual Graphs. Springer, London (2009)

Utility and Feasibility of Reasoning beyond Decidability in Semantic Technologies

Sebastian Rudolph and Michael Schneider

Institute AIFB, Karlsruhe Institute of Technology, DE
rudolph@kit.edu

FZI Research Center for Information Technology, Karlsruhe, DE
schneid@fzi.de

Abstract. Semantic Web knowledge representation standards such as RDF and OWL have gained momentum in the last years and are widely applied today. In the course of the standardization process of these and other knowledge representation formalisms, decidability of logical entailment has often been advocated as a central design criterion. On the other hand, restricting to decidable formalisms inevitably comes with constraints in terms of modeling power. Therefore, in this paper, we examine the requirement of decidability and weigh its importance in different scenarios. Subsequently, we discuss a way to establish incomplete – yet useful – reasoning support for undecidable formalisms by deploying machinery from the successful domain of theorem proving in first-order predicate logic. While elaborating on the undecidable variants of the ontology language OWL 2 as our primary examples, we argue that this approach could likewise serve as a role model for knowledge representation formalisms from the Conceptual Structures community.

1 Introduction

Today, the Semantic Web serves as the primary testbed for practical application of knowledge representation. A plethora of formalisms for representing and reasoning with Web knowledge has been designed and standardized under the auspices of the World Wide Web Consortium (W3C). While the early days of this endeavor saw ad-hoc and semantically underspecified approaches, interoperability requirements enforced their evolution into mature logical languages with clearly specified formal semantics. In the process of defining more and more expressive such formalisms, an often-debated requirement is decidability of logical entailment, i.e. the principled existence of an algorithm that decides whether a body of knowledge has a certain proposition as a consequence. While it goes without saying that such an algorithm is clearly useful for all kind of querying or knowledge management tasks, results established back in the 1930s show that this property does not hold for all types of logics [6,24]. In particular, in many expressive knowledge representation formalisms (most notably first-order predicate logic), entailment is undecidable.

Hence, whenever a knowledge representation formalism is to be designed, the trade-off between decidability and expressivity has to be taken into account. An examination of the Semantic Web languages hitherto standardized by the W3C yields a mixed picture in that respect: logical entailment in the basic data description language RDF [13] and its light-weight terminological extension RDF Schema [4] is decidable (although already NP-complete in both cases). Within the OWL 2 language family [17], only the most expressive variant OWL 2 Full [21] is undecidable, whereas OWL 2 DL [16,15] as well as its specified sublanguages (called tractable profiles) OWL 2 EL, OWL 2 QL, OWL 2 RL [14] are decidable (the latter three even in polynomial time). On the other hand, for the rule interchange format RIF [12], only the very elementary core dialect RIF-Core [2] is decidable whereas already the basic logic dialect RIF-BLD [3] – and hence every prospective extension of it – turns out to be undecidable.

This small survey already shows that decidability is far from being a common feature of the standardized Semantic Web languages. However, extensive reasoning and knowledge engineering support is currently only available for the decidable languages in the form of RDF(S) triple stores or OWL 2 DL reasoners hinting at a clear practitioners’ focus on these languages.

In this paper, we argue that inferencing support is important and feasible also in formalisms which are undecidable and we provide an outlook how this can be achieved, referring to our recent work on reasoning in undecidable Semantic Web languages as a showcase. We proceed as follows: Section 2 will remind the reader of the important notions from theoretical computer science. Section 3 proposes a schematic classification of inferencing algorithms by their practical usefulness. Section 4 distinguishes cases where decidability is crucial to enable “failsafe” reasoning from cases where it may make sense to trade decidability for expressivity. After these general considerations, we turn to variants of OWL to demonstrate ways to provide reasoning support for undecidable Semantic Web formalisms. To this end, Section 5 gives an overview of OWL syntaxes and the associated semantics. Section 6 shows two different ways of translating OWL reasoning problems into first-order logic and Section 7 briefly reports on our recent work of employing FOL reasoners in that context. In Section 8, we discuss ramifications of our ideas for Common Logic. Section 9 concludes. An extended version of this paper with examples of reasoning in diverse undecidable languages is available as technical report [20].

2 Recap: Decidability and Semidecidability

Let us first recap some basic notions from theoretical computer science which are essential for our considerations. From an abstract viewpoint, a logic is just a (possibly infinite) set of *sentences*. The *syntax* of the logic defines how these sentences look like. The *semantics* of the logic is captured by an *entailment relation* \models between sets of sentences Φ and sentences φ of the logic. $\Phi \models \varphi$ then means that Φ logically entails φ or that φ is a logical consequence of Φ . Usually, the logical entailment relation is defined in a model-theoretic way.

A logic is said to have a *decidable* entailment problem (often this wording is shortened as to calling the logic itself decidable – we will adopt this common practice in the following) if there is an algorithm which, taking as an input a finite set $\Phi = \{\phi_1, \dots, \phi_n\}$ of sentences of that logic and a further sentence φ , always terminates and provides the output *YES* iff $\Phi \models \varphi$ or *NO* iff $\Phi \not\models \varphi$. As a standard example for a decidable logic, propositional logic is often mentioned, a straightforward decision procedure being based on truth tables. However, there are decidable logics of much higher expressivity, e.g., the guarded fragment of first-order logic [4].

A logic is said to have a *semidecidable* entailment problem (also here, this can be abbreviated by calling the logic itself semidecidable) if there exists an algorithm that, again given Φ and φ as input, terminates and provides the output *YES* iff $\Phi \models \varphi$, but may not terminate otherwise. Consequently, such an algorithm is complete in the following sense: every consequence will eventually be identified as such. However the algorithm cannot be used to get a guarantee that a certain sentence is *not* a consequence. Clearly, every decidable logic is also semidecidable, yet, the converse does not hold. The prototypical example for a logic that is semidecidable but not decidable is first-order predicate logic (FOL). While undecidability of FOL can be shown by encoding the halting problem of a Turing machine into a FOL entailment problem, its semidecidability is a consequence from the fact that there exists a sound and complete deduction calculus for FOL [7], hence every consequence can be found in finite time by a breadth-first search in the space of all proofs w.r.t. that deduction calculus. Clearly, today’s first-order theorem provers use much more elaborated and goal-directed strategies to find a proof of a given entailment.

Obviously, in semi-decidable logics, the critical task which cannot be completely solved, is to detect the non-entailment $\Phi \not\models \varphi$. In model-theoretically defined logics such as FOL, $\Phi \not\models \varphi$ means that there exists a model \mathcal{M} of Φ that is not a model of φ . In other words, finding such a model means proving the above non-entailment. Indeed, there are rather effective (yet incomplete) FOL model finders available dedicated to this purpose. For straightforward reasons, most of these model finders focus on finite models. In fact, if there is a finite model with the wanted property, it is always possible to find it (due to the reason that the set of finite models is enumerable and first-order model checking is easy). Hence, the case which is intrinsically hard to automatically detect is when $\Phi \not\models \varphi$ but every model of Φ that is not a model of φ has *infinite* size. While seemingly exotic at first sight, such cases exist and are not very hard to construct. An example for such a situation is the question whether $\varphi = \exists x.(p(x, x))$ is a logical consequence of $\Phi = \{\varphi_1, \varphi_2\}$ with $\varphi_1 = \forall x.\exists y.(p(x, y))$ and $\varphi_2 = \forall x\forall y\forall z.(p(x, y) \wedge p(y, z) \rightarrow p(x, z))$. In this example, φ_1 enforces that in every model of Φ every element must be in a p -relationship to something, whereas φ_2 requires that p must be interpreted by a transitive relation. A side effect of p ’s transitivity is that whenever a model contains a p -cycle, all the elements in that cycle are p -related to themselves which makes φ satisfied. Therefore, any model of Φ that does not satisfy φ must

be p -cycle-free which is only possible if the model is infinite. The problem with infinite models is that, even if one has a method to represent and refer to them somehow, the set of all infinite models cannot fully be enumerated. Hence, whatever enumeration strategy is used, it will only cover a strict subset of all possible models.

3 A Classification of Decision Procedures by Usefulness

We will now take a closer look on the question how useful a sound and complete decision algorithm may be in practice. For the sake of better presentation, assume that we consider not all the (infinitely many) possible inputs to the algorithm but only the finitely many (denoted by the set P) below a fixed size s . Let us furthermore make the very simplifying assumption that every entailment problem in P is equally “important” in the sense that it will be posed to our reasoning algorithm with the same probability as the other problems. Now, a decision algorithm \mathcal{A} comes with the guarantee, that it terminates on each of the inputs after finite time, hence it gives rise to a function $\text{runtime}_{\mathcal{A}} : P \rightarrow \mathbb{R}^+$ assigning to each of the inputs the corresponding (finite) runtime of the algorithm. Now, the algorithm can be described by a *characteristic curve* assigning to a time span Δt the fraction of elements from P on which \mathcal{A} terminates after time less or equal to Δt . Formally, this function $\text{char}_{\mathcal{A}} : \mathbb{R}^+ \rightarrow [0, 1]$ would be defined by

$$\text{char}_{\mathcal{A}}(\Delta t) = \frac{|\{p \in P \mid \text{runtime}_{\mathcal{A}}(p) \leq \Delta t\}|}{|P|}.$$

Figure [1](#) schematically displays characteristic curves which may be encountered for complete decision procedures. As a common feature, note that the curves are always monotonic by definition. Moreover, every such curve will hit the 100% line at some time point due to the facts that we have a complete decision algorithm and P is finite.

We now assume that the decision algorithm is to be employed in a practical scenario, which gives rise to the following specific figures:

- a maximal time span that is worth to be spent on the computation of an answer to an entailment problem of size s , referred to as *acceptable waiting time*;
- a ratio characterizing the probability of getting an answer below which a use of the algorithm will be considered worthless, called the *perceived added value threshold*; and
- a ratio characterizing the probability of getting an answer above which the algorithm can be considered practically reliable, called the *acceptable reliability threshold*.

Figure [1](#) also depicts these values, according to which the four schematic characteristic curves can now be coarsely distinguished in terms of practical usefulness.

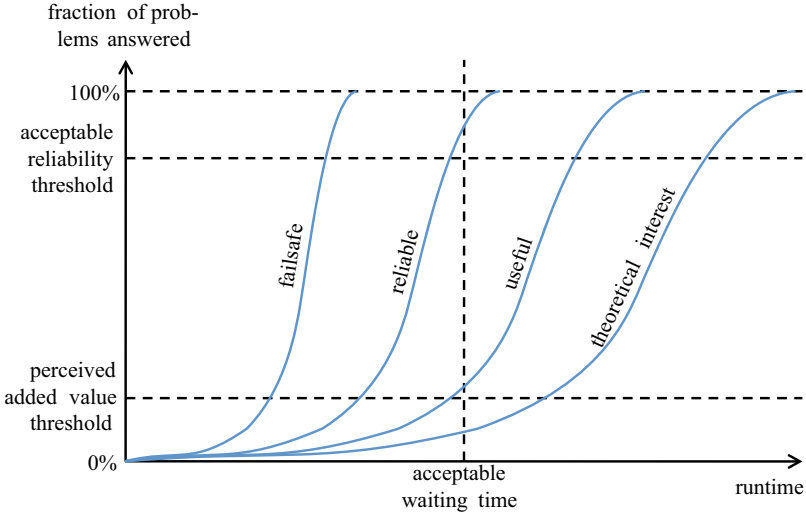


Fig. 1. Schematic representation of different characteristic curves for complete decision algorithms

- In the ideal case, the maximal runtime of the algorithm is smaller than the acceptable waiting time. Then every size s problem is guaranteed to be decided within the available time span, which allows for calling the algorithm *failsafe*.
- If this guarantee cannot be given, yet the probability that a solution will be obtained in the available time lies above the acceptable reliability threshold, the algorithm can still be said to be (practically) *reliable* and may be used within regular and automated knowledge management work flows. Then the rare cases not being covered could be dealt with by a kind of controlled exception handling mechanism.
- If the expected frequency of termination within the available time span is below the acceptable reliability threshold but above the perceived added value threshold, the algorithm's output should be perceived as nice-to-have additional information which may be taken into account if available. The overall work flow in which it is to be used should not crucially rely on this, yet we can still see that the algorithm may be rightfully called *useful*.
- Finally, if the ratio of the timely obtainable answers is even below the perceived added value threshold, the algorithm is of little or no practical value. Note however, that the existence of a complete decision algorithm – even if it happens to lie within this class – is still a research question worthwhile pursuing since optimizations and hardware improvements may well turn such an algorithm into something practically useful. Conversely, the proven non-existence of a decision algorithm may prevent many ambitious researchers from vainly trying to establish one. This justifies to at least characterize this

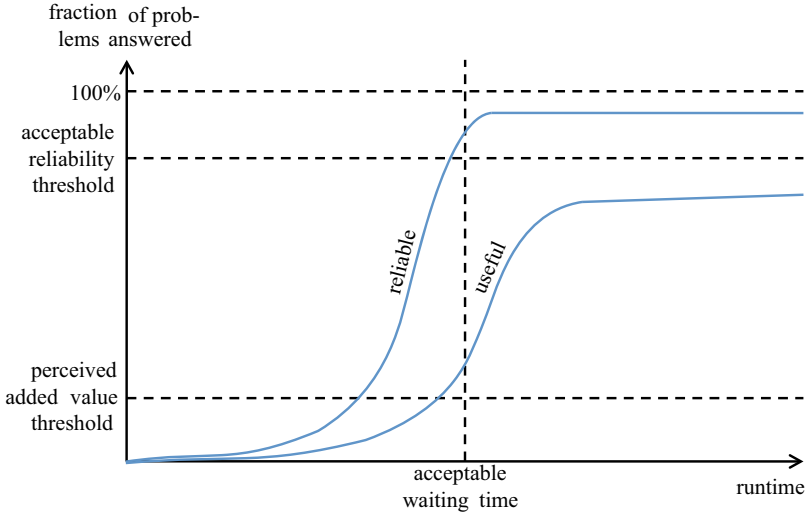


Fig. 2. Sketches of characteristic curves for incomplete decision procedures of practical interest

type of algorithm as of *theoretical interest*. For a prototypical example of this kind see [19].

Now, turning our attention to incomplete decision algorithms, we can assign to them characteristic curves in the same way as to complete ones (extending the range of runtime_A to $\mathbb{R}^+ \cup \{\infty\}$). The only difference here is, that the curve will not hit the 100% line. Nevertheless, as Fig. 2 illustrates, there may be incomplete algorithms still satisfying the usefulness or even the reliability criterion defined above leading to the conclusion that there may be cases where the practically relevant behavior of incomplete decision algorithms is just as good as that of complete ones, the notable exception being that no failsafe behavior can be obtained. It should be noted here that inferencing in expressive decidable formalisms comes with high worst-case complexities (normally ExpTime and higher, for instance N2ExpTime for OWL 2 DL) which implies that there are “malicious” inputs of already small size for which the runtime exceeds any reasonable waiting time. Although the runtime on average cases is usually much better, this fact normally vitiates failsafety guarantees. Of course, the case is different for so-called tractable languages which are of time-polynomial or even lower complexity.

4 On the Importance of Failsafe Decision Procedures

Having identified the possible existence of a failsafe decision procedure as the only principled practical advantage that the use of a decidable formalism may have, let us now investigate under which circumstances such a procedure is

strictly needed and in what cases it is dispensable. In the following, we will identify general criteria for this assessment.

4.1 Where Failsafe Decidability Is Crucial

Failsafe decidability is important in situations where automated reasoning is a central part of the functionality of a knowledge-based system and the questions posed to the system are normally “yes-or-no” questions of a symmetric form, the answers to which are to be used to trigger different follow-up actions in a highly or purely automated system (“if yes, then do this, otherwise do that”).

One important case of this is when entailment checks are to be carried out in “closed-world situations,” that is, when one can assume that all necessary information about a state of affairs has been collected in a knowledge base such that the non-entailment of a queried proposition justifies the assumption that the situation expressed by that proposition does indeed not hold. Under these circumstances, both possible answers to an entailment check provide information about the real state of affairs.

In other cases, the entailment check may be aimed at finding out facts about the considered knowledge base itself, instead of the real-world domain that it describes. In fact, this can be seen as a particular closed-world situation. As an example for this, consider the reasoning task of classification, i.e. the computation of a conceptual hierarchy. Here, the typical reasoning task to be performed is to answer the question “are two given classes in a subsumption relationship, or not?” This is a common task in today’s ontology management systems, useful both for supporting human ontology engineers and speeding up further reasoning tasks.

4.2 Where Failsafe Decidability Is Dispensable

Yet, there are also scenarios, where failsafe decision procedures for logical entailment seem to be of less importance.

First of all, this is the case when the emphasis of the usage of knowledge representation is on modeling rather than on automated inferencing. In fact, there will often be situations where logical languages are just used for noting down specifications in an unambiguous way and probably making use of available tool support for modeling and navigating these specifications. Clearly, in these cases, no reasoning support whatsoever is required, let alone failsafe decision procedures.

In other cases, reasoning may still be required, yet the available knowledge is assumed to be sound but incomplete w.r.t. the described domain, i.e. we have an “open-world situation.” Then, non-entailments cannot be conceived as guarantees for non-validity in the domain, as opposed to entailments, which (given that the knowledge base is sound) ensure validity. As a consequence thereof, non-entailment information is of less immediate value and an entailment checker just notifying the user of established entailments may be sufficient.

A more concrete case where failsafe decidability will often not be a strict requirement is human modeling support that will alarm the knowledge engineer upon the detection of inconsistencies in the knowledge base. (Note that detecting if a knowledge base KB is inconsistent is equivalent to checking the entailment $KB \models false$.) In practical ontology engineering, one certainly wants to make sure that a created ontology is free of semantic errors, and therefore good reasoning support for the detection of inconsistencies should be available. For many application domains, it will then be sufficient if such inconsistency checking does not find an issue after some considerable time of searching, while true consistency confirmation will only be nice to have.

Query answering scenarios represent a further type of setting normally not requiring failsafe decision procedures. When doing query answering, as e.g. in SPARQL, one has, in the simplest case, one or more axioms containing variables where otherwise individuals or class expressions would occur. One asks for all solutions that semantically follow from the queried knowledge base and match the query pattern. In this scenario, one is not interested in non-solutions, but in an *enumeration* of solutions. Therefore, what one needs is an enumeration algorithm, which does not really require a complete decision procedure. Also, from the practical viewpoint, in many applications such as search, completeness of the presented set of solutions is less important than when just one entailment is to be checked. What normally matters is that *enough* solutions are provided and that *relevant* solutions are among them, relevance being a measure that has to be defined for the specific purpose.

5 OWL, Syntactic and Semantic Variants

After these abstract considerations we will turn to OWL 2 as a specific knowledge representation formalism where the aforementioned issues are of particular interest, since both decidable and undecidable versions exist as well as well-investigated reasoning approaches for either. We will particularly focus on approaches for reasoning in undecidable formalisms.

First, let us recap the main aspects of syntax and semantics of OWL 2 to the degree needed for our considerations. When dealing with the Web Ontology Language OWL, one has to distinguish between two representation strategies which are interrelated but not fully interchangeable. The so called *functional syntax* [16] emphasizes the formula structure of what is said in the logic. As an example, consider the fact that an individual characterized as “happy cat owner” indeed owns some cat and everything he or she cares for is healthy. Expressed in OWL functional syntax, this statement would look as follows:

```
SubClassOf (
  ex:HappyCatOwner
  ObjectIntersectionOf (
    ObjectSomeValuesFrom( ex:owns ex:Cat )
    ObjectAllValuesFrom( ex:caresFor ex:Healthy ) ) )
```

On the other hand, there is the RDF syntax [18] which expresses all information in a graph (which in turn is usually encoded as a set of vertice-edge-vertice triples labeled with so-called Uniform Resource Identifiers, short: URIs). The

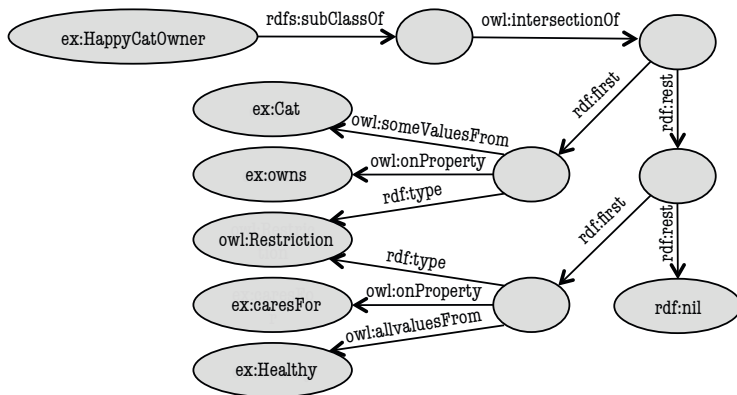


Fig. 3. RDF representation as graph

above proposition in RDF representation would look as displayed in Fig. 3. As opposed to the XML-based syntax often used for machine processing, the so-called Turtle syntax better reflects the triple structure, yet allows for abbreviations for structural bnodes¹ and list structures. The Turtle representation of the RDF graph from Fig. 3 would look as follows:

```
ex:HappyCatOwner rdfs:subClassOf
  [ owl:intersectionOf
    ( [ rdf:type owl:Restriction ;
      owl:onProperty ex:owns ;
      owl:someValuesFrom ex:Cat ]
      [ rdf:type owl:Restriction ;
      owl:onProperty ex:caresFor ;
      owl:allValuesFrom ex:Healthy ] ) ] .
```

Clearly, every functional syntax ontology description can be transformed into RDF representation. The converse is, however, not always the case. For all ontologies expressible in functional syntax, the specification provides the so-called *direct semantics* [15]. This semantics is very close to the extensional semantics commonly used in description logics: an interpretation is defined by choosing a set as domain, URIs denoting individuals are mapped to elements of that domain, class URIs to subsets and property URIs to sets of pairs of domain elements. It is noteworthy that the class of ontologies expressible via the functional syntax is

¹ Bnodes, also called blank nodes, are unlabeled vertices in the RDF graph representation and often used as auxiliary elements to encode complex structures.

not decidable, but only a subclass of it where further, so-called *global restrictions* apply. It is this decidable class which is referred to as OWL 2 DL and for which comprehensive tool support is available both in terms of ontology management, infrastructure and inferencing (software implementing decision procedures for entailment are typically called *reasoners*).

On the other hand, there is a formal semantics applicable to arbitrary RDF graphs. This so-called *RDF-based semantics* [21] is more in the spirit of the semantics of RDF(S): all URIs are mapped to domain elements in the first place and might be further interpreted through an extension function.

The two different semantics are related. A *correspondence theorem* [21] ensures that, given certain conditions which are easy to establish, the direct semantics on an ontology in functional syntax (taking into account the global restrictions of OWL 2 DL) will only provide conclusions that the RDF-based semantics provides from the ontology’s RDF counterpart as well. However, the RDF-based semantics will often provide additional conclusions that are not provided by the direct semantics, for instance conclusions based on metamodeling. The difference between the two semantics becomes significant for those ontologies that are still covered by the direct semantics but are not OWL 2 DL ontologies, i.e. ontologies that are beyond the scope of the correspondence theorem. Furthermore, there are ontologies to which only the RDF-based semantics applies but not the direct semantics, since not every RDF graph can be translated into an ontology in functional syntax. Figure 4 summarizes the relationships just stated.

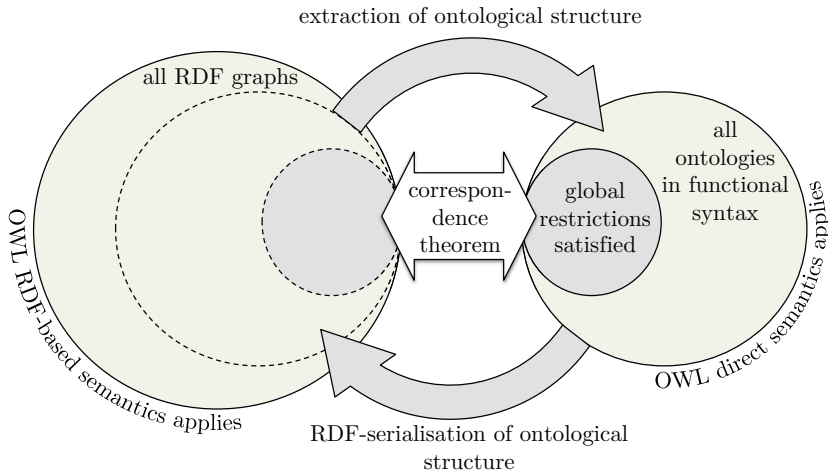


Fig. 4. Syntaxes and semantics for OWL 2

6 FOL-Empowered OWL Reasoning beyond Decidability

As stated before, for the decidable fraction of OWL 2, decision procedures have been implemented and are widely used today. Here, we will focus on the problem

of providing practical inferencing support for the more expressive yet undecidable versions of OWL. Thereby, we want to question an attitude sometimes encountered in the OWL community according to which the quest for automated inferencing beyond decidability is futile.

It seems in place to take a look at the prototypical example of undecidability: first-order predicate logic. Despite the fact that its undecidability was proven way before the wide-spread adoption of computers, automated first-order logic reasoning has always been a focus of interest and is meanwhile well-established also in practical applications such as hard- and software verification [26]. There are many FOL theorem provers which are able to find out when a set of FOL formulas is unsatisfiable or implies another set of formulas. Even model-finding is well supported, although typically restricted to finding finite models.²

In view of the success of FOL-based inferencing, which also substantiates our claim that reasoning in undecidable formalisms can be useful, it seems tempting to harness the comprehensive theoretical work and highly optimized implementations originating from that line of research for coping with inferencing problems in the undecidable variants of OWL. As it turns out, OWL inferencing with respect to both syntaxes and according semantics admits translation into FOL entailment problems. In the following, we will describe these embeddings in a bit more detail.

6.1 Translating Direct Semantics Reasoning into FOL

Given an ontology in functional syntax, the translation into FOL can be performed along the lines of the standard translation of description logics into FOL as, e.g., described in [9]. For instance, the ontology axiom presented in Section 5 would be translated into the following FOL sentence:

$$\forall x. \text{HappyCatOwner}(x) \rightarrow \exists y. (\text{owns}(x, y) \wedge \text{Cat}(y)) \wedge \forall z. (\text{caresFor}(x, z) \rightarrow \text{Healthy}(z))$$

We see that URIs referring to classes are translated into unary predicates while those denoting properties are mapped to binary predicates. The existential and the universal property restrictions have been translated into an existentially and a universally quantified subformula, respectively. The two subformulas are connected by a conjunction symbol, which is the result of translating the intersection class expression. Finally, the outermost class subsumption axiom has been translated into a universally quantified implication formula.

After this transformation, reasoning can be performed by means of FOL reasoners: FOL theorem provers can be used to find entailments and to detect inconsistent ontologies, whereas FOL model finders can be used to detect non-entailment and to confirm consistency of an ontology. In fact this approach is not new but has already been demonstrated in [23].

² For a comprehensive collection of online-testable state-of-the-art provers and model finders, we refer the reader to

<http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP>

In general, one can use this approach to embed OWL 2 DL into arbitrary subsets of FOL. This includes reasoning in the OWL 2 direct semantics beyond OWL 2 DL, i.e., reasoning with ontologies that are given in the functional syntax but are not constrained by any of the global restrictions of OWL 2 DL. This relaxation allows, for example, to recognize circular relationships such as cyclic chemical molecules, a feature that has often been asked for but is not available in OWL 2 DL due to its syntactic restrictiveness (see e.g. [25]). This strategy further covers many well-known undecidable extensions of OWL such as the semantic web rules language SWRL [10] as well as the combination of the rule interchange dialect RIF BLD with OWL described in [5].

6.2 Translating RDF-Based Semantics Reasoning into FOL

Under the RDF-based semantics, a translation into FOL is also possible but works differently than for the direct semantics. The RDF graph representation of the example ontology is translated into a conjunction of ternary atomic FOL formulas, which correspond to the different RDF triples in the RDF graph:

$$\begin{aligned} &\exists b_0, b_1, b_2, b_3, b_4. (\\ &\quad \text{iext}(\text{rdfs:subclassOf}, \text{ex:HappyCatOwner}, b_0) \\ &\quad \wedge \text{iext}(\text{owl:intersectionOf}, b_0, b_1) \\ &\quad \wedge \text{iext}(\text{rdf:first}, b_1, b_3) \\ &\quad \wedge \text{iext}(\text{rdf:rest}, b_1, b_2) \\ &\quad \wedge \text{iext}(\text{rdf:first}, b_2, b_4) \\ &\quad \wedge \text{iext}(\text{rdf:rest}, b_2, \text{rdf:nil}) \\ &\quad \wedge \text{iext}(\text{rdf:type}, b_3, \text{owl:Restriction}) \\ &\quad \wedge \text{iext}(\text{owl:onProperty}, b_3, \text{ex:owns}) \\ &\quad \wedge \text{iext}(\text{owl:someValuesFrom}, b_3, \text{ex:Cat}) \\ &\quad \wedge \text{iext}(\text{rdf:type}, b_4, \text{owl:Restriction}) \\ &\quad \wedge \text{iext}(\text{owl:onProperty}, b_4, \text{ex:caresFor}) \\ &\quad \wedge \text{iext}(\text{owl:allValuesFrom}, b_4, \text{ex:Healthy})) \end{aligned}$$

All atoms are built from a single FOL predicate ‘iext’, which corresponds to the function ‘IEXT(.)’ used in the RDF-based semantics to represent *property extensions*. Terms within the atoms are either constants or existentially quantified variables, which correspond to the URIs and the blank nodes in the RDF triples, respectively.

The above formula only represents the RDF graph itself without its meaning as an OWL 2 Full ontology. The OWL 2 Full meaning of an RDF graph is primarily defined by the collection of model-theoretic *semantic conditions* that underly the RDF-based semantics. The semantic conditions add meaning to the terms being used in the graph, such as ‘rdfs:subclassOf’, and by this means put constraints on the use of the ‘IEXT(.)’ function. The semantic conditions have the form of first-order formulas and can therefore be directly translated into FOL formulas. For example, the FOL representation for the semantic condition

about the term ‘`rdfs:subClassOf`’ [21, Sec. 5.8] has the form:

$$\begin{aligned} & \forall c_1, c_2. (\text{iext}(\text{rdfs:subClassOf}, c_1, c_2) \leftrightarrow \\ & \quad \text{iext}(\text{rdf:type}, c_1, \text{owl:Class}) \\ & \quad \wedge \text{iext}(\text{rdf:type}, c_2, \text{owl:Class}) \\ & \quad \wedge \forall x. (\text{iext}(\text{rdf:type}, x, c_1) \rightarrow \text{iext}(\text{rdf:type}, x, c_2))) \end{aligned}$$

The RDF-based semantics consists of several hundred semantic conditions. The meaning of the example ontology is given by the FOL translation of the actual RDF graph plus the FOL translations of all the semantic conditions of the RDF-based semantics.

While the semantic conditions of the RDF-based semantics only quantify over elements of the domain, a restricted form of higher-order logic (HOL) is provided by both the syntax and semantics of OWL 2 Full, without however the full semantic expressivity of HOL. Nevertheless, ontologies with flexible metamodeling are supported and some useful HOL-style reasoning results can be obtained.

7 Experimental Results

In previous work, we have translated the OWL 2 RDF-based semantics into a FOL theory and done a series of reasoning experiments using FOL reasoners, which generally indicate that rather complicated reasoning beyond OWL 2 DL is possible. These experiments included reasoning based on metamodeling as well as on syntactic constellations that violate the global restrictions of OWL 2 DL. The used FOL reasoners generally succeeded on most or all of the executed tests, and even often did so considerably fast, while the state-of-the-art OWL 2 DL reasoners that were used for comparison only succeeded on a small fraction of the tests. However, one often observed problem of the FOL reasoners needs to be mentioned: while they were very successful in solving complicated problems, they often showed difficulties on large input sizes. In the future, we will have to further investigate how to cope with this scalability issue. Our results have been described in [22].

In addition, we have recently conducted some initial experiments concerning reasoning in the direct semantics beyond OWL 2 DL, including positive and negative entailment checking based on cyclic relationships. Again, we have found that FOL reasoners can often quickly solve such problems, provided that the input ontologies are of reasonable size.

8 Come on, Logic!

Common Logic (CL) [11] has been proposed as a unifying approach to different knowledge representation formalisms, including several of the formalisms currently deployed on the Web. Semantically, CL is firmly rooted in FOI³, whereas

³ Strictly speaking, in order to keep within FOL, one has to avoid the use of so called *sequence markers*.

its syntax has been designed in a way that accounts for the open spirit of the Web by avoiding syntactic constraints typically occurring in FOL, such as fixed arities of predicates, or the strict distinction between predicate symbols and terms.

The syntactic freedom that CL provides allows for flexible modeling in a higher-order logic (HOL) style and it is even possible to receive some useful HOL-style semantic results, similar to metamodeling in OWL 2 Full. In fact, OWL 2 Full can be translated into CL in an even more natural way than into standard FOL, as demonstrated by Hayes [8]. Furthermore, CL allows to represent the OWL 2 Direct Semantics, as well as other Semantic Web formalisms, such as SWRL and RIF. Compared to all these Semantic Web formalisms, CL is significantly more expressive and flexible and, therefore, users of CL should benefit from extended modeling and reasoning capabilities.

By becoming an ISO standard, CL has taken another dissemination path than the commonly adopted Semantic Web languages. While syntax and semantics of CL are well-defined, software support for editing and managing CL knowledge bases has just started to be developed and support for reasoning in CL is close to non-existent to the best of our knowledge. This arguably constitutes the major obstacle for wide deployment – as we have argued in the preceding section, the often criticized undecidability of CL does not qualify as a sufficient reason to dismiss this formalism right away.

While the development of scalable, ergonomic tools is certainly an endeavor that should not be underestimated, the above presented approach provides a clear strategy for accomplishing readily available state-of-the-art reasoning support. In fact, since the translation of CL into FOL is even more direct than for the two considered variants of OWL, creating a reasoning back-end for CL should be even more straight-forward.

We are convinced that a system capable of reading CL knowledge bases and performing inferences with it would lead to a significant breakthrough toward a wider visibility and practical deployment of CL.

9 Conclusion

In our paper, we have discussed the importance of decidability for practical knowledge representation, putting particular emphasis on well-established Semantic Web languages. On a general level, we argued that from a practical perspective, decidability only provides a qualitative advantage if it comes with a decision algorithm the runtime of which can be guaranteed to be below an acceptable time span. We then identified scenarios where such an algorithm is strictly required as well as scenarios where this is not the case.

We therefore conclude that the necessity to constrain to decidable formalisms strongly depends on the typical automated reasoning tasks to be performed in a knowledge-based system dedicated to a concrete purpose.

In order to still provide necessary reasoning services, we proposed to use available highly optimized implementations of first-order theorem provers and model finders. Realistically, specialized reasoners such as existing OWL 2 DL

reasoners are likely to be more efficient on their specific language fragment than generic FOL reasoners. But as these reasoners happen to provide (syntactically restricted) FOL reasoning themselves, all considered reasoners are largely interoperable and, therefore, could be applied in parallel to solve complex reasoning tasks conjointly. This approach offers a smooth transition path towards advanced OWL 2 reasoning without loss of efficiency on existing application areas.

Our own experiments, currently using off-the-shelf standard first-order reasoners, have yielded first encouraging results. We are confident that the good results obtained for undecidable OWL variants will carry over to Semantic Web knowledge representation formalisms that even go beyond the OWL 2 specification, such as SWRL or RIF-BLD combined with OWL 2. We will, in principle, only be limited by what first-order logic provides us.

In the light of these promising results, we strongly believe that the described strategy of providing inferencing services via a translation into FOL and the deployment of first-order reasoning machinery can also pave the way to establishing reasoning support for undecidable formalisms cherished by the conceptual structures community. Endowing conceptual graphs and common logic with ready-to-use inferencing services along the same lines seems a feasible and worthwhile endeavor and will most likely lead to a more wide-spread adoption of these formalisms among knowledge representation practitioners.

References

1. Andréka, H., van Benthem, J.F.A.K., Németi, I.: Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic* 27(3), 217–274 (1998)
2. Boley, H., Hallmark, G., Kifer, M., Paschke, A., Polleres, A., Reynolds, D. (eds.): RIF Core Dialect. W3C Recommendation (June 22, 2010), <http://www.w3.org/TR/rif-core/>
3. Boley, H., Kifer, M. (eds.): RIF Basic Logic Dialect. W3C Recommendation (June 22, 2010), <http://www.w3.org/TR/rif-bld/>
4. Brickley, D., Guha, R. (eds.): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation (February 10, 2004), <http://www.w3.org/TR/rdf-schema/>
5. de Bruijn, J. (ed.): RIF RDF and OWL Compatibility. W3C Recommendation (June 22, 2010), <http://www.w3.org/TR/rif-rdf-owl/>
6. Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. *Monatshefte für Mathematik und Physik* 38, 173–198 (1931)
7. Gödel, K.: Über die Vollständigkeit des Logikkalküls. Ph.D. thesis, Universität Wien (1929)
8. Hayes, P.: Translating Semantic Web Languages into Common Logic. Tech. rep., IHMC Florida Institute for Human & Machine Cognition, 40 South Alcaniz Street, Pensacola, FL 32502 (July 18, 2005), <http://www.ihmc.us/users/phayes/CL/SW2SCL.html>
9. Hitzler, P., Krötzsch, M., Rudolph, S.: *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC (2009)
10. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B.N., Dean, M.: SWRL: A Semantic Web Rule Language. W3C Member Submission (May 21, 2004), <http://www.w3.org/Submission/SWRL/>

11. ISO/IEC JTC 1: Common Logic (CL): A Framework for a Family of Logic-based Languages. No. ISO/IEC 24707: 2007(E), ISO International Standard (October 1, 2007), <http://cl.tamu.edu/>
12. Kifer, M., Boley, H. (eds.): RIF Overview. W3C Recommendation (June 22, 2010), <http://www.w3.org/TR/rif-overview/>
13. Manola, F., Miller, E. (eds.): Resource Description Framework (RDF). Primer. W3C Recommendation (February 10, 2004), <http://www.w3.org/TR/rdf-primer/>
14. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language: Profiles. W3C Recommendation (October 27, 2009), <http://www.w3.org/TR/owl2-profiles/>
15. Motik, B., Patel-Schneider, P.F., Cuenca Grau, B. (eds.): OWL 2 Web Ontology Language: Direct Semantics. W3C Recommendation (October 27, 2009), <http://www.w3.org/TR/owl2-direct-semantics/>
16. Motik, B., Patel-Schneider, P.F., Parsia, B. (eds.): OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C Recommendation (October 27, 2009), <http://www.w3.org/TR/owl2-syntax/>
17. OWL Working Group, W: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (October 27, 2009), <http://www.w3.org/TR/owl2-overview/>
18. Patel-Schneider, P.F., Motik, B. (eds.): OWL 2 Web Ontology Language: Mapping to RDF Graphs. W3C Recommendation (October 27, 2009), <http://www.w3.org/TR/owl2-mapping-to-rdf/>
19. Rudolph, S., Glimm, B.: Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *J. Artif. Intell. Res.* (JAIR) 39, 429–481 (2010)
20. Rudolph, S., Schneider, M.: On the utility and feasibility of reasoning with undecidable semantic web formalisms. Technical Report 3016, Institute AIFB, Karlsruhe Institute of Technology (2011), <http://www.aifb.kit.edu/web/Techreport3016/en>
21. Schneider, M. (ed.): OWL 2 Web Ontology Language: RDF-Based Semantics. W3C Recommendation (27 October 2009), <http://www.w3.org/TR/owl2-rdf-based-semantics/>
22. Schneider, M., Sutcliffe, G.: Reasoning in the OWL 2 Full Ontology Language using First-Order Automated Theorem Proving. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) Proceedings of the 23rd International Conference on Automated Deduction, CADE 23 (2011) (to appear)
23. Tsarkov, D., Riazanov, A., Bechhofer, S., Horrocks, I.: Using Vampire to Reason with OWL. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 471–485. Springer, Heidelberg (2004)
24. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* 42(2), 230–265 (1937)
25. Villanueva-Rosales, N., Dumontier, M.: Describing Chemical Functional Groups in OWL-DL for the Classification of Chemical Compounds. In: Golbreich, C., Kalyanpur, A., Parsia, B. (eds.) Proceedings of the 3rd International Workshop on OWL: Experiences and Directions (OWLED 2007). CEUR Workshop Proceedings, vol. 258 (2007), <http://ceur-ws.org/Vol-258/paper28.pdf>
26. Voronkov, A.: Automated Reasoning: Past Story and New Trends. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), pp. 1607–1612. Morgan Kaufmann Publishers Inc., San Francisco (2003)

Cognitive Architectures for Conceptual Structures

John F. Sowa

VivoMind Research, LLC

Abstract. The book *Conceptual Structures: Information Processing in Mind and Machine* surveyed the state of the art in artificial intelligence and cognitive science in the early 1980s and outlined a cognitive architecture as a foundation for further research and development. The basic ideas stimulated a broad range of research that built on and extended the original topics. This paper reviews that architecture and compares it to four other cognitive architectures with their roots in the same era: Cyc, Soar, Society of Mind, and Neurocognitive Networks. The CS architecture has some overlaps with each of the others, but it also has some characteristic features of its own: a foundation in Peirce's logic and semiotics; a grounding of symbols in Peirce's twin gates of perception and action; and a treatment of logic as a refinement and extension of more primitive mechanisms of language and reasoning. The concluding section surveys the VivoMind Cognitive Architecture, which builds on and extends the original version presented in the CS book.

1 Cognitive Architectures

A *cognitive architecture* is a design for a computational system for simulating some aspect of human cognition. During the past half century, dozens of cognitive architectures have been proposed, implemented, and compared with human performance (Samsonovich 2010). The book *Conceptual Structures* (Sowa 1984) surveyed the state of the art in the early 1980s and proposed a design that has stimulated a broad range of research and development projects. After more than a quarter century, it's time to review the progress in terms of recent developments in cognitive science, artificial intelligence, and computational linguistics. To provide perspective, it's useful to review some related architectures that have also been under development for a quarter century or more: Cyc, Soar, Society of Mind, and Neurocognitive Networks.

The Cyc project, whose name comes from the stressed syllable of *encyclopedia*, was chartered in 1984 as an engineering project. It placed a higher priority on computational efficiency than simulating psycholinguistic theories. Its technical foundation was based on the previous decade of research on knowledge-based systems (Lenat & Feigenbaum 1987):

- Lenat estimated that encyclopedic coverage of the common knowledge of typical high-school graduates would require 30,000 articles with about 30 concepts per article, for a total of about 900,000 concepts.

- The Japanese Electronic Dictionary Research Project (EDR) estimated that the knowledge of an educated speaker of several languages would require about 200K concepts represented in each language.
- Marvin Minsky noted that less than 200,000 hours elapses between birth and age 21. If each person adds four new concepts per hour, the total would be less than a million.

All three estimates suggested that human-level cognition could be achieved with a knowledge base of about a million concept definitions. At a cost of \$50 per definition, Lenat and Feigenbaum believed that the project could be finished in one decade for \$50 million and less than two person-centuries of work.

After the first five years, Cyc had become an informal system of frames with heuristic procedures for processing them (Lenat & Guha 1990). But as the knowledge base grew, the dangers of contradictions, spurious inferences, and incompatibilities became critical. The developers decided to design a more structured representation with more systematic and tightly controlled procedures. Eventually, the CycL language and its inference engines evolved as a superset of first-order logic with extensions to support defaults, modality, metalanguage, and higher-order logic. An important innovation was a context mechanism for partitioning the knowledge base into a basic core and an open-ended collection of independently developed *microtheories* (Guha 1991).

After the first 25 years, Cyc grew far beyond its original goals: 100 million dollars had been invested in 10 person-centuries of work to define 600,000 concepts by 5 million axioms organized in 6,000 microtheories. Cyc can also access relational databases and the Semantic Web to supplement its own knowledge base. For some kinds of reasoning, Cyc is faster and more thorough than most humans. Yet Cyc is not as flexible as a child, and it can't read, write, or speak as well as a child. It has not yet reached the goal of acquiring new knowledge by reading a textbook and generating rules and definitions in CycL.

Unlike the engineering design for Cyc, the Soar design was based on “a unified theory of cognition” (Newell 1990), which evolved from four decades of earlier research in AI and cognitive science: the General Problem Solver as “a program that simulates human thought” (Newell & Simon 1961) and production rules for simulating “human problem solving” (Newell & Simon 1972). The foundations for Soar are based on the earlier mechanisms: production rules for procedural knowledge; semantic networks for declarative knowledge; and learning by building new units called *chunks* as assemblies of earlier units. Declarative knowledge can be stored in either long-term memory (LTM) or short-term (working) memory. It can represent semantic knowledge about concept definitions or episodic knowledge about particular instances of objects or occurrences. More recent extensions (Laird 2008) have added support for emotions and iconic memory for uninterpreted imagery.

In the books *Society of Mind* and *Emotion Engine*, Minsky (1986, 2006) presented a cognitive architecture that he had developed in five decades of research and collaboration with students and colleagues. In a review of Minsky's theories, Singh (2003) compared the Society of Mind to the Soar architecture:

- To the developers of Soar, the interesting question is what are the least set of basic mechanisms needed to support the widest range of cognitive processes. The

opposing argument of the Society of Mind theory is that the space of cognitive processes is so broad that no particular set of mechanisms has any special advantage; there will always be some things that are easy to implement in your cognitive architecture and other things that are hard. Perhaps the question we should be asking is not so much how do you unify all of AI into one cognitive architecture, but rather, how do you get several cognitive architectures to work together?

That question is the central theme of Minsky's books, but Singh admitted that the complexity of the ideas and the lack of detail has discouraged implementers: "While Soar has seen a series of implementations, the Society of Mind theory has not. Minsky chose to discuss many aspects of the theory but left many of the details for others to fill in. This, however, has been slow to happen."

Neurocognitive networks were developed by the linguist Sydney Lamb (1966, 1999, 2004, 2010), who had written a PhD dissertation on native American languages, directed an early project on machine translation, developed a theory of *stratificational grammar*, and spent five decades in studying and collaborating with neuroscientists. Lamb's fundamental assumption is that all knowledge consists of connections in networks and all reasoning is performed by making, strengthening, or weakening connections. That assumption, with variations, was the basis for his linguistic theories in the 1960s and his most recent neurocognitive networks. Lamb avoided the symbol-grounding problem by a simple ploy: he didn't assume any symbols — the meaning of any node in a network is purely determined by its direct or indirect connections to sensory inputs and motor outputs. Harrison (2000) implemented Lamb's hypothesis in the PureNet system and showed that it made some cognitively realistic predictions.

The *Conceptual Structures* book discussed early work by the developers of these four systems, but the influences were stronger than mere citations. The first version of conceptual graphs was written in 1968 as a term paper for Minsky's AI course at MIT. Among the topics in that course were the General Problem Solver and the semantic networks by Quillian (1966), whose advisers were Newell and Simon. The early cognitive influences evolved from another term paper written in 1968 for a psycholinguistics course at Harvard taught by David McNeill (1970). The first published paper on conceptual graphs (Sowa 1976) was written at IBM, but influenced by the research at Stanford that led to Cyc. One of the early implementations of CGs (Sowa & Way 1986) used software that evolved from the dissertation by Heidorn (1972), whose adviser was Sydney Lamb. The goal for conceptual structures was to synthesize all these sources in a psychologically realistic, linguistically motivated, logically sound, and computationally efficient cognitive architecture.

2 The CS Cognitive Architecture

The cognitive architecture of the *Conceptual Structures* book overlaps some aspects of each of the four architectures reviewed in Section 1. That is not surprising, since the founders of each had a strong influence on the book. But the CS architecture also has some unique features that originated from other sources:

- The first and most important is the logic and semiotics of Charles Sanders Peirce, who has been called “the first philosopher of the 21st century.” His ideas and orientation have influenced the presentation and organization of every aspect of the book and every feature that makes it unique.
- The second feature, which follows from Peirce and which is shared with Lamb, is to ground the symbolic aspects of cognition in the “twin gates” of perception and action. Chapter 2 begins with perception, and Chapter 3 treats conceptual graphs as a special case of perceptual graphs. The ultimate goal of all reasoning is purposive action.
- The third, which also originates with Peirce, is to treat logic as a branch of semiotics. Although some sentences in language can be translated to logic, the semantic foundation is based on prelinguistic mechanisms shared with the higher mammals. (Sowa 2010)
- The fourth, which originated in skepticism about AI before I ever took a course in the subject, is a critical outlook on the often exaggerated claims for the latest and greatest technology. It appears in a strong preference for Wittgenstein’s later philosophy, in which he criticized his first book and the assumptions by his mentors Frege and Russell. That skepticism is the basis for the concluding Chapter 7 on “The Limits of Conceptualization.” It also appears in later cautionary lectures and writings about “The Challenge of Knowledge Soup” (Sowa 2005).
- Finally, my preference for a historical perspective on every major topic helps avoid passing fads. Some so-called innovations are based on ideas that are as old as Aristotle and his sources, many of which came, directly or indirectly, from every civilization in Asia and Africa.

Figure 1 is a copy of Figure 2.2 in the CS book. It illustrates the hypothesis that the mechanisms of perception draw upon a stock of previous *percepts* to interpret incoming sensory *icons*. Those icons are uninterpreted input in the sensory projection areas of the cortex. The percepts are stored in LTM, which is also in an area of the cortex close to or perhaps identical with the same projection area. Percepts may be exact copies of earlier icons or parts of icons. But they could also be copies or parts of copies of a previous *working model*, which is assembled as an interpretation of the current sensory input.

The working model in Figure 1 is either an interpretation of sensory input or a mental model that has the same neural representation. In 1984, that assumption was controversial, and many AI researchers claimed that knowledge was stored and processed in a propositional form. Since then, the assumptions illustrated in Figure 1 have been supported by both psychological and neural evidence (Barsalou 2009). The following quotation explains Figure 1:

- “The associative comparator searches for available percepts that match all or part of an incoming sensory icon. Attention determines which parts of a sensory icon are matched first or which classes of percepts are searched.
- The assembler combines percepts from long-term memory under the guidance of schemata. The result is a working model that matches the sensory icons. Larger percepts assembled from smaller ones are added to the stock of percepts and become available for future matching by the associative comparator.

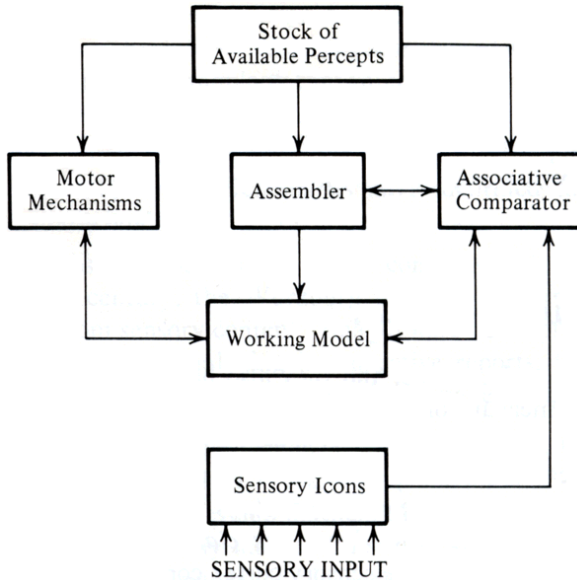


Fig. 1. Mechanisms of Perception

- Motor mechanisms help the assembler construct a working model, and they, in turn, are directed by a working model that represents the goal to be achieved.” (Sowa 1984:34)

Instead of assuming distinct mechanisms for propositions and mental imagery, Chapter 3 adds the assumption that the propositional representation in *conceptual graphs* is part of the same construction: “Perception is the process of building a *working model* that represents and interprets sensory input. The model has two components: a sensory part formed from a mosaic of *percepts*, each of which matches some aspect of the input; and a more abstract part called a *conceptual graph*, which describes how the percepts fit together to form the mosaic. Perception is based on the following mechanisms:

- Stimulation is recorded for a fraction of a second in a form called a *sensory icon*.
- The *associative comparator* searches long-term memory for percepts that match all or part of an icon.
- The *assembler* puts the percepts together in a working model that forms a close approximation to the input. A record of the assembly is stored as a conceptual graph.
- Conceptual mechanisms process *concrete concepts* that have associated percepts and *abstract concepts* that do not have any associated percepts.

When a person sees a cat, light waves reflected from the cat are received as a sensory icon s . The associative comparator matches s either to a single cat percept p or to a collection of percepts, which are combined by the assembler into a complete image. As the assembler combines percepts, it records the percepts and their interconnections

in a conceptual graph. In diagrams, conceptual graphs are drawn as linked boxes and circles. Those links represent logical associations in the brain, not the actual shapes of the neural excitations.” (Sowa 1984:69-70)

The CS book cited a variety of psychological and neural evidence, which is just as valid today as it ever was. But much more evidence has been gathered, and the old evidence has been interpreted in new ways. The primary hypothesis illustrated by Figure 1 has been supported: the mechanisms of perception are used to build and reason about mental models, and conceptual structures are intimately related to perceptual structures. The assumption that percepts can be related to one another by graphs is sufficiently general that it can't be contradicted. But the more specific assumption that those graphs are the same as those used for logic, language, and reasoning requires further research to fill in the details. The framework is sound, but the developments of the past quarter century have raised more issues to explore and questions to ask.

3 Neural and Psycholinguistic Evidence

Many of the controversies about implementing NLP systems are related to issues about how the human brain processes language. Figure 2 shows the left hemisphere of the brain; the base drawing was copied from Wikipedia, and the labels come from a variety of sources, of which MacNeilage (2008) is the most useful. Broca's area and Wernicke's area were the first two areas of the human brain recognized as critical to language. Lesions to Broca's area impair the ability to generate speech, but they cause only a minor impairment in the ability to recognize speech. Significantly, the impairment in recognition is caused by an inability to resolve ambiguities that depend on subtle syntactic features. Lesions to Wernicke's area impair the ability to understand language, but they don't impair the ability to generate syntactically correct speech. Unfortunately, that speech tends to be grammatical nonsense whose semantic content is incoherent.

The neural interconnections explain these observations: Wernicke's area is closely connected to the sensory projection areas for visual and auditory information. Wernicke's area is the first to receive speech input and link it to the store of semantic information derived from previous sensory input. Most of language can be interpreted by these linkages, even if Broca's area is damaged. Broca's area is close to the motor mechanisms for producing speech. It is responsible for fine-grained motions of various kinds, especially the detailed syntactic and phonological nuances in language generation. Lesions in Broca's area make it impossible to generate coherent syntactic structures and phonological patterns. For language understanding, Broca's area is not necessary to make semantic associations, but it can help resolve syntactic ambiguities.

These observations support the CS hypothesis that semantic-based methods are fundamental to language understanding. Wernicke's area processes semantics first, Broca's area operates in parallel to check syntax, and ambiguities in one can be resolved by information from the other. Meanwhile, the right hemisphere interprets pragmatics: emotion, prosody, context, metaphor, irony, and jokes, any of which could clarify, modify, or override syntax and semantics. Conflicts create puzzles that may require conscious attention (or laughter) to resolve.

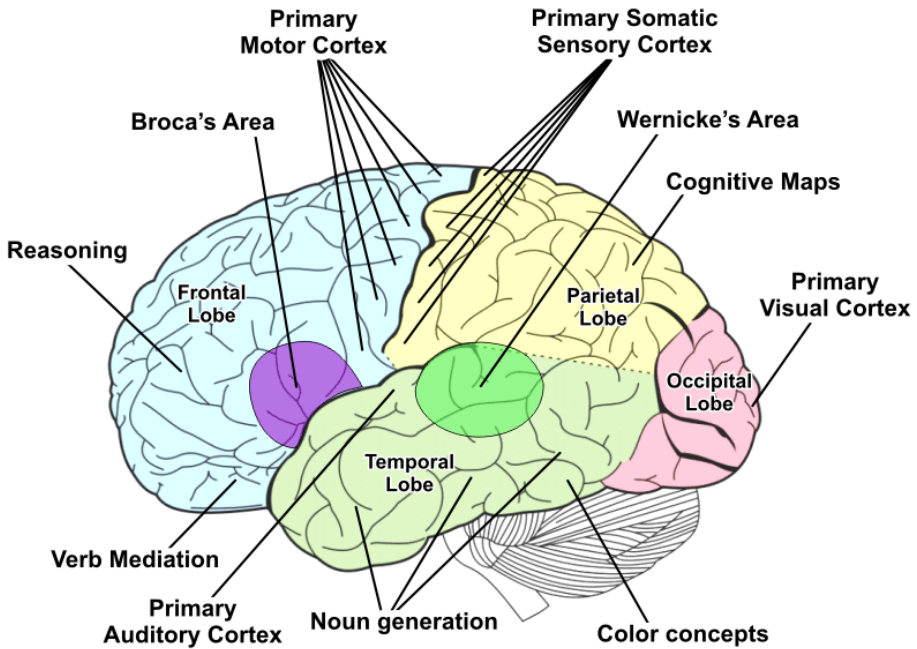


Fig. 2. Language areas of the left hemisphere

The evidence also gives some support for the claim that generative syntax is independent of semantics (Chomsky 1957). Lesions in Broca's area impair the ability to generate grammatical speech, and lesions in Wernicke's area cause patients to generate grammatically correct, but meaningless sentences. But there is no evidence for the claim of an innate "universal grammar." Furthermore, the strong evidence for the importance of pragmatics suggests that Chomsky's emphasis on competence is more of a distraction than an aid to understanding cognition.

MacNeilage (2008) and Bybee (2010) argued that the structural support required for language need not be innate. General cognitive abilities are sufficient for a child to learn the syntactic and semantic patterns. Some of the commonalities found in all languages could result from the need to convert the internal forms to and from the linear stream of speech. In evolutionary terms, the various language areas have different origins, and their functions have similarities to the corresponding areas in monkeys and apes. As Figure 2 shows, verbs are closely associated with motor mechanisms while nouns are more closely connected to perception. It suggests that the syntactic structure of verbs evolved from their association with the corresponding actions, but nouns have primarily semantic connections. Deacon (1997, 2004) argued that the cognitive limitations of infants would impose further constraints on the patterns common to all languages: any patterns that a highly distractible infant finds hard to learn will not be preserved from one generation to the next.

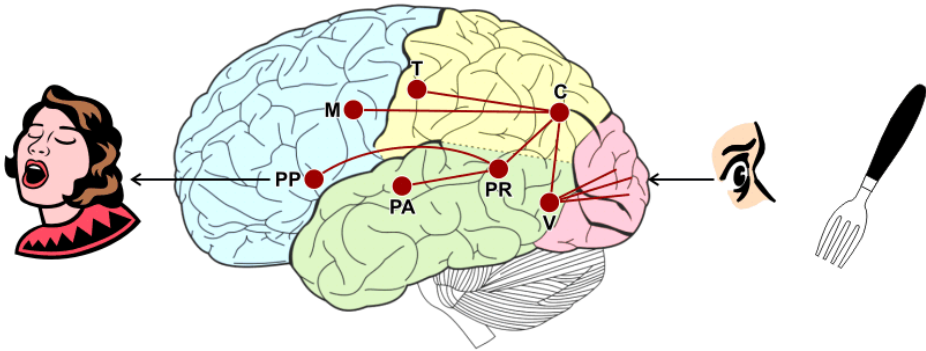


Fig. 3. Neurocognitive network for the word *fork*

Figure 3 overlays the base drawing of Figure 2 with a network of connections for the word *fork* as proposed by Lamb (2010). The node labeled C represents the concept of a fork. It occurs in the parietal lobe, which is closely linked to the primary projection areas for all the sensory modalities. For the image of a fork, C is connected to node V, which has links to percepts for the parts and features of a fork in the visual cortex (occipital lobe). For the tactile sensation of a fork, C links to node T in the sensory area for input from the hand. For the motor schemata for manipulating a fork, C links to node M in the motor area for the hand. For the phonology for recognizing the word *fork*, C links to node PR in Wernicke’s area. Finally, PR is linked to node PA for the sound /fork/ in the primary auditory cortex and to node PP in Broca’s area for producing the sound.

The network in Figure 3 represents *semantic* or *metalevel* information about the links from a concept node C to associated sensory, motor, and verbal nodes. It shows how Lamb solves the symbol-grounding problem. Similar networks can link instance nodes to type nodes to represent *episodic* information about particular people, places, things, and events. Lamb’s networks have many similarities to other versions of semantic networks, and they could be represented as conceptual graphs. CGs do have labels on the nodes, but those labels could be considered internal indexes that identify type nodes in Lamb’s networks. Those networks, however, cannot express all the logical options of CGs, CycL, and other AI systems. Only one additional feature is needed to support them, and Peirce showed how.

4 Peirce’s Logic and Semiotics

To support reasoning at the human level or at the level of Cyc and other engineering systems, a cognitive architecture requires the ability to express the logical operators used in ordinary language. Following are some sentences spoken by a child named Laura at age three (Limber 1973):

Here’s a seat. It must be mine if it’s a little one.

I want this doll because she’s big.

When I was a little girl I could go “geek-geek” like that. But now I can go “this is a chair.

In these sentences, Laura correctly expressed possibility, necessity, tenses, indexicals, conditionals, causality, quotations, and metalanguage about her own language at different stages of life. She had a fluent command of a larger subset of intensional logic than Montague formalized, but it's doubtful that her mental models support infinite families of possible worlds.

Lamb's neurocognitive networks can't express those sentences, but Peirce discovered a method for extending similar networks to express all of them. In 1885, he had invented the algebraic notation for predicate calculus and used it to express both first-order and higher-order logic. But he also experimented with graph notations to find a simpler way to express "the atoms and molecules of logic." His first version, called *relational graphs*, could express relations, conjunctions, and the existential quantifier. Following is a relational graph for the sentence *A cat is on a mat*:

Cat—On—Mat

In this notation, a bar by itself represents existence. The strings **Cat**, **On**, and **Mat** represent relations. In combination, the graph above says that there exists something, it's a cat, it's on something, and the thing it's on is a mat. Peirce invented this notation in 1883, but he couldn't find a systematic way to express all the formulas he could state in the algebraic notation. In 1897, he finally discovered a simple method: use an oval to enclose any graph or part of a graph that is negated. Peirce coined the term *existential graph* for relational graphs with the option of using ovals to negate any part. Figure 4 shows some examples.

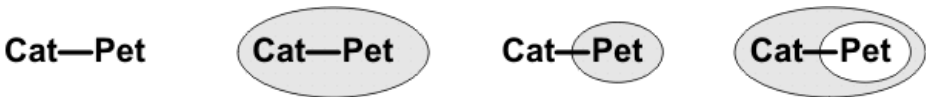


Fig. 4. Four existential graphs about pet cats

The first graph on the left of Figure 4 says that some cat is a pet. The second graph is completely contained in a shaded oval, which negates the entire statement. It says that no cat is a pet. The third graph negates just the pet relation. It says that some cat is not a pet. The fourth graph negates the third graph. The simplest way to negate a sentence is to put the phrase "It is false that" in front of it: *It is false that there exists a cat which is not a pet*. But that combination of two negations can be read in much more natural ways: with a conditional, *If there is a cat, then it is a pet*; or with a universal quantifier, *Every cat is a pet*. Both readings are logically equivalent.

In general, Peirce's relational graphs, when combined with ovals for negation, have the full expressive power of first-order logic. Peirce later experimented with other features to express higher-order logic, modal logic, and metalanguage. With these extensions, existential graphs (EGs) have the full expressive power of CycL and most other AI logics. The CS book adopted Peirce's EGs as the foundation for conceptual graphs. In effect, CGs are typed versions of EGs with some extra features. But every CG can be translated to a logically equivalent EG. For further discussion of EGs, CGs, their rules of inference, and their extensions to metalanguage and modalities, see the articles by Sowa (2003, 2006, 2009).

Even more important than the notation, the EG rules of inference do not require the complex substitutions and transformations of predicate calculus. They perform only two kinds of operations: inserting a graph or subgraph under certain conditions; or the inverse operation of deleting a graph or a subgraph under opposite conditions. These rules are sufficiently simple that they could be implemented on networks like Lamb’s with only the operations of making, strengthening, or weakening connections.

Peirce called EGs his “chef d’oeuvre” and claimed that the operations on EGs represented “a moving picture of the mind in thought.” After a detailed comparison of Peirce’s EGs to current theories about mental models, the psychologist Johnson-Laird (2002) agreed:

Peirce’s existential graphs are remarkable. They establish the feasibility of a diagrammatic system of reasoning equivalent to the first-order predicate calculus. They anticipate the theory of mental models in many respects, including their iconic and symbolic components, their eschewal of variables, and their fundamental operations of insertion and deletion. Much is known about the psychology of reasoning... But we still lack a comprehensive account of how individuals represent multiply-quantified assertions, and so the graphs may provide a guide to the future development of psychological theory.

Although Peirce is best known for his work on logic, he incorporated logic in a much broader theory of signs that subsumes all possible cognitive architectures within a common framework. Every thought, feeling, or perception is a sign. Semiotics includes neural networks because every signal that passes between neurons or within neurons is a sign. Even a single bacterium is a semiotic processor when it swims upstream in following a glucose gradient. But the most fundamental semiotic process in any life form is the act of reproducing itself by interpreting signs called DNA. Figure 5 illustrates the evolution of cognitive systems according to the sophistication of their semiotic abilities.

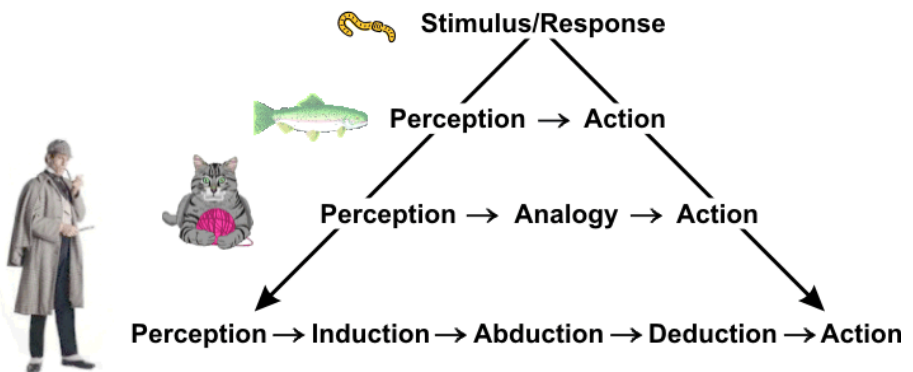


Fig. 5. Evolution of cognition

The cognitive architectures of the animals at each stage of Figure 5 build on and extend the capabilities of the simpler stages. The worms at the top have rudimentary

sensory and motor mechanisms connected by ganglia with a small number of neurons. A neural net that connects stimulus to response with just a few intermediate layers might be an adequate model. The fish brain is tiny compared to mammals, but it supports rich sensory and motor mechanisms. At the next stage, mammals have a cerebral cortex with distinct *projection areas* for each of the sensory and motor systems. It can support networks with analogies for case-based learning and reasoning. The cat playing with a ball of yarn is practicing hunting skills with a mouse analog. At the human level, Sherlock Holmes is famous for his ability at induction, abduction, and deduction. Peirce distinguished those three ways of using logic and observed that each of them may be classified as a disciplined special case of analogy.

5 VivoMind Cognitive Architecture

The single most important feature of the VivoMind Cognitive Architecture (VCA) is the high-speed Cognitive Memory™. The first version, implemented in the VivoMind Analogy Engine (VAE), was invented by Arun Majumdar to support the associative comparator illustrated in Figure 1. Another feature, which was inspired by Minsky's Society of Mind, is the distribution of intelligent processing among heterogeneous agents that communicate by passing messages in the Flexible Modular Framework™ (Sowa 2002). More recent research (Paradis 2009) supports *neurofunctional modularity* for human language processing. Practical experience on multithreaded systems with multiple CPUs has demonstrated the flexibility and scalability of a society of distributed heterogeneous agents:

- Asynchronous message passing for control and communication.
- Conceptual graphs for representing knowledge in the messages.
- Language understanding as a knowledge-based perceptual process.
- Analogies for rapidly accessing large volumes of knowledge of any kind.

Learning occurs at every step: perception and reasoning generate new conceptual graphs; analogies assimilate the CGs into Cognitive Memory™ for future use.

The VivoMind Language Processor (VLP) is a semantics-based language interpreter, which uses VAE as a high-speed associative memory and a society of agents for processing syntax, semantics, and pragmatics in parallel (Sowa & Majumdar 2003; Majumdar et al. 2008). During language analysis, thousands of agents may be involved, most of which remain dormant until they are triggered by something that matches their patterns. This architecture is not only computationally efficient, but it produces more accurate results than any single algorithm for NLP, either rule based or statistical.

With changing constraints on the permissible pattern matching, a general-purpose analogy engine can perform any combination of informal analogies or formal deduction, induction, and abduction. At the neat extreme, conceptual graphs have the model-theoretic semantics of Common Logic (ISO/IEC 24707), and VAE can find matching graphs that satisfy the strict constraints of unification. At the scruffy extreme, CGs can represent Schank's conceptual dependencies, scripts, MOPs, and TOPs. VAE can support case-based reasoning (Schank 1982) or any heuristics used

with semantic networks. Multiple reasoning methods — neat, scruffy, and statistical — support combinations of heterogeneous theories, encodings, and algorithms that are rarely exploited in AI.

The Structure-Mapping Engine (SME) pioneered a wide range of methods for using analogies (Falkenhainer et al. 1989; Lovett et al. 2010). But SME takes N -cubed time to find analogies in a knowledge base with N options. For better performance, conventional search engines can reduce the options, but they are based on an unordered bag of words or other labels. Methods that ignore the graph structure cannot find graphs with similar structure but different labels, and they find too many graphs with the same labels in different structures.

Organic chemists developed some of the fastest algorithms for representing large labeled graphs and efficiently finding graphs with similar structure and labels. Chemical graphs have fewer types of labels and links than conceptual graphs, but they have many similarities. Among them are frequently occurring subgraphs, such as a benzene ring or a methyl group, which can be defined and encoded as single types. Algorithms designed for chemical graphs (Levinson & Ellis 1992) were used in the first high-speed method for encoding, storing, and retrieving CGs in a generalization hierarchy. More recent algorithms encode and store millions of chemical graphs in a database and find similar graphs in logarithmic time (Rhodes et al. 2007). By using a measure of graph similarity and locality-sensitive hashing, their software can retrieve a set of similar graphs with each search.

The original version of VAE used algorithms related to those for chemical graphs. More recent variations have led to a family of algorithms that encode a graph in a *Cognitive Signature*TM that preserves both the structure and the ontology. The encoding time is polynomial in the size of a graph. With a semantic distance measure based on both the structure of the graphs and an ontology of their labels, locality-sensitive hashing can retrieve a set of similar graphs in $\log(N)$ time, where N is the total number of graphs in the knowledge base. With this speed, VAE can find analogies in a knowledge base of any size without requiring a search engine as a preliminary filter. For examples of applications, see the slides by Sowa and Majumdar (2009).

The distributed processing among heterogeneous agents supports Peirce's cycle of pragmatism, as illustrated in Figure 6. That cycle relates perception to action by repeated steps of induction, abduction, reasoning, and testing. Each step can be performed by an application of analogy or by a wide variety of specialized algorithms.

The cycle of pragmatism shows how the VivoMind architecture brings order out of a potential chaos (or Pandemonium). The labels on the arrows suggest the open-ended variety of heterogeneous algorithms, each performed by one or more agents. During the cycle, the details of the internal processing by any agent are irrelevant to other agents. It could be neat, scruffy, statistical, or biologically inspired. The only requirement is the conventions on the interface to the FMF. An agent that uses a different interface could be enclosed in a wrapper. The overall system is fail soft: a failing agent that doesn't respond to messages is automatically replaced by another agent that can answer the same messages, but perhaps in a very different way. Agents that consistently produce more useful results are rewarded with more time and space resources. Agents that are useless for one application might be rewarded in another application for which their talents are appropriate

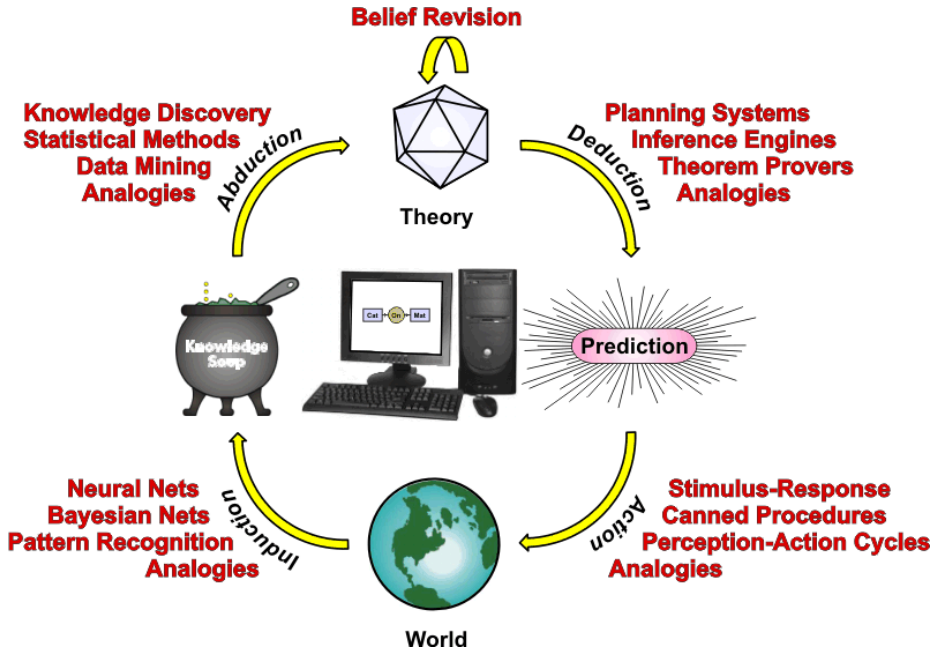


Fig. 6. Cycle of Pragmatism

The society of agents can have subsocieties that traverse the cycle of pragmatism at different speeds. Societies devoted to low-level perception and action may traverse each cycle in milliseconds. Societies for reasoning and planning may take seconds or minutes. A society for complex research might take hours, days, or even years.

References

- Barsalou, L.W.: Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society B* 364, 1281–1289 (2009)
- Bybee, J.: *Language, Usage, and Cognition*. University Press, Cambridge (2010)
- Chomsky, N.: *Syntactic Structures*. Mouton, The Hague (1957)
- Deacon, T.W.: *The Symbolic Species: The Co-evolution of Language and the Brain*. W. W. Norton, New York (1997)
- Deacon, T.W.: Memes as signs in the dynamic logic of semiosis: Beyond molecular science and computation theory. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) *ICCS 2004. LNCS (LNAI)*, vol. 3127, pp. 17–30. Springer, Heidelberg (2004)
- Falkenhainer, B., Forbus, K.D., Gentner, D.: The structure mapping engine: algorithm and examples. *Artificial Intelligence* 41, 1–63 (1989)
- Harrison, C.J.: *PureNet: A modeling program for neurocognitive linguistics*, PhD dissertation, Rice University (2000)
- Heidorn, G.E.: *Natural Language Inputs to a Simulation Programming System*, Report NPS-55HD72101A, Naval Postgraduate School, Monterey, CA (1972)
- ISO/IEC. *Common Logic (CL) — A Framework for a family of Logic-Based Languages*, IS 24707, International Organisation for Standardisation, Geneva (2007)

- Johnson-Laird, P.N.: Peirce, logic diagrams, and the elementary processes of reasoning. *Thinking and Reasoning* 8(2), 69–95 (2002)
- Laird, J.E.: Extending the Soar cognitive architecture. In: Wang, P., Goertzel, B., Franklin, S. (eds.) *Artificial General Intelligence 2008*, pp. 224–235. IOS Press, Amsterdam (2008)
- Lamb, S.M.: *Outline of Stratificational Grammar*. Georgetown University Press, Washington, DC (1966)
- Lamb, S.M.: *Pathways of the Brain: The Neurocognitive Basis of Language*. John Benjamins, Amsterdam (1999)
- Lamb, S.M.: *Language and Reality*. Continuum, London (2004)
- Lamb, S.M.: *Neurolinguistics, Lecture Notes for Linguistics 411*, Rice University (2010), <http://www.owl.net.rice.edu/~ling411>
- Lenat, D.B., Feigenbaum, E.A.: On the thresholds of knowledge. In: *Proc. IJCAI 1987*, pp. 1173–1182 (1987)
- Lenat, D.B., Guha, R.V.: *Building Large Knowledge-Based Systems*. Addison-Wesley, Reading (1990)
- Limber, J.: The genesis of complex sentences. In: Moore, T. (ed.) *Cognitive Development and the Acquisition of Language*, pp. 169–186. Academic Press, New York (1973)
- Lovett, A., Forbus, K., Usher, J.: A structure-mapping model of Raven’s Progressive Matrices. In: *Proceedings of CogSci 2010*, pp. 2761–2766 (2010)
- MacNeillage, P.F.: *The Origin of Speech*. University Press, Oxford (2008)
- Majumdar, A.K., Sowa, J.F., Stewart, J.: Pursuing the goal of language understanding. In: Eklund, P., Haemmerlé, O. (eds.) *ICCS 2008. LNCS (LNAI)*, vol. 5113, pp. 21–42. Springer, Heidelberg (2008), <http://www.jfsowa.com/pubs/pursuing.pdf>
- Majumdar, A.K., Sowa, J.F.: Two paradigms are better than one and multiple paradigms are even better. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) (2009), <http://www.jfsowa.com/pubs/pursuing.pdf>
- McNeill, D.: *The Acquisition of Language*. Harper & Row, New York (1970)
- Minsky, M.: *The Society of Mind*. Simon & Schuster, New York (1986)
- Minsky, M.L.: *The Emotion Machine: Commonsense Thinking*. In: *Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, New York (2006)
- Newell, A.: *Unified Theories of Cognition*. Harvard University Press, Cambridge (1990)
- Newell, A., Simon, H.A.: GPS, a program that simulates human thought (1961), reprinted in Feigenbaum & Feldman, pp. 279–293 (1963)
- Newell, A., Simon, H.A.: *Human Problem Solving*. Prentice-Hall, Englewood Cliffs (1972)
- Paradis, M.: *Declarative and Procedural Determinants of Second Languages*. John Benjamins, Amsterdam (2009)
- Peirce, C.S. (CP) *Collected Papers*. In: Hartshorne, C., Weiss, P., Burks, A. (eds.), vol. 8, Harvard University Press, Cambridge (1931-1958)
- Quillian, M.R.: *Semantic Memory*, Report AD-641671, Clearinghouse for Federal Scientific and Technical Information (1966)
- Samsonovich, A.V.: Toward a unified catalog of implemented cognitive architectures. In: Samsonovich, A.V., et al. (eds.) *Biologically Inspired Cognitive Architectures 2010*, pp. 195–244. IOS Press, Amsterdam (2010)
- Singh, P.: Examining the society of mind. *Computing and Informatics* 22, 521–543 (2003)
- Sowa, J.F.: Conceptual graphs for a data base interface. *IBM Journal of Research and Development* 20(4), 336–357 (1976), <http://www.jfsowa.com/pubs/cg1976.pdf>
- Sowa, J.F.: *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading (1984)
- Sowa, J.F.: Architectures for intelligent systems. *IBM Systems Journal* 41(3), 331–349 (2002), <http://www.jfsowa.com/pubs/arch.htm>

- Sowa, J.F.: Laws, facts, and contexts: Foundations for multimodal reasoning. In: Hendricks, V.F., Jørgensen, K.F., Pedersen, S.A. (eds.) *Knowledge Contributors*, pp. 145–184. Kluwer Academic Publishers, Dordrecht (2003), <http://www.jfsowa.com/pubs/laws.htm>
- Sowa, J.F.: The challenge of knowledge soup. In: Ramadas, J., Chunawala, S. (eds.) *Research Trends in Science, Technology, and Mathematics Education*, pp. 55–90. Homi Bhabha Centre, Mumbai (2005), <http://www.jfsowa.com/pubs/challenge.pdf>
- Sowa, J.F.: Worlds, Models, and Descriptions. *Studia Logica*, Special Issue Ways of Worlds II, 84(2), 323–360 (2006), <http://www.jfsowa.com/pubs/worlds.pdf>
- Sowa, J.F.: Conceptual Graphs for Conceptual Structures. In: Hitzler, P., Schärfe, H. (eds.) *Conceptual Structures in Practice*, pp. 102–136. Chapman & Hall/CRC Press, Boca Raton (2009), <http://www.jfsowa.com/pubs/cg4cs.pdf>
- Sowa, J.F.: The role of logic and ontology in language and reasoning. In: Poli, R., Seibt, J. (eds.) *Theory and Applications of Ontology: Philosophical Perspectives*. ch. 11, pp. 231–263. Springer, Berlin (2010), <http://www.jfsowa.com/pubs/rolelog.pdf>
- Sowa, J.F., Way, E.C.: Implementing a semantic interpreter using conceptual graphs. *IBM Journal of Research and Development* 30(1), 57–69 (1986), <http://www.jfsowa.com/pubs/cg1986.pdf>
- Sowa, J.F., Majumdar, A.K.: Analogical reasoning. In: de Moor, A., Lex, W., Ganter, B. (eds.) *ICCS 2003*. LNCS (LNAI), vol. 2746, pp. 16–36. Springer, Heidelberg (2003), <http://www.jfsowa.com/pubs/analog.htm>
- Sowa, J.F., Majumdar, A.K.: Slides of VivoMind applications (2009), <http://www.jfsowa.com/talks/pursue.pdf>

In-Close2, a High Performance Formal Concept Miner

Simon Andrews

Conceptual Structures Research Group
Communication and Computing Research Centre
Faculty of Arts, Computing, Engineering and Sciences
Sheffield Hallam University, Sheffield, UK
s.andrews@shu.ac.uk

Abstract. This paper presents a program, called **In-Close2**, that is a high performance realisation of the Close-by-One (CbO) algorithm. The design of In-Close2 is discussed and some new optimisation and data preprocessing techniques are presented. The performance of In-Close2 is favourably compared with another contemporary CbO variant called FCbO. An application of In-Close2 is given, using minimum support to reduce the size and complexity of a large formal context. Based on this application, an analysis of gene expression data is presented. In-Close2 can be downloaded from Sourceforge¹.

1 Introduction

The emergence of Formal Concept Analysis (FCA) as a data analysis technology [2, 15, 7] has increased the need for algorithms that compute formal concepts quickly. When Kuznetsov specified, in Close-by-One (CbO) [11], how repeated computations of concepts could be detected using their natural canonicity, it was no longer necessary to exhaustively search through previously generated concepts to determine the uniqueness of a newly generated one. Algorithms have been developed based on CbO, such as In-Close [1], FCbO [9] and FCbO's predecessor [8], which significantly outperform older algorithms that relied on searching [12].

In a recent, albeit small, international competition between concept computing algorithms [14], FCbO took first place and In-Close took second. In-Close2, the program described in this paper, develops the previous implementation of In-Close by adding breadth searching to allow elements of an intent to be inherited, and by implementing some new optimisation and data preprocessing techniques.

2 The In-Close2 Design

In-Close2 has been designed with the fast back-tracking canonicity test of In-Close [1] combined with a dual depth-first and breadth-first approach, used in

¹ <http://sourceforge.net/projects/inclose/>

algorithms such as FCbO [9], to provide attribute inheritance. In-Close2 incrementally closes parent concepts whilst noting new child extents along the way. The attributes collected during closure are then passed down to each child extent, so that during the closure of a child concept these attributes do not need to be tested for inclusion.

2.1 Structure of In-Close2

In In-Close2 a formal context is represented by a Boolean matrix, I , with m rows, representing a set of objects $\{0, 1, \dots, m - 1\}$, and n columns, representing a set of attributes $\{0, 1, \dots, n - 1\}$. For an object i and an attribute j , $I[i][j] = true$ says that object i has attribute j .

A formal concept is represented by an extent, $A[r]$ (an ordered list of objects), and an intent, $B[r]$ (an ordered list of attributes), where r is the concept number (index). For example, if $B[r] = (3, 5, 7)$, $B[r][2] = 7$. For the purposes of the following pseudocode, $A[r]$ and $B[r]$ will be treated as sets, where convenient. Thus, $B[r] \cup \{j\}$ appends attribute j to $B[r]$.

In the algorithm, there is a current attribute, j , the index of the parent concept, r , and a global index of the *candidate* new concept, r_{new} . The candidate concept is instantiated if it is canonical.

There are two procedures, $InCloseII(r, y)$ and $IsCanonical(r, r_{new}, j)$, where y is a starting attribute. The supremum is the concept with index 0 and is initialised as $A[0] = (0, 1, \dots, m - 1)$, $B[0] = \emptyset$. Initially, $r_{new} = 1$ and the invocation of $InCloseII$ is $InCloseII(0, 0)$.

The pseudocode is presented below, with a line-by-line explanation.

2.2 Explanation of Main Procedure, InCloseII

The procedure iterates across the context from y , forming intersections between the current extent and the next attribute extent. When the current extent is found in an attribute extent, that attribute is added to the current intent. When a different intersection results, its canonicity is tested to determine if it is a new extent. If it is new, it is added to the children of the current concept for closing later. The current intent is inherited by the children by passing it down though the recursion.

Lines 2 and 3 - Initially there are no children of the current concept.

Line 4 - Iterate over the context, starting at attribute y .

Line 5 - If the next attribute, j , is not an inherited attribute then...

Lines 6 to 9 - ...form a new extent by intersecting the current extent with the attribute extent of j .

Line 10 - If the new extent is the same as the current extent then...

Line 11 - ...add attribute j to the current intent.

Line 13 - Else if the new extent is canonical then...

Line 14 - ...add the starting attribute of the child concept,...

Line 15 - ...add the index number of the child concept,...

```

1 InCloseII( $r, y$ )
2    $jchildren \leftarrow \emptyset;$ 
3    $rchildren \leftarrow \emptyset;$ 
4   for  $j \leftarrow y$  upto  $n - 1$  do
5     if  $j \notin B[r]$  then
6        $A[r_{new}] \leftarrow \emptyset;$ 
7       foreach  $i$  in  $A[r]$  do
8         if  $I[i][j]$  then
9            $A[r_{new}] \leftarrow A[r_{new}] \cup \{i\};$ 
10      if  $A[r_{new}] = A[r]$  then
11         $B[r] \leftarrow B[r] \cup \{j\};$ 
12      else
13        if IsCanonical( $r, r_{new}, j$ ) then
14           $jchildren \leftarrow jchildren \cup \{j\};$ 
15           $rchildren \leftarrow rchildren \cup \{r_{new}\};$ 
16           $B[r_{new}] \leftarrow B[r] \cup \{j\};$ 
17           $r_{new} \leftarrow r_{new} + 1;$ 
18  for  $k \leftarrow 0$  upto  $|jchildren| - 1$  do
19    InCloseII( $rchildren[k], jchildren[k] + 1$ );
20 end

```

Line 16 - ...inherit the current intent and...

Line 17 - ...increment the concept index.

Line 18 - Iterate across the children...

Line 19 - ...closing each by passing the concept number and next attribute to InCloseII.

2.3 Explanation of Procedure IsCanonical

The procedure searches backwards in the context for the new extent, skipping attributes that are part of the current intent.

Line 2 - Starting at one less than the current attribute, j , iterate backwards across the context.

Line 3 - If the next attribute, k , is not part of the current intent then...

Lines 4 and 5 - ...intersect the new extent with the attribute extent of k .

Line 6 - If the extent is found, stop searching and return *false* (the new extent is not canonical).

Line 7 - If this line is reached, the new extent has not been found, so return *true* (the new extent is canonical).

 IsCanonical(r, r_{new}, j)

Result: Returns *false* if $A[r_{new}]$ is found, *true* if not found

```

1 begin
2   for  $k \leftarrow j - 1$  downto 0 do
3     if  $k \notin B[r]$  then
4       for  $h \leftarrow 0$  upto  $|A[r_{new}]| - 1$  do
5         if not  $I[A[r_{new}][h]][k]$  then break;
6       if  $h = |A[r_{new}]|$  then return false;
7   return true;
8 end
  
```

2.4 The Effect of Attribute Inheritance on Performance

Table 1 compares the number of intersections performed by In-Close2 (lines 6-17 of InCloseII, above), with and without attribute inheritance, using three well-known public data sets from UCI [4]. An ‘extended’ formal context of the UCI Adult data set is used that includes scaled continuous attributes from the data and was created using a context creation tool called FcaBedrock [3]². The context for the UCI Internet Ads data set used here also includes scaled continuous attributes, created using the same tool. Hence both the extended Adult and the Internet Ads contexts contain more attributes and concepts than is typically found in other publications. The times, in seconds, are also given. The experiments were carried out using a standard Windows PC with an Intel E4600 2.39GHz processor with 32KB of level one data cache and 3GB of RAM. Note that the times in Table 1 are taken from a version of In-Close2 that incorporates the data preprocessing and optimisation techniques presented later in this paper.

Table 1. Number of intersections performed and time taken by In-Close2, with and without attribute inheritance

UCI Dataset	Mushroom	Adult	Internet Ads
$ G \times M $	$8,124 \times 125$	$32,561 \times 124$	$3,279 \times 1,565$
#Concepts	226,921	1,388,468	16,570
no inheritance: #intersections	2,729,388	4,644,001	1,953,155
time	0.824	3.323	0.345
inheritance: #intersections	1,193,779	2,919,135	1,925,734
time	0.568	2.632	0.323

² <http://sourceforge.net/projects/fcabedrock/>

3 Data Preprocessing and Optimisation Techniques for Efficient Use of Cache Memory

3.1 Physical Column Sorting

The well-known technique of sorting context columns in ascending order of support is implemented in In-Close2. The typical approach is to sort pointers to the columns, rather than the columns themselves, as this takes less time. However, in In-Close2, the columns are physically sorted to make more efficient use of cache memory. If data is contiguous in RAM, cache lines will be filled with data that are more likely to be used when carrying out column intersections in `InCloseII` and when finding an already closed extent in `IsCanonical`. This significantly reduces level one data cache misses, particularly when large contexts are being processed. The overhead of physically sorting the context is outweighed by the saving in memory loads. A comparison between logical and physical column sorting is given in Table 2. The figures for level one data cache misses (L1-DCM) were measured using Intel’s V-Tune profiling tool.

Table 2. L1 data cache misses and time taken by In-Close2, comparing logical and physical column sorting

UCI Dataset	Mushroom	Adult	Internet Ads
$ G \times M $	$8,124 \times 125$	$32,561 \times 124$	$3,279 \times 1,565$
#Concepts	226,921	1,388,468	16,570
logical sort: L1-DCM	67,300,000	617,300,000	47,000,000
time	0.922	4.563	0.362
physical sort: L1-DCM	33,900,000	252,000,000	32,100,000
time	0.743	3.104	0.341

3.2 Reducing Row Hamming Distance

The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different [6]. In bit-strings this is the number of positions in the bit-strings at which 0s and 1s differ. In In-Close2, after physical column sorting, the *rows* are then also physically sorted to reduce the Hamming distance between consecutive rows. By treating rows as unsigned binary integers, sorting them numerically minimises the number of bits that change from row to row (see Figure 1). This increased uniformity in the data significantly reduces level one data cache misses (see Table 3).

3.3 Using a Bit-Array

The well-known technique of storing the context as a bit-array, rather than an array of bool (byte) data, is implemented in In-Close2. The usual reason for this is that it allows larger contexts to be stored in RAM. However, in the implementation of In-Close2, because the array is sorted physically there is a further

32	16	8	4	2	1	Value	HD	
	X			X		18		
		X				8	3	
	X					16	2	
			X	X		6	3	
		X			X	9	4	
	X			X		18	4	
X	X					48	2	
			X	X	X	7	5	
		X			X	9	3	
	X	X				24	2	
Total HD:							28	

Unsorted

32	16	8	4	2	1	Value	HD	
			X	X		6		
			X	X	X	7	1	
		X				8	3	
		X			X	9	1	
		X			X	9	0	
	X					16	3	
	X			X		18	1	
	X			X		18	0	
	X	X				24	2	
X	X					48	2	
Total HD:							13	

Sorted

Fig. 1. Row sorting reduces Hamming distance (HD)

Table 3. L1 data cache misses and time taken by In-Close2, with and without reduced Hamming distance

UCI Dataset	Mushroom	Adult	Internet Ads
$ G \times M $	8,124 × 125	32,561 × 124	3,279 × 1,565
#Concepts	226,921	1,388,468	16,570
HD not reduced: L1-DCM	33,900,000	252,000,000	32,100,000
time	0.743	3.104	0.341
HD reduced: L1-DCM	10,200,000	50,400,000	21,900,000
time	0.568	2.632	0.323

improvement in the efficiency of the cache. Although there is an overhead in the extra code required to access individual bits, once contexts have become too large to comfortably fit into cache memory in one-byte data form, this is outweighed by the efficiency gained in the use of the cache. Using two versions of In-Close2, one implementing the context as a bool-array and the other implementing the context as a bit-array, the point at which this occurs is clearly visible in Figure 2. Using a context density of 1% and 200 attributes, the number of objects was increased from 5000 to 25000. With fewer than 10,000 objects, the bool-array implementation was quicker. With more than 10,000 objects, the bit-array implementation was quicker.

4 In-Close2 Performance Evaluation

A number of experiments were carried out to compare In-Close2 with FCbO. An implementation of FCbO was supplied by an author of FCbO, with the understanding that it was a highly optimised version, but with only some details about the optimisations used. In testing, it was a faster version of FCbO than the one in the competition at ICCS 2010 [14].

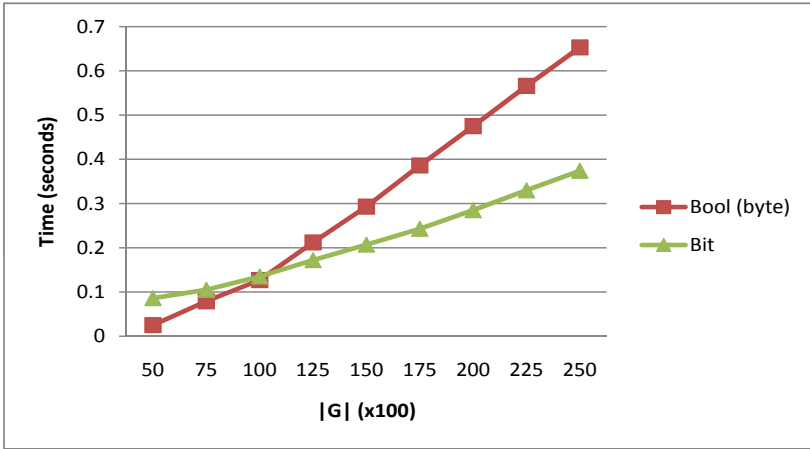


Fig. 2. Comparison of performance between bool and bit-array context data

Another promising program, called AddExtent, was also tested in the experiments. This program is an attribute-incremental implementation of an algorithm called AddIntent [16] and was supplied by an author of AddIntent.

Experiments were carried out using the three well known public data sets already mentioned in this paper, three artificial data sets and several series of random data sets. The public data sets allowed the programs to be compared under real conditions, the artificial data sets were a means of simulating real conditions but with larger data sizes and the random data sets allowed controlled comparison over key variables: number of attributes, context density and number of objects.

The experiments were carried out using a standard Windows PC with an Intel E4600 2.39GHz processor with 32KB of level one data cache memory and 3GB of RAM. The times for the programs are total times to include data pre-processing. The results are given below.

4.1 Public Data Set Experiments

The results of the public data set experiments are given in Table 4. There was no clear winner; each of the programs performing best with one of the data sets. The largest of the contexts was that of the Adult data set, and here In-Close2 significantly outperformed the other two. The strong performance of In-Close2 with regard to large context size is borne out by the results of the artificial and random data set experiments that follow.

Table 4. UCI data set results (timings in seconds)

	Mushroom	Adult	Internet Ads
$ G \times M $	$8,124 \times 125$	$32,561 \times 124$	$3,279 \times 1,565$
Density	17.36%	11.29%	0.97%
#Concepts	226,921	1,388,469	16,570
AddExtent	5.787	72.080	0.324
FCbO	0.508	5.687	0.812
In-Close2	0.568	2.365	0.328

4.2 Artificial Data Set Experiments

The following artificial data sets were used:

M7X10G120K - a data set based on simulating many-valued attributes. The scaling of many-valued attributes is simulated by creating ‘blocks’ in the context containing disjoint columns. There are 7 blocks, each containing 10 disjoint columns.

M10X30G120K - a similar data set, but with a context containing 10 blocks, each with 30 disjoint columns.

T10I4D100K - an artificial data set from the FIMI data set repository [5].

In these experiments, only times for FCbO and In-Close2 are given (Table 5), because times for AddExtent were very large. For each of the three artificial data sets, In-Close2 significantly outperformed FCbO.

Table 5. Artificial data set results (timings in seconds)

	M7X10G120K	M10X30G120K	T10I4D100K
$ G \times M $	$120,000 \times 70$	$120,000 \times 300$	$100,000 \times 1,000$
Density	10.00%	3.33%	1.01%
#Concepts	1,166,343	4,570,498	2,347,376
FCbO	4.281	28.812	40.765
In-Close2	2.233	20.648	24.277

4.3 Random Data Set Experiments

Three series of random data experiments were carried out to compare the performance of In-Close2 and FCbO, testing the affect of changes in the number of attributes, context density, and number of objects:

Attributes series - with 1% density and 10,000 objects, the number of attributes was varied between 400 and 1,800 (Figure 3). In-Close2 significantly outperformed FCbO, increasingly so as the number of attributes increased.

Density series - with 200 attributes and 10,000 objects, the density of 1s in the context was varied between 1 and 9% (Figure 4). The performance was simliar, although In-Close2 was slightly faster with densities below 7% and FCbO outperformed In-Close with densities greater than 7%, increasingly so as the density increased.

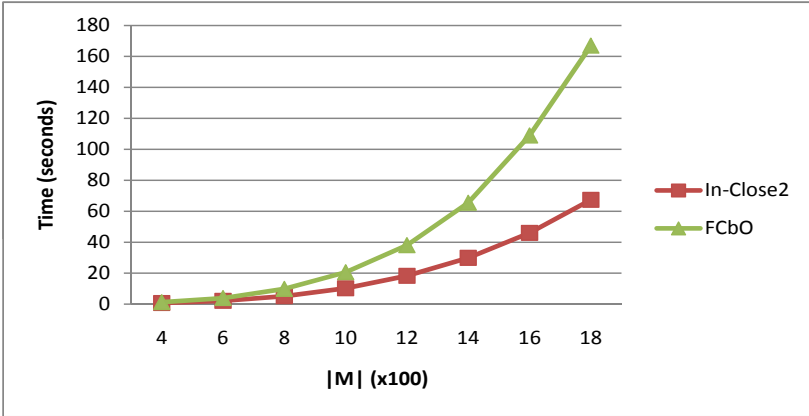


Fig. 3. Comparison of performance with varying number of attributes

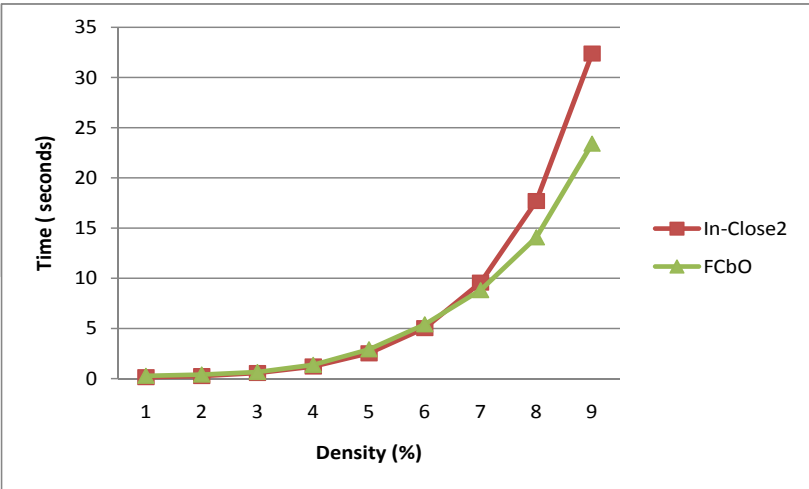


Fig. 4. Comparison of performance with varying context density

Objects series - with 1% density and 200 attributes, the number of objects was varied between 100,000 and 500,000 (Figure 5). In-Close2 significantly outperformed FCbO, consistently so as the number of objects increased.

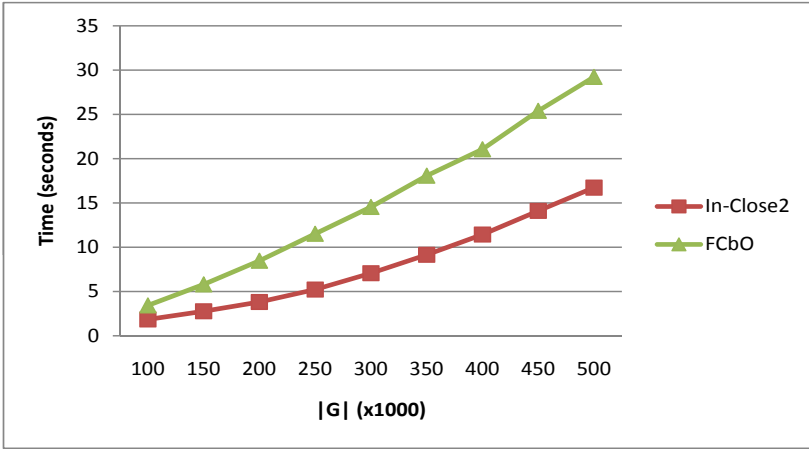


Fig. 5. Comparison of performance with varying number of objects

5 Application of Context Reduction by In-Close2 in Gene Co-expression Analysis

A feature of In-Close2 is to use the well-known idea of minimum support to reduce the size and complexity of formal contexts. A minimum support for both objects and attributes can be specified so that only concepts that satisfy the support will be mined. In-Close2 then outputs a reduced context excluding all objects and attributes that are not part of concepts that satisfy the support. This allows a readable concept lattice to be produced from a large and complex context.

This technique was applied to a data set produced in collaboration with Herriot-Watt University, Edinburgh, Scotland, UK, from the EMAGE Edinburgh Mouse Atlas Project database. The dataset consisted of mouse gene expression data for 6,838 genes in 2,335 mouse tissues (coded as so-called EMAP numbers). There were seven levels of strength of gene expression in the tissues, ranging from ‘not detected’ to ‘strong’. By interpreting a gene as a formal object and a combination of tissue with level of expression as a formal attribute, this data was converted into a formal context using the context creator tool, FcaBedrock. In the context created, In-Close2 detected 208,378 concepts, each representing a gene co-expression. By specifying a minimum support of 14 for genes and 18 for tissue/levels (through a process of trial and error), 13 concepts were detected that satisfied the support, and the context became reduced to 24 objects and 14 attributes. The reduced context is shown in Figure 6.

If a technique such as fault tolerance [13] was applied to this context, so that, for example, the ‘missing’ relation (*Tgfb1*, *EMAP:8146-strong*) was assumed to exist, it could be argued that the context should be approximated to a single concept of gene co-expression; all 24 genes being strongly expressed in all 14

Gene Co-Exp	EMAP:8385-strong	EMAP:8339-strong	EMAP:7371-strong	EMAP:7204-strong	EMAP:8360-strong	EMAP:8359-strong	EMAP:8146-strong	EMAP:7847-strong	EMAP:8389-strong	EMAP:7364-strong	EMAP:7363-strong	EMAP:7749-strong	EMAP:8371-strong	EMAP:8394-strong
	Mapk8ip2	x	x	x	x	x	x	x	x	x	x	x	x	x
Tgfb1	x	x	x	x	x	x		x	x	x	x	x	x	x
Zcchc6	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Brpf3	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Caly	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Bcl9l	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Zc3h18	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Dnajc18	x	x	x	x	x	x	x	x	x	x	x	x	x	
H2-T22	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Colec12	x	x	x	x	x	x		x	x	x	x	x	x	x
Wwp2	x	x	x	x	x	x	x		x	x	x	x	x	x
Emp3	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Tcam1	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Papss2	x	x	x	x	x	x	x	x	x	x		x	x	x
Ubxn10	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Cyt11	x	x	x	x	x	x	x	x	x	x	x	x	x	x
BC024814	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Plekhb1	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Apitd1	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Cebpz	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1110017D15Rik	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Copb1	x	x	x	x	x	x	x		x	x	x	x	x	x
Unc5cl	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Haus4	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Fig. 6. Reduced context of gene expression in mouse tissues

tissues. Alternatively, the ‘missing’ relations could be investigated by examining the original data, where it was found, for example, that there was no record for gene *Tgfb1* in tissue *EMAP:8146* (perhaps indicating that such an experiment has not yet been carried out). On the other hand, a record for gene *Dnajc18* in tissue *EMAP:8349* did exist, stating a ‘moderate’ level of detection; a good enough indication, perhaps, that this gene/tissue pair should be part of this concept of co-expression.

It also became interesting to discover what bones are missing from the co-expression and why. There are bones in the skull for which there were no data recorded for particular genes, but when an image of the embryo from the relevant experiment was subsequently examined, it was clear that these bones *did* have expression in them. Thus, another possible use for this FCA technique is in

inferring the most likely level of expression for a gene-tissue pair when data is missing.

It was also noted that the tissues are mostly bones in the skull and that they are all from the same development stage of the mouse embryo (*Theiler Stage 23*). Further biological investigation is now required to determine the significance of this co-expression; what is happening in the development of the mouse skull at Theiler Stage 23?

6 Conclusion

In-Close2 has been shown to outperform FCbO and AddExtent, apart from where the formal context was a combination of dense ($> 7\%$) and random. In such cases, FCbO was the best performer. This is probably because, although FCbO closes a concept before testing its canonicity, it does it very quickly by using bitwise operations on 32 bit data when performing closure (i.e., 32 comparisons of context table cells are performed at a time). In-Close2 tests each cell individually in its backtracking canonicity test, which avoids having to close a concept before testing its canonicity, but can be slower if the context is random and dense. This is probably because it is more likely that several cells will have common crosses tested before the comparison fails. In real and artificial data sets, density seems to have less of a detrimental effect on In-Close2, probably because of the natural predominance of patterns over random noise, reducing the number of ‘near misses’ when testing canonicity. Further investigation would be required to confirm this hypothesis.

Optimisations have been shown in this paper that significantly improve the performance of In-Close2, by making better use of cache memory. These optimisations are quite general and could be applied to many algorithms that operate on Boolean data. It is not clear from the authors of FCbO the extent to which similar techniques have been used in their implementation of FCbO, although they stated that their implementation was “highly tuned”.

The usefulness of a high-performance concept miner has been shown by the analysis of a large, complex, context of gene expression data using the technique of context reduction though minimum support. A large co-expression of genes in the skull bones of a mouse embryo has been discovered. Uses for FCA in focusing further investigation and inferring missing data have been shown.

Acknowledgments. The help is acknowledged of Kenneth McLeod and Albert Berger of the School of Mathematical and Computer Sciences, Herriot-Watt University, Edinburgh, Scotland, UK, in the preparation and analysis of the mouse gene expression data.

This work is part of the CUBIST project (“Combining and Uniting Business Intelligence with Semantic Technologies”), funded by the European Commission’s 7th Framework Programme of ICT, under topic 4.3: Intelligent Information Management.

References

1. Andrews, S.: In-close, a fast algorithm for computing formal concepts. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS 2009. CEUR WS, vol. 483 (2009), <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-483/>
2. Andrews, S., Orphanides, C.: Analysis of large data sets using formal concept lattices. In: Kryszkiewicz, M., Obiedkov, S. (eds.) [10], pp. 104–115
3. Andrews, S., Orphanides, C.: Fcabledrock, a formal context creator. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 181–184. Springer, Heidelberg (2010)
4. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
5. Goethals, B.: Frequent itemset implementations (fimi) repository (2010), <http://fimi.cs.helsinki.fi/>
6. Hamming, R.W.: Error detecting and error correcting codes. Bell System Technical Journal 29(2), 147–160 (1950)
7. Kaytoue, M., Duplessis, S., Kuznetsov, S.O., Napoli, A.: Two fca-based methods for mining gene expression data. In: Ferré, S., Rudolph, S. (eds.) ICFCFA 2009. LNCS (LNAI), vol. 5548, pp. 251–266. Springer, Heidelberg (2009)
8. Krajca, P., Outrata, J., Vychodil, V.: Parallel recursive algorithm for fca. In: Belohavlek, R., Kuznetsov, S.O. (eds.) CLA 2008 (2008)
9. Krajca, P., Vychodil, V., Outrata, J.: Advances in algorithms based on cbo. In: Kryszkiewicz, M., Obiedkov, S. (eds.) [10], pages 325–337
10. Kryszkiewicz, M., Obiedkov, S. (eds.): 7th International Conference on Concept Lattices and Their Applications, CLA 2010. University of Sevilla, Seville (2010)
11. Kuznetsov, S.O.: Learning of simple conceptual graphs from positive and negative examples. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 384–391. Springer, Heidelberg (1999)
12. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental and Theoretical Artificial Intelligence 14, 189–216 (2002)
13. Pensa, R.G., Boulicaut, J.-F.: Towards fault-tolerant formal concept analysis. In: Bandini, S., Manzoni, S. (eds.) AI*IA 2005. LNCS (LNAI), vol. 3673, pp. 212–223. Springer, Heidelberg (2005)
14. Priss, U.: Fca algorithms (2009), <http://www.upriss.org.uk/fca/fcaalgorithms.html>
15. Tanabata, T., Sawase, K., Nobuhara, H., Bede, B.: Interactive data mining for image databases based on fca. Journal of Advanced Computational Intelligence and Intelligent Informatics 14(3), 303–308 (2010)
16. van der Merwe, D., Obiedkov, S.A., Kourie, D.G.: AddIntent: A new incremental algorithm for constructing concept lattices. In: Eklund, P.W. (ed.) ICFCFA 2004. LNCS (LNAI), vol. 2961, pp. 372–385. Springer, Heidelberg (2004)

A Mapping from Conceptual Graphs to Formal Concept Analysis

Simon Andrews and Simon Polovina

Conceptual Structures Research Group
Communication and Computing Research Centre
Faculty of Arts, Computing, Engineering and Sciences
Sheffield Hallam University, Sheffield, UK
s.andrews@shu.ac.uk, s.polovina@shu.ac.uk

Abstract. A straightforward mapping from Conceptual Graphs (CGs) to Formal Concept Analysis (FCA) is presented. It is shown that the benefits of FCA can be added to those of CGs, in, for example, formally reasoning about a system design. In the mapping, a formal attribute in FCA is formed by combining a CG source concept with its relation. The corresponding formal object in FCA is the corresponding CG target concept. It is described how a CG, represented by triples of the form source-concept, relation, target-concept, can be transformed into a set of binary relations of the form (target-concept, source-concept $\hat{\ } relation) creating a formal context in FCA. An algorithm for the transformation is presented and for which there is a software implementation. The approach is compared to that of Wille. An example is given of a simple University Transaction Model (TM) scenario that demonstrates how FCA can be applied to CGs, combining the power of each in an integrated and intuitive way.$

1 Introduction

Conceptual Graphs (CGs) and Formal Concept Analysis (FCA) are related disciplines in that they both aim to help us understand the world and systems within it by structuring, formalising and depicting their semantics. CGs and FCA have communities of researchers and practitioners that, although independent, share common goals of knowledge discovery and elucidating the meaning of human systems and interactions. Indeed, each community has come to be conversant in the other's discipline, not least through the shared dissemination of their work at the annual International Conference on Conceptual Structures. As a result, CGs and FCA have been linked in several works [1,2,3,4,5]. Although, in these cases, the powers of both conceptual disciplines have been brought to bear on a particular domain, a direct mapping from CGs to FCA is not being attempted. Such a mapping was, however, proposed by Wille to obtain a unified mathematical theory of Elementary Logic [7]. Referring to Wille's translation as our comparison along the way, we present a straightforward mapping from CGs to FCA.

2 Motivation

At Sheffield Hallam University we have established a Conceptual Structures Research Group. One of us (Polovina) has had a long-standing interest in CGs whilst the other (Andrews) has developed an active interest in FCA. Our group has applied CGs through the Transaction Model (TM) [6]. For FCA there is our contribution to the CUBIST project (www.cubist-project.eu). One of our core interests is how we can at a practical level bring CGs and FCA together. For this purpose we took a simple TM example, namely ‘P-H University’ to illustrate the discussion that we now present [6].

3 Conceptual Graphs (CGs)

A Conceptual Graph (CG) is a bipartite graph as shown by the general form in Figure 1 and may be read as: “The relation of a Concept_1 is a Concept_2”.



Fig. 1. General Form of a CG

3.1 Concepts and Relations

The direction of the arcs between a concept and a relation assist the direction of the reading. It distinguishes the source concept from the target concept. Here therefore Concept_1 is the source concept and Concept_2 the target concept. Alternatively to this ‘display’ form (produced using the *CharGer* CGs software, charger.sourceforge.net/), a CG may be written in the following ‘linear’ text-based form:

```
[Concept_1] -> (relation) -> [Concept_2]
```

Consider the following example:

```
[Transaction] -> (part) -> [Cash_Payment]
```

This example will form a part of an illustrative case study involving a fictitious university P-H University. The example graph reads as “The part of a transaction is a cash payment”. This may create readings that may sound long-winded or ungrammatical, but is a useful mnemonic aid in constructing and interpreting any CG. It is easier in this case to state “A cash payment is part of a transaction”. Furthermore, a concept has a referent that refers to the particular instance, or individual, in that concept. For example consider the concept:

```
[Educational_Institution: P-H_University]
```

This reads as “The educational institution known as P-H University”, where P-H University is the referent of the type label Educational Institution in this concept. A concept that appears without an explicit referent has a ‘generic’ referent, thereby referring to an individual that is implicit. Thus for example the concept [Transaction] simply means “A transaction” or “There is a transaction”. It could be that the transaction has a distinct reference number e.g. #tx1 as its referent, resulting in [Transaction: #tx1], if it conforms to that particular transaction.

4 Mapping CGs to FCA

4.1 A CG to FCA Algorithm

The following algorithm takes a Conceptual Graph (CG) in the form of a set of (*SourceConcept*, *Relation*, *TargetConcept*) triples and creates a corresponding set of binary relations of the form (*TargetConcept*, *SourceConcept* \wedge *Relation*) and thereby makes an FCA formal context whose attributes are the *SourceConcept* \wedge *Relation* parts and whose objects are the corresponding *TargetConcept* parts of each binary relation.

The *SourceConcept*, *Relation* and *TargetConcept* parts of a CG triple, t , are denoted by $t.source$, $t.relation$ and $t.target$, respectively. I is the set of (*Object*, *Attribute*) pairs making the FCA formal context, where

$$t \in T \Rightarrow (t.target, t.source \wedge t.relation) \in I$$

The mapping is transitive in that the target CG concept of one relation can become the source CG concept of another relation. This transitivity gives rise to the inference of implicit mappings:

$$\forall t_1, t_2 \in T \bullet t_1.target = t_2.source \Rightarrow (t_2.target, t_1.source \wedge t_1.relation) \in I$$

```

1 begin
2   foreach  $t \in T$  do
3     FormBinaries( $t.target$ ,  $t.target$ );
4 end

```

Fig. 2. *CGtoFCA()*

The main algorithm, *CGtoFCA* (see Figure 2) takes each triple, t , in a set of CG triples, T , and forms all binary relations associated with the target concept of the triple by calling the procedure *FormBinaries*, given in Figure 3. *FormBinaries* takes two CG concepts as arguments: *FixedTarget* and *MovingTarget*. *FixedTarget* is used as a stem target concept so that inferred,

```

1 begin
2   foreach  $t \in T$  do
3     if  $t.target = MovingTarget$  then
4        $Attribute \leftarrow t.source \hat{\ } t.relation;$ 
5        $Object \leftarrow FixedTarget;$ 
6        $I \leftarrow I \cup \{(Object, Attribute)\};$ 
7        $FormBinaries(FixedTarget, t.source);$ 
8 end

```

Fig. 3. $FormBinaries(FixedTarget, MovingTarget)$

transitive, binaries can be formed by backward chaining. This is achieved by setting *MovingTarget* to a source concept that is searched for as a target concept in an inferred relation.

The algorithm works by initially setting the target concept of each triple as both *MovingTarget* and *FixedTarget* and then calling *FormBinaries* to iterate through all triples, forming a corresponding FCA (*Object*, *Attribute*) pair each time *MovingTarget* matches the target concept of a triple. The transitive binaries are formed by recursively calling *FormBinaries*, setting *MovingTarget* to the current source concept and leaving *FixedTarget* unchanged.

To demonstrate the algorithm (and provide a partial proof) it is applied to the simple CG example shown in Figure 4. The corresponding set of triples is given in Table 1.

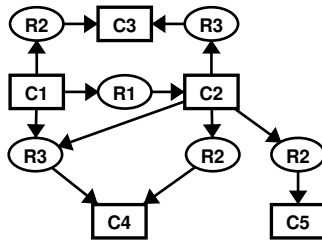


Fig. 4. A Simple Conceptual Graph

Rather than go through all of the triples in the main algorithm, for brevity it is sufficient to show the features of the algorithm by listing a sequence of steps from the call to *FormBinaries* from *CGtoFCA* when $t \leftarrow (C2, R2, C4)$.

```

FormBinaries(C4, C4) //the fixed target is C4, moving target is C4
   $t \leftarrow (C1, R1, C2)$ 
     $C2 \neq C4$ 
   $t \leftarrow (C1, R2, C3)$ 
     $C3 \neq C4$ 

```

```

t ← (C1, R3, C4)
C4 = C4 //the target of the triple matches the moving target
I ← I ∪ {(C4, C1 ∧ R3)} //add FCA (Object,Attribute)
FormBinaries(C4, C1) //the fixed target is C4, moving target is C1
  t ← (C1, R1, C2)
    C2 ≠ C1
  t ← (C1, R2, C3)
    C3 ≠ C1
  t ← (C1, R3, C4)
    C4 ≠ C1
  t ← (C2, R2, C4)
    C4 ≠ C1
  t ← (C2, R2, C5)
    C5 ≠ C1
  t ← (C2, R3, C3)
    C3 ≠ C1
  t ← (C2, R3, C4)
    C4 ≠ C1
  //no targets of triples match C1
//so back up one level to moving target being C4
t ← (C2, R2, C4)
C4 = C4 //the target of the triple matches the moving target
I ← I ∪ {(C4, C2 ∧ R2)}
FormBinaries(C4, C2) //moving target is now C2
  t ← (C1, R1, C2)
    C2 = C2 //the target of the triple matches the moving target
      I ← I ∪ {(C4, C1 ∧ R1)}
      //thus adding an implied FCA binary by backward chaining
      FormBinaries(C4, C1) //already completed above
  
```

A small deficiency of the algorithm is that binaries associated with a target concept may be generated multiple times. Each call to *FormBinaries* from *CGtoFCA* generates all binaries associated with *FixedTarget*, so multiple

Table 1. Concept-Relation-Concept Triples from the Simple Conceptual Graph

source concept	relation	target concept
C1	R1	C2
C1	R2	C3
C1	R3	C4
C2	R2	C4
C2	R2	C5
C2	R3	C3
C2	R3	C4

Simple CG	C1-R1	C1-R2	C1-R3	C2-R2	C2-R3
	C1				
C2	x				
C3	x	x			x
C4	x		x	x	x
C5	x			x	

Fig. 5. The Corresponding Formal Context of the Simple Conceptual Graph

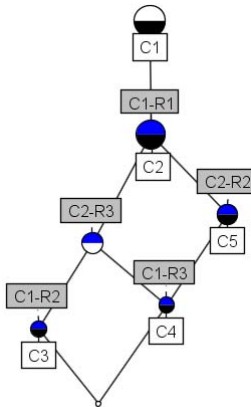


Fig. 6. The Corresponding Concept Lattice for the Simple Conceptual Graph

instances of a target concept means this will happen multiple times. An implementation of the algorithm may need to take this into account (by removing repeated binaries after generation, for example). However, in practical terms, multiple instances of a binary only means that a ‘cross’ is entered into a table cell multiple times.

The algorithm will produce an infinite recursion, generating repeated binaries, if a cycle exists in the CG. The simplest example of this is a CG concept that is both the source and target of a relation, although larger cycles are easily possible. A sensible approach to implementing the algorithm could include the capture of such cycles (by noting that the program has ‘been here before’ in the CG) and reporting them to the user. It may then be useful for the author of the CG to consider how desirable the cycles are in their design.

The resulting binaries generated from the simple CG triples are shown as a cross-table (FCA context) in Figure 5 and hence as an FCA concept lattice in Figure 6. Using well known notions from FCA, a number of observations can be made:

- $C1$ is not a target CG concept of any relation.
- $C1 - R1$ is the relation with the most results (four): $C2, C5, C4$ and $C3$.
- $C4$ results from the most relations (four): $C1 - R1, C2 - R3, C2 - R2$ and $C1 - R3$.

5 A Simple University Scenario Example

We now specify the example. It is a simple case study that is discussed extensively elsewhere to demonstrate the Transaction Model (TM) [6]. Essentially the case study is about P-H University, which is a fictional higher education institution. The University is not primarily a profit-making institution; rather it has to remain financially sound whilst recognising its community objectives. Key to these objectives are its research activities. P-H University thereby needs to explicate the relationship between its research and its community objectives. 40% of its staff are emerging researchers that receive time off for research instead of revenue-generating activities (e.g. teaching), added to which there is a diversion of revenue to give the established researchers' time to support their emerging colleagues. In financial terms these items represent a 'cost' to the University for which there is no corresponding revenue. Yet the financial cost saving from not investing in the emerging research staff's psychological stimulation would undermine the very purpose of the university in meeting its community objectives. To achieve the correct balance, P-H University turns to the TM. The TM shows that psychological stimulation and motivated staff are balanced with its financial obligations to sustain the University's existence.

5.1 The CG for the Example

P-H University's TM is given by Figure 7. From the TM modelled in CG for P-H University we can observe the following:

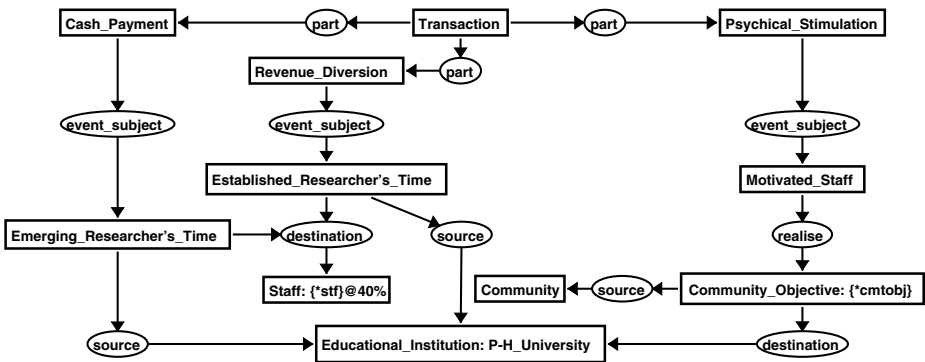


Fig. 7. CG of P-H University Scenario

P-H University's TM										
	Transaction part	Established Researcher's Time source	Psychical Stimulation event subject	Revenue Diversion event subject	Cash Payment event subject	Motivated Staff realise	Emerging Researcher's Time destination	Established Researcher's Time destination	Community Objective: {*cmtobj} destination	Community Objective: {*cmtobj} source
Staff: {*stf}@40%	x			x	x		x	x		
Educational Institution: P-H University	x	x	x	x	x	x			x	x
Community Objective: {*}	x	x				x				
Motivated Staff	x	x								
Revenue Diversion	x									
Emerging Researcher's Time	x				x					
Community	x	x				x			x	
Established Researcher's Time	x		x							
Cash Payment	x									
Psychical Stimulation	x									
Transaction										

Fig. 8. Formal Context of P-H University Scenario

1. The University's TM reveals its validity through the costs being balanced by the benefits to the university achieving its community objectives. As Community Objective refers to *communitites*, in CG it is shown as the plural referent {*cmtobj}.
2. The balancing of these debits and credits denotes the exchange of resources over and above the simple monetary aspects. Thus the qualitative Psychic Enjoyment is as much a part of the transaction as the quantitative Cash Payment.
3. The TM shows that Cash Payment and Revenue Diversion versus Psychical Stimulation are the two complementary sets of economic events that trigger the transaction.
4. The event subject relations (i.e. the states altered by the economic events) point to the relevant economic resources that in this case are the researchers' time and staff motivation.
5. The source and destination relations (i.e. providers and recipients) of the economic resources are the agents in the transaction. These are the educational

institution P-H University and the agents it transacts with, namely its staff and the community it serves.

6. The $\{*\text{stf}\} @ 40\%$ describes a plural of staff, specialised by the $@ 40\%$ thereby denoting the 40% of staff that are supported by emerging researchers time.
7. Motivated Staff is an economic resource of the University in that they add value to the assets of the University by being motivated.

5.2 *CGtoFCA*

Using a software implementation of the *CGtoFCA* algorithm (*CGFCA*, sourceforge.net/projects/cgfca/), and displaying the result using the Concept Explorer software (sourceforge.net/projects/conexp/), Figure 9 shows the FCA lattice for P-H University's TM in CG as shown by Figure 7. The formal context table is given by Figure 8.

As described earlier the CG source concept concatenated with its relation become formal attributes in FCA and the CG target concept becomes a formal object. Thus for example $[\text{Emerging_Researcher's_Time}]^{\wedge}(\text{destination})$ becomes the formal attribute `Emerging_Researcher's_Time destination` and $[\text{Educational_Institution: P-H_University}]$ becomes the formal object `Educational_Institution: P-H_University`.

From the lattice for P-H University we can observe the following:

1. A node's own attribute(s) and its own objects follow the triple structure of CGs i.e. source concept \rightarrow relation \rightarrow target concept. For example:
The `Cash_Payment event_subject` is `Emerging_Researcher's_Time` and `Motivated_Staff realise` `Community_Objective: {*cmtobj}`
2. The formal attributes and objects of a concept automatically 'collect' the the dependencies in the CG. For example:
`Community` is dependent on `Community_Objective: {*cmtobj}` source, `Motivated_Staff realise`, `Psychical_Stimulation event_subject` and `Transaction part`.
3. The dependencies culminate in `Transaction`.

5.3 An Integrated, Interoperable Conceptual Structure

The lattice reveals that each CG concept is dependent on the CG concepts in its formal attributes and for the TM their reliance on the `Transaction` CG concept, which is the epitome of the TM. In simple terms *all* the objects further down from `Transaction part` describe the extent of the transaction; without which the transaction cannot exist and the lattice shows the hierarchical interdependencies of the CG concepts and their relations. The direction of the arcs of the CGs model are preserved in the FCA lattice. As for each concept's referent, they too are preserved in exactly the same way as they appear in the CG.

As the inherent nature of CGs are preserved in *CGtoFCA*, CGs operations such as conformity, projection and maximal join can still be performed and

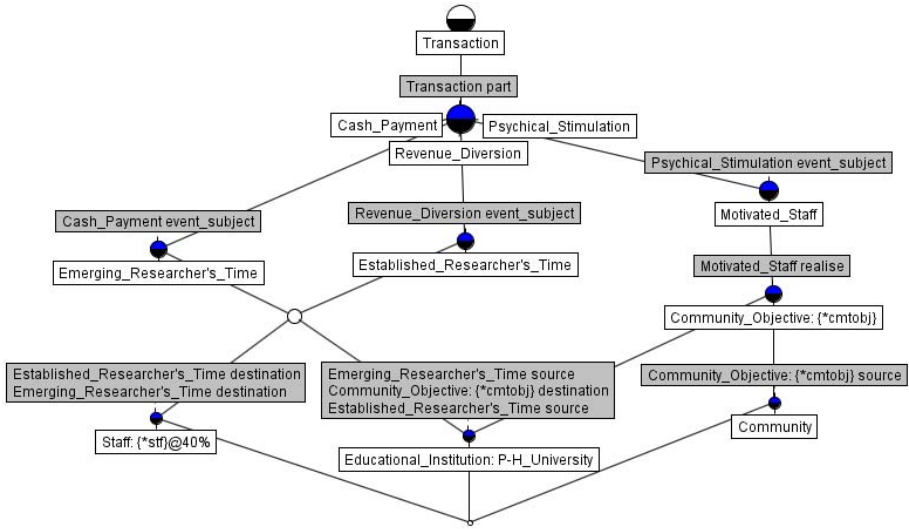


Fig. 9. Concept Lattice of P-H University Scenario

iterated with those of FCA (e.g. attribute exploration). Thus to give one simple scenario, P-H University could model its TM in CGs, exploring the dimensions of its model through such CG operations. Using *CGtoFCA* it could simultaneously translate that TM into an FCA lattice and bring the power of FCA to bear on its TM too. Any enhancements that FCA identified could be translated back into CGs. Models could then be round-tripped between FCA and CGs. Put simply, through *CGtoFCA* we have merged CGs and FCA into a *single* and *interoperable* conceptual structure that provides a superset of operations by combining those of CGs with FCA.

5.4 An Enhanced TM

As a simple illustration let us re-examine Figure 9. Whilst it has identified Transaction as the overarching superconcept, interestingly there is no object identified with the bottommost subconcept. This prompts P-H University to re-examine its CGs TM. It sees that whilst the other concepts in this TM eventually point to it via its arcs and relations in that TM, there are none pointing from [Staff: {*stf}@40%] or [Community]. There should be some explicit relationship that points from these ‘outside’ agents in this transaction to P-H University, which is the ‘inside’ agent given it is P-H University’s TM. Each of these outside agents is a contract party to P-H University. It therefore joins the following CG to its TM:

```
[Educational_Institution: P-H_University]-
(contract_party)<-[Staff: {*stf}@40%]
(contract_party)<-[Community]
```

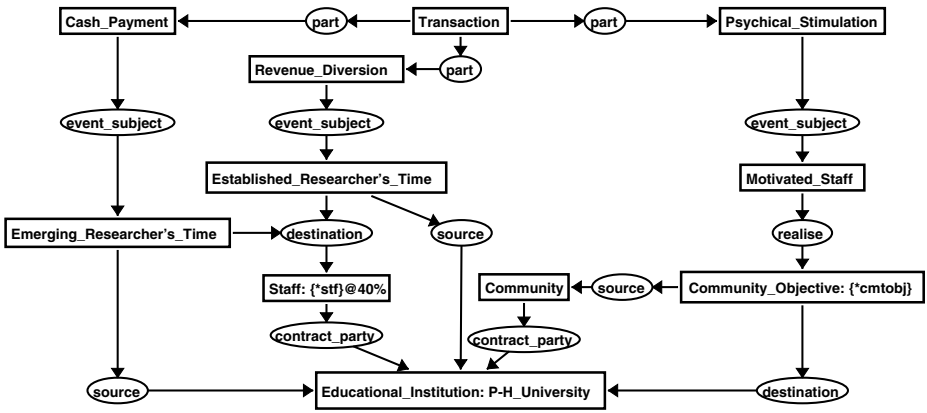


Fig. 10. CG of P-H University Scenario v2

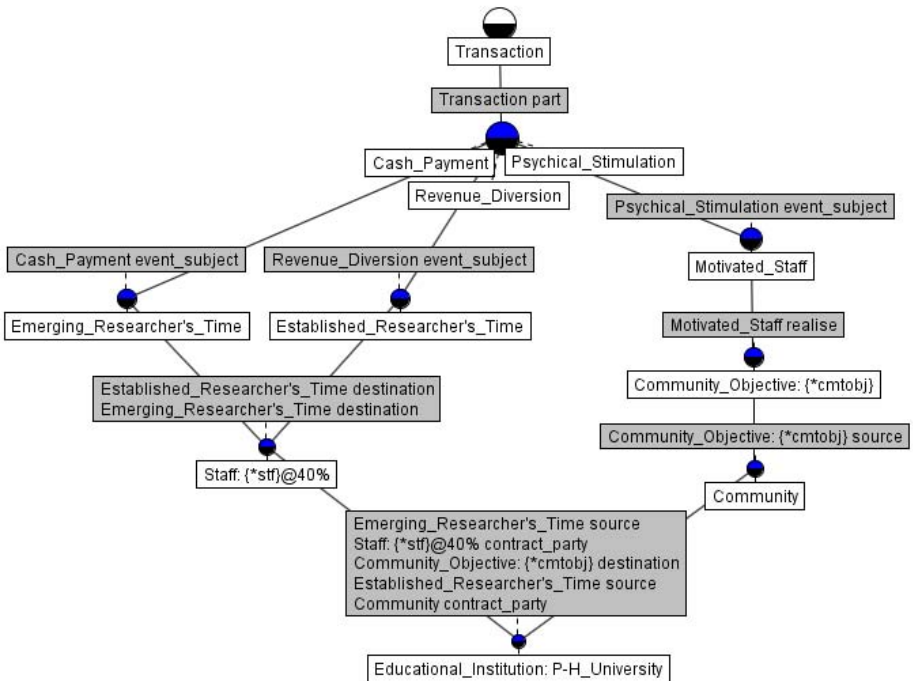


Fig. 11. Concept Lattice of P-H University Scenario v2

The result is the CG TM Figure 10 and the lattice Figure 11. In comparison with Figure 9 we can now observe that the extent of the bottommost concept in Figure 11 is P-H University with the contract party relations duly added in the description of its attributes. Additionally, all the concepts contain own objects. This reveals that all the key concepts have been identified in both the CGs TM and the lattice TM; it transpires that there should not be a formal concept without at least one own attribute and at least one own object. Though such a simple change, it was not obvious in the previous work on the TM using CGs alone 6.

5.5 Wille's CG to FCA Approach

To evaluate the comparative value of *CGtoFCA*, the P-H University's TM in CG as originally depicted by Figure 9 was also converted into an FCA formal context and concept lattice based on the translation given by Wille 7. Wille remarks that CGs capture knowledge at the logical level and FCA adds mathematical rigour, terming it as the 'mathematization' of conceptual structures. We have seen an illustration of the enhancement that FCA brings through the P-H University scenario. It is therefore particularly interesting to peruse Wille's own translation between CGs and FCA. The results according to Wille's translation are shown by Figures 12 and 13 respectively.

As described in the introduction and shown by these figures, Wille's translation essentially takes the referents of CG concepts, as aggregated by the relation that links the concepts. The resulting aggregated referents are then presented as formal objects with their formal attributes as the given relation. This is followed by the source and target concept's type label. As we have seen, the source type label is given the index '1' and the target is indexed as '2'. In line with Wille's translation for each object we have had also to give the referents an explicit identifier e.g. (**#tx1**,**#cp1**) to support the attributes **part**, **Transaction 1** and **Cash Payment 2**.

Wille's translation also touches upon CG type and relation hierarchies in his examples, though co-referent links between concepts. Even allowing for this consideration, which is not replicated for the P-H University example, fundamental to Wille's translation is the choice of ordered pairs of instances of CG concepts as formal objects and the choice of both CG relations and CG concepts as formal attributes. Although the resulting concept lattice provides useful insights into the underlying CGs, it can be argued that this approach leads on the one hand to lists of object pairs that share the same relation but on the other hand lead to rather more complex lattices that show a hierarchy of separated out CG source and target concepts and relations. This is evidenced by the outcomes demonstrated by Figures 12 and 13 for P-H University. The elegant interdependence and simpler lattices as shown in Figure 9 is not as easily discerned and the insight that lead to Figure 11 is not evident. Notably, Transaction as the key concept in the TM is not at the head of the lattice. For the TM it is the relationship of other concepts to the Transaction concept that is the nub of the TM as we have seen.

P-H University's TM	part	event subject	source	destination	realise	Transaction 1	Cash Payment 1	Cash Payment 2	Emerging Researchers Time 1	Emerging Researchers Time 2	Educational Institution 2	Revenue Diversion 1	Revenue Diversion 2	Established Researchers Time 1	Established Researchers Time 2	Staff 2	Psychical Stimulation 1	Psychical Stimulation 2	Motivated Staff 1	Motivated Staff 2	Community Objective 1	Community Objective 2
(#tx1,#cp1)	×				×		×															
(#tx1,#ps1)	×				×													×				
(#tx1,#rd1)	×				×							×										
(#cp1,#emrt1)		×					×		×													
(#rd1,#esrt1)		×									×			×								
(#ps1,#ms1)		×															×		×			
(#emrt1,P-H University)			×						×	×												
(#esrt1,P-H University)			×								×			×								
(#ms1,{*cmtobj})					×														×		×	
({*cmtobj},#cmt1)			×																		×	×
(#emrt1,#esrt1,{*stf}@40%)				×					×					×	×							
({*cmtobj},P-H University)				×																	×	

Fig. 12. Formal Context of P-H University Scenario after Wille

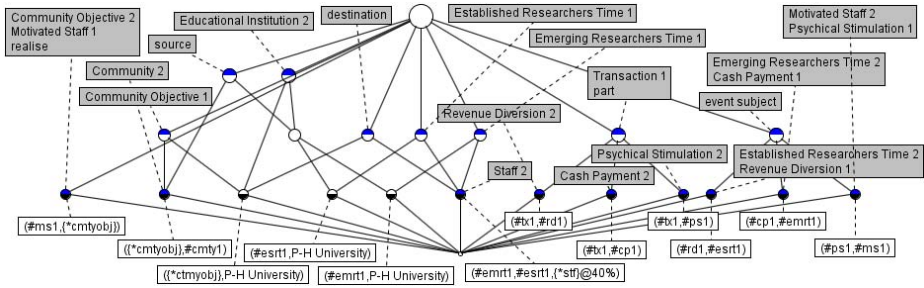


Fig. 13. Concept Lattice of P-H University Scenario after Wille

6 Conclusions

We have demonstrated *CGtoFCA* as a straightforward mapping from Conceptual Graphs (CGs) to Formal Concept Analysis (FCA). CGs and FCA can thus interoperate in a practical way, combining their power in an integrated and intuitive conceptual structure. The simple case study (P-H University) has shown an enhancement to the Transaction Model (TM) modelled in CGs alone. It thus

illustrates the benefits of *CGtoFCA*, which will spur the wider development of conceptual structures' and their practical applications.

References

1. Dau, F., Klinger, J.: From Formal Concept Analysis to Contextual Logic. In: Ganter, B., Stumme, G., Wille, R. (eds.) *Formal Concept Analysis*. LNCS (LNAI), vol. 3626, pp. 81–100. Springer, Heidelberg (2005)
2. Delugach, H., Lampkin, B.: Troika: Using Grids, Lattices and Graphs in Knowledge Acquisition. In: Stumme, G. (ed.) *Working with Conceptual Structures: Contributions to ICCS 2000*, pp. 201–214. Shaker Verlag, Aachen (2000)
3. Ganter, B., Rudolph, S.: Formal Concept Analysis Methods for Dynamic Conceptual Graphs. In: Delugach, H., Stumme, G. (eds.) *ICCS 2001*. LNCS (LNAI), vol. 2120, pp. 143–156. Springer, Heidelberg (2001)
4. Kuznetsov, S.O.: Learning of Simple Conceptual Graphs from Positive and Negative Examples. In: Żytkow, J.M., Rauch, J. (eds.) *PKDD 1999*. LNCS (LNAI), vol. 1704, pp. 384–391. Springer, Heidelberg (1999)
5. Mineau, G., Stumme, G., Wille, R.: Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis. In: Tepfenhart, W., Cyre, W. (eds.) *ICCS 1999*. LNCS, vol. 1640, pp. 423–441. Springer, Heidelberg (1999)
6. Polovina, S., Hill, R.: A Transactions Pattern for Structuring Unstructured Corporate Information in Enterprise Applications. *International Journal of Intelligent Information Technologies* 5(2), 34–47 (2009)
7. Wille, R.: Conceptual Graphs and Formal Concept Analysis. In: Lukose, D., Delugach, H.S., Keeler, M., Searle, L., Sowa, J.F. (eds.) *ICCS 1997*. LNCS, vol. 1257, pp. 290–303. Springer, Heidelberg (1997)

Partial Orders and Logical Concept Analysis to Explore Patterns Extracted by Data Mining

Peggy Cellier¹, Sébastien Ferré², Mireille Ducassé¹, and Thierry Charnois³

¹ IRISA/INSA Rennes

² IRISA/University of Rennes

Campus Beaulieu, F-35043 Rennes Cedex, France

`firstname.lastname@irisa.fr`

³ GREYC/University of Caen

Campus Côte de Nacre, F-14032 Caen cedex, France

`thierry.charnois@unicaen.fr`

Abstract. Data mining techniques are used in order to discover emerging knowledge (patterns) in databases. The problem of such techniques is that there are, in general, too many resulting patterns for a user to explore them all by hand. Some methods try to reduce the number of patterns without a priori pruning. The number of patterns remains, nevertheless, high. Other approaches, based on a total ranking, propose to show to the user the top-k patterns with respect to a measure. Those methods do not take into account the user's knowledge and the dependencies that exist between patterns. In this paper, we propose a new way for the user to explore extracted patterns. The method is based on navigation in a partial order over the set of all patterns in the Logical Concept Analysis framework. It accommodates several kinds of patterns and the dependencies between patterns are taken into account thanks to partial orders. It allows the user to use his/her background knowledge to navigate through the partial order, without a priori pruning. We illustrate how our method can be applied on two different tasks (software engineering and natural language processing) and two different kinds of patterns (association rules and sequential patterns).

Keywords: data mining, partial order, selection of patterns, logical concept analysis, formal concept analysis.

1 Introduction

Knowledge Discovery in Databases (KDD) [FPSS96] can be seen as a process in three steps: preparation of data, data mining and exploitation of extracted patterns. In the second step of the KDD process, data mining techniques are used in order to discover emerging knowledge (patterns) in databases. That step highlights regularities and tendencies which can give important information to a user about the data. The problem for a practical use is that, in general, too many patterns are generated. It is not easy for a user to explore by hand a large amount of patterns. The problem is not specific to one kind of patterns. Indeed,

a huge amount of association rules [AIS93] but also sequential patterns [AS95] or graph patterns can be extracted with data mining techniques.

In order to address this problem, some methods try to reduce the number of patterns without a priori pruning, for example condensed representation [PBT99, PC09] or constraints [PHL01]. The number of patterns remains, nevertheless, high. Other approaches, based on a total ranking [KB10], propose to show to a user the top-k patterns with respect to a specific measure. User's knowledge and dependencies between patterns are not taken into account by that kind of methods.

In this paper we propose an application of Logical Concept Analysis (LCA) [FR04] to build a generic framework to explore patterns extracted by data mining techniques. The framework is based on a data structure which organizes the set of patterns, and provides operations on that structure, namely navigation in the set of patterns, selection of patterns of interest and pruning off patterns without interest. The data structure is given in the form of *Hasse diagram* [DP90], exploiting the fact that patterns are naturally partially ordered. Indeed, some patterns are *sub-patterns* of others. The operations take advantage of the power of LCA. As LCA can be applied to any ordering, its navigation capabilities can be re-used as such. Furthermore, the operations of selection of patterns of interest and pruning off patterns without interest can be straightforwardly implemented on top of its updating capabilities. We illustrate how our framework can be instantiated on two different tasks (natural language processing (NLP) and fault localization) and two different kinds of patterns (sequential patterns and association rules). The tasks had ad hoc formalizations [CDFR08 [CC10] which are unified and generalized by the framework proposed in this paper.

The contribution of the paper is twofold. Firstly, as opposed to existing approaches, users can benefit from their background knowledge to navigate through the patterns until their goal(s) have been reached, without a priori pruning. Secondly, the framework is generic, there is no constraint on the kind of patterns and it can accommodate several kinds of tasks. The genericity is mainly due to the power of LCA. Indeed, LCA can be applied to any ordering of patterns. Whereas Formal Concept Analysis (FCA) [GW99] can also be used on partial ordering, it is tied to sets of attributes ordered by inclusion. Furthermore, in LCA partial orders can be combined with other logics allowing a rich description of patterns. For example, the extracted patterns often have some information about statistical measures such as a support value or a confidence value.

In the remaining of the paper, Section 2 presents the case studies. Section 3 gives background knowledge about LCA. Section 4 defines the proposed approach and Section 5 discusses related work.

2 Case Studies

In this section, we describe two different tasks. Those case studies are used in the paper to illustrate the theory (Section 4).

2.1 Natural Language Processing Task

The first application is a Natural Language Processing (NLP) task [CC10]. Some linguistic patterns are automatically extracted from a corpus. The linguistic patterns have to recognize *appositive qualifying phrases* in French texts. Some examples of appositive qualifying phrases are: “*En bon père de famille,*” (“As a good father;”), “*, connu pour sa cruauté,*” (“, known for his cruelty;”). Several parts of different sentences representing appositive qualifying phrases are collected to build a training corpus. In the training corpus, each appositive qualifying phrase is replaced by a sequence where each word is associated to part-of-speech information. For example, $\langle\langle en\ en\ PRP \rangle\rangle (bon\ bon\ ADJ) (père\ père\ NOUN) (de\ de\ PRP) (famille\ famille\ NOUN)\rangle$ is a sequence¹.

From the training corpus, patterns are extracted. The patterns are *closed sequential patterns under constraints*. For instance, $\langle\langle champion\ NOUN \rangle\rangle (PRP)$ is a sequential pattern that describes phrases starting by a noun whose lemma is “champion” and followed by a preposition (*PRP*). The *support* of a sequential pattern S in a corpus is the number of sequences of the corpus matching S . The *frequent* sequential patterns are the sequences with a support greater than a threshold. An extracted sequential pattern, S_1 , is *closed* if there is no other extracted sequential pattern, S_2 , such that S_1 is included in S_2 and $sup(S_1) = sup(S_2)$. The advantage of closed sequential patterns is the reduction of redundancy. A preliminary automatic filtering is done with the application of two constraints in order to keep only closed sequential patterns that are relevant for the task: **no gap** in patterns (i.e. between itemsets of sequential patterns) and patterns represent **the beginning** of the appositive qualifying phrases. The number of patterns is high (1 789 patterns). The goal of the user is to identify interesting linguistic patterns among extracted sequential patterns.

2.2 Fault Localization Task

The second application is a fault localization task [CDFR08]. When the result of a program execution is not the same as the expected one, that execution is called a *failure*. Fault localization is a software engineering task that tries to find an explanation to the failures by examining information from the executions. To each execution is associated an *execution trace* that contains information about the execution: the executed lines and the verdict of the execution (*Pass* when the result of the execution is the same as the expected one, otherwise *Fail*).

From the execution traces of different executions of a given program, particular association rules are computed where the conclusion is set to *Fail*. For example, the rule “ $r_2 = 78, \dots, 81, 84, 87, 90 \rightarrow Fail$ ” means that “when the lines 78, ..., 81, 84, 87 and 90 are executed, most of the time it implies a failure”. In order to measure the relevance of the rules, the *support* and the *lift* values are also computed. The support measures the number of execution traces that

¹ Each word is replaced by three elements : the word itself, its lemma and grammatical information. Sometimes the word and its lemma are identical. *ADJ* means “adjective” and *PRP* means “preposition”.

execute the lines of the premise of the rule and fail. The lift value measures how the observation of the premise in an execution trace increases the probability that this execution fails. Some rules are identified as “failure rules”. A rule r is a failure rule if there exist some failed executions that contain the whole premise of r in their trace but not the whole premise of rules more specific than r .

The number of extracted rules can be high. The goal of the user is to give at least one explanation for each failure.

3 Logical Concept Analysis (LCA)

Logical Concept Analysis (LCA) is defined in [FR04]. It is a general theory allowing extensions of Formal Concept Analysis (FCA) [GW99] to be easily specified in a formal way. In LCA the description of an object is a logical formula instead of a set of attributes as in FCA. Pattern structures [GK01] are an equivalent alternative to LCA, where “patterns” are used instead of formulas.

3.1 Logic and Partial Order

Definition 1 (logic). *A logic is a lattice $\mathcal{L} = (L, \sqsubseteq, \sqcap, \sqcup, \top, \perp)$ where*

- L is the language of formulas,
- \sqsubseteq is the subsumption relation (the order on the formulas),
- \sqcap and \sqcup are, respectively, the lower bound and the upper bound,
- \top and \perp are, respectively, the top and the bottom of the lattice.

Let f and g be two formulas, i.e. $f, g \in L$, if $f \sqsubseteq g$ and $g \sqsubseteq f$ then f and g are said *logically equivalent*. It is denoted by $f \equiv g$. Some logics are partially defined, *partial logic*, namely the lower and upper bounds are not always defined. The definition of a logic is left very abstract. This makes it possible to accommodate non-standard types of logics. For example, \mathcal{L}_{base} is the logic that describes base domains, e.g. $support = 3 \sqsubseteq support \geq 2$. The subsumption relation also allows the terms of a taxonomy to be ordered (see line attributes in the fault localization illustration in Section 4.5).

We define a partial order, \mathcal{P} , as a couple (P, \leq) where P is a set and \leq is a binary relation on P that is reflexive, anti-symmetric and transitive [DP90]. In the LCA framework, we can define a logic associated to a partial order thanks to *logic functors* (see [FR04] for details). There are several logic functors. $\mathcal{F}_{POSET}(P)$ is the functor that builds a partial logic from a partial order, $\mathcal{P} = (P, \leq)$, such that $p_1 \sqsubseteq p_2$ if $p_1 \leq p_2$. \mathcal{F}_{UNION} , also denoted by \cup , is the functor that combines several logics into a logic, potentially partial. \mathcal{F}_{LIS} is the functor that builds a well-defined logic from a partial logic by adding boolean connectors (“and”, “or” and “not”) and the closed world assumption. Indeed, the “and” and “or” connectors guarantee the lower and upper bound of the logic.

3.2 Logical Context

Definition 2 gives the definition of a logical context in the LCA framework. Definition 3 defines the logical versions of *extent* and *intent*. The extent of a logical formula f is the set of objects in \mathcal{O} whose description is subsumed by f . The intent of a set of objects O is the most precise formula that subsumes all descriptions of objects in O . Definition 4 gives the definition of a *logical concept*.

Definition 2 (logical context). A logical context is a triple $(\mathcal{O}, \mathcal{L}, d)$ where \mathcal{O} is a set of objects, \mathcal{L} is a logic and d is a mapping from \mathcal{O} to \mathcal{L} that describes each object by a formula.

Definition 3 (extent, intent). Let $\mathcal{K} = (\mathcal{O}, \mathcal{L}, d)$ be a logical context. The definition of the extent is: $\forall f \in \mathcal{L}, \text{ext}(f) = \{o \in \mathcal{O} \mid d(o) \sqsubseteq f\}$. The definition of the intent is: $\forall O \subseteq \mathcal{O}, \text{int}(O) = \bigsqcup_{o \in O} d(o)$.

Definition 4 (logical concept). Let $\mathcal{K} = (\mathcal{O}, \mathcal{L}, d)$ be a logical context. A logical concept is a pair $c = (O, f)$ where $O \subseteq \mathcal{O}$, and $f \in \mathcal{L}$, such that $\text{int}(O) \equiv f$ and $\text{ext}(f) = O$. O is called the extent of the concept c , i.e. ext_c , and f is called its intent, i.e. int_c .

The set of all logical concepts is ordered and forms a *lattice*: let c and c' be two concepts, $c \leq c'$ iff $\text{ext}_c \subseteq \text{ext}_{c'}$. Note that $c \leq c'$ iff $\text{int}_c \sqsubseteq \text{int}_{c'}$. Concept c is called a **sub-concept** of c' .

4 The Proposed LCA Framework to Navigate into the Set of Extracted Patterns

In this section, we present the general framework and show how it can be instantiated for the case studies presented in Section 2. Firstly, some pre-requisite to use the method are presented (Section 4.1). Secondly, the logical context is defined and an example with the NLP task is given (Section 4.2). Thirdly, the user actions are presented (Section 4.3) and a stopping criterion is defined (Section 4.4). Then a complete example is given with the fault localization task (Section 4.5). Finally, there is a discussion about the proposed approach (Section 4.6).

4.1 Preliminaries

In our method there are three important parameters: the patterns, the user and the goal of the user. The first hypothesis is that the patterns are already extracted. We do not make any assumption about the kind of patterns or about the extraction technique. The only one pre-requisite is the definition of a partial order over the set of patterns. Note that there is a natural order between patterns: the inclusion order. The second hypothesis is that the user is a domain expert. This means that he/she can judge the relevance of any individual pattern with sufficient information. The proposed method is designed to help a user to understand patterns using his/her background knowledge. If the user is not

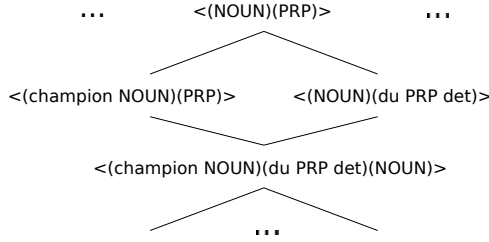


Fig. 1. Excerpt of the partial order on the patterns extracted from a corpus. The most general patterns are at the top.

an expert, the method cannot provide much help. The last hypothesis is that the goal of the user is clearly expressed as a subset of patterns that have to be identified. This hypothesis is important to define a relevant stopping criterion (see Section 4.4).

4.2 A Logical Context to Explore Extracted Patterns

Partial Order. The patterns are naturally partially ordered. Indeed, some patterns are more general than others: *sub-patterns*. Note that constraints can provide other, possibly more relevant, partial orders.

For example, Definition 5 gives the partial order on the patterns, \mathcal{P}_{seq} , for the NLP task. As mentioned in Section 2.1, the patterns used for that task are closed sequential patterns under two constraints. Figure 1 shows a part of that partial order. We see that $\langle\langle champion\ NOUN \rangle\rangle(PR P)$ is more specific than pattern $\langle\langle NOUN \rangle\rangle(PR P)$. Indeed, all phrases that match $\langle\langle champion\ NOUN \rangle\rangle(PR P)$ also match $\langle\langle NOUN \rangle\rangle(PR P)$, but the converse is not true. The phrase “*Champion du monde*” (“Champion of the world”) matches both patterns, but “*Gagnant du concours*” (“Winner of the contest”) only matches $\langle\langle NOUN \rangle\rangle(PR P)$. From that partial order, a logic is derived $\mathcal{L}_{seq} = \mathcal{F}_{POSET}(\mathcal{P}_{seq})$.

Definition 5 (\mathcal{P}_{seq}). Let \mathcal{P}_{seq} be a couple (P_{seq}, \leq_{seq}) such that:

- P_{seq} is all extracted closed sequential patterns that check the constraints,
- Let $l = \langle I_1 \dots I_n \rangle$ and $l' = \langle J_1 \dots J_m \rangle$ be two patterns of P_{seq} then $l \leq_{seq} l'$ if $m \leq n$ and $\forall i \in 1..m\ J_i \subseteq I_i$.

Pattern Context. From the extracted patterns and their associated partial order, a logical context is defined. That context is called *pattern context*. Definition 6 defines pattern contexts by instantiating Definition 2. In this context the objects are identifiers of the extracted patterns. Each pattern is described by the pattern itself. That part of the description is unique for each pattern and mandatory. In addition, the pattern description can contain additional optional information, for example statistical measures (e.g., support, lift). The concept lattice of a *pattern context* represents the search space for exploring the information about patterns.

Definition 6 (pattern context). Let $\mathcal{P} = (P, \leq)$ be a partial order over the pattern set. The associated Pattern Context is a triple $\mathcal{K}(\mathcal{P}) = (\mathcal{O}_p, \mathcal{L}_p, d_p)$ where

- \mathcal{O}_p are the identifiers of the patterns of P ,
- $\mathcal{L}_p = \mathcal{F}_{LIS}(\mathcal{F}_{POSET}(\mathcal{P}) \cup \mathcal{L}_{base})$,
- Let $p \in P$, the description of p is a conjunction (defined in \mathcal{F}_{LIS}) of p and optional additional information about p (defined in \mathcal{L}_{base}).

Table 1. Excerpt of pattern context for the natural language processing task

Pattern ID	\mathcal{P}_{seq}	Add. information: support
p_1	$\langle\langle(NOUN)(PRP)\rangle\rangle$	805
p_2	$\langle\langle(champion\ NOUN)(PRP)\rangle\rangle$	106
p_3	$\langle\langle(NOUN)(du\ PRP\ det)\rangle\rangle$	187
p_4	$\langle\langle(champion\ NOUN)(du\ PRP\ det)(NOUN)\rangle\rangle$	94
...		

Table 1 gives an example of *Pattern Context* for the NLP task. The objects are the identifier of frequent closed sequential patterns. Each line describes a pattern (in \mathcal{P}_{seq}) and the associated support value (in \mathcal{L}_{base}). The support values come from a previous data mining step.

4.3 User Actions: Navigation and Updating

The lattice defined in the previous section can be very large and cannot be displayed. We propose to navigate through the lattice thanks to a LCA tool such as Camelis² [Fer09] and Abilis³ [AFR10]. They allow a user to navigate in logical contexts and to update them. They do not compute the whole lattice a priori but compute parts of the lattice on demand when relevant to the navigation. Figure 2 shows Camelis with the NLP context⁴.

In LCA tools, the interface has three main parts. At the top, the *query view* displays the current query. In Figure 2 the query is “**support** \geq 2”, it means that only patterns whose support is greater than 2, are displayed. At the bottom left hand part, the *navigation tree* displays the features of the navigation (the patterns themselves and additional information). The number next to a feature is the number of patterns that have that feature in their description. For example, 198 patterns have the feature $\langle\langle(NOUN)(PRP)\rangle\rangle$ in their description, namely 198 patterns are more specific than the pattern $\langle\langle(NOUN)(PRP)\rangle\rangle$. We see the patterns from the excerpt of \mathcal{P}_{seq} of Figure 1 (underlined patterns). On the right hand part, the *pattern view* displays all patterns whose description is subsumed by the query. With respect to the query view, only patterns having a support equal or greater than 2 are shown there.

² <http://www.irisa.fr/LIS/ferre/camelis>

³ <http://ledenez.insa-rennes.fr/abilis/>

⁴ Annotations in bold red have been added for this paper.

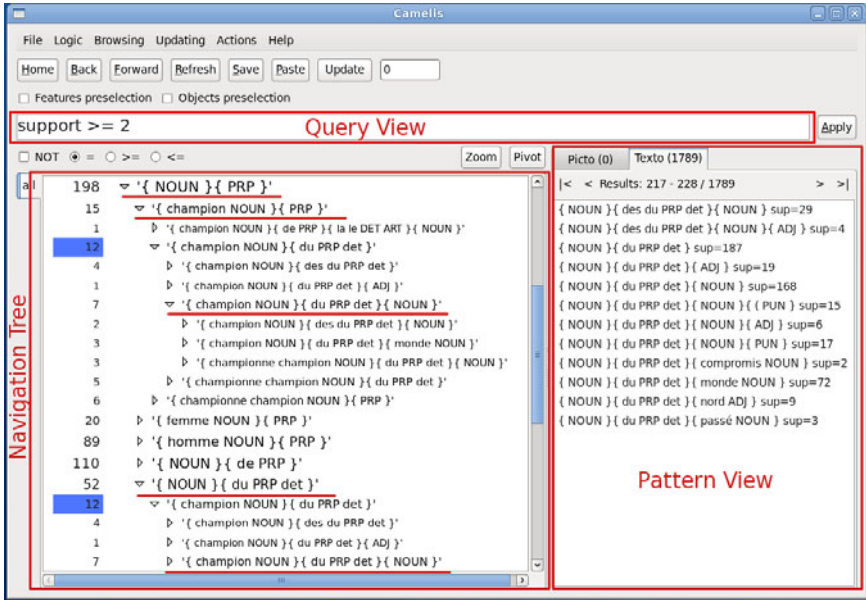


Fig. 2. Camelis with the NLP context

Navigation. The user can navigate through different kinds of attributes (patterns and additional information). The flexibility in the navigation comes from the logics. Indeed, thanks to the combination of logics (logic functors), the user can create queries that mix elements of the partial order and additional information such as support values.

For the NLP task, the user explores the patterns from the most general ones, which are matched by a lot of phrases (e.g., $\langle\langle champion NOUN \rangle\rangle(PRP)$ matched by 805 phrases), to the most specific patterns, that are matched by less phrases (e.g., $\langle\langle champion NOUN \rangle\rangle(du PRP det)(NOUN)$ ⁵ matched by 94 phrases). The partial order over the set of patterns is highlighted in the navigation tree. Note that behind the so called navigation tree, there is not a tree structure but a partial order (\mathcal{P}_{seq}). It explains the fact that the pattern $\langle\langle champion NOUN \rangle\rangle(du PRP det)(NOUN)$ appears twice in the navigation tree. Indeed, it is the same pattern that has two parents.

Context Updating to Select and Prune Patterns. When exploring the patterns the user may add some information about the patterns by adding some features to their description. The two main advantages are that it permits to build a result set with selected patterns and to prune patterns without interest, i.e. patterns already selected or patterns not interesting for the purpose. If a pattern p is selected, all more specific patterns than p do not have to be explored,

⁵ *det* means “determiner”.

they are subsumed by p . In the same way, if a pattern p is away from the point, all more specific patterns than p are away from the point and do not have to be explored. Therefore, when a pattern is tagged, all more specific patterns are also tagged and the search space is pruned. To facilitate the navigation and to reduce the search space, we propose to create a taxonomy of tags. All tags required to update the descriptions of patterns are subsumed by a general tag: **Tags**. When navigating, the user adds to the query **not Tags** in order to eliminate patterns already tagged from the views.

For instance, for the NLP task there are two tags in **Tags**: **LinguisticPattern** and **NotLinguisticPattern**. Those tags are used in two cases. The first case is when the user finds a pattern, p , really interesting to recognize appositive qualifying phrases. The user selects p and the query becomes “ p ”. In the pattern view, all patterns that are more specific than p are thus displayed. Then the user adds the tag **LinguisticPattern** to all patterns displayed. Thus, p and all more specific patterns than p are tagged as **LinguisticPattern**, i.e. they belong to the resulting set of linguistic patterns. In order to avoid to explore those patterns again, the user has just to add **not Tags** to the current query. The second case is when the user finds a pattern, p , clearly not relevant to recognize appositive qualifying phrases. The user selects p and the query becomes “ p ”. In the pattern view, all patterns that are more specific than p are thus displayed. Then the user adds the tag **NotLinguisticPattern** to all patterns displayed. Thus, p and all more specific patterns are tagged as **NotLinguisticPattern**. As previously, thanks to the “**not Tags**” query, the patterns already labelled are pruned from the navigation space of the user.

4.4 Stopping Criterion

When the user can clearly define a goal as a set of patterns to label, P_{goal} , a stopping criterion is provided (Property [1](#)). The user specifies his/her goal by adding a specific feature to the description of the goal patterns.

Property 1. Let P be a set of patterns. Let $P_{goal} \subseteq P$ be the user goal. The process stops when all elements of P_{goal} are labelled.

Let goal be the feature that enables to tag a pattern as being in the user goal. Assuming (hypothesis 2) that the user is competent, every time he identifies a pattern in the goal he tags it accordingly. When query “**not Tags and goal**” has no answer, the process ends. The advantage is that users constantly know without any effort how much of the information they still have to investigate; another advantage is that users do not need to explore the whole set of patterns even if the goal is the whole set of patterns.

4.5 The Fault Localization Example

As presented in Section [2.2](#), the goal of the fault localization is to understand why a program fails. In this section, we show how the proposed approach can be instantiated to that task.

Table 2. Excerpt of pattern context for the fault localization task

Pattern ID	Pattern	Add. Information												
		support	lift	Failure	line					inv_line				
					81	90	93	101	...	81	90	93	101	...
r_1	$81, 90, 93, \dots \rightarrow Fail$	60	1.48		X	X	X		...		X			...
r_2	$81, 93, \dots \rightarrow Fail$	112	1.42		X		X		...	X	X		X	...
r_3	$81, 93, 101, \dots \rightarrow Fail$	52	1.36		X		X	X	...				X	...
r_4	$93, \dots \rightarrow Fail$	112	1.35				X		...	X	X	X	X	...
...														

Partial Order. The first step is the definition of the partial order over the association rules extracted from execution traces (Definition 7). All association rules have the same conclusion: *Fail*. The partial order is thus derived from the inclusion on the premise of the rules. A rule, r , is more specific than another one, r' , if the premise of r' is included into the premise of r . For example, $r_1 = 81, 90, 93 \rightarrow Fail \leq_{ar} r_2 = 81, 93 \rightarrow Fail$.

Definition 7 (\mathcal{P}_{ar}). Let $\mathcal{P}_{ar} = (P_{ar}, \leq_{ar})$ be a partial order where:

- P_{ar} is the set of extracted association rules;
- Let $r = L \rightarrow Fail$ and $r' = L' \rightarrow Fail$ be two association rules of P_{ar} then $r \leq_{ar} r'$ if $L' \subseteq L$.

Pattern Context. Table 2 gives an excerpt of a pattern context for the fault localization task. The pattern context associated to the fault localization task, describes each rule, r , by: (1) r , the rule itself; (2) the support value and (3) the lift value that are computed during the data mining step; (4) the **Failure** attribute if r is a failure rule; (5) the lines that belong to the premise of the rule (**line**); (6) the lines that belong only to the premise of r or more specific rules than r (**inv_line**).

The elements of **line** form a taxonomy on the lines. That taxonomy is described by the partial order defined in Definition 8. At the top of the taxonomy, there are the lines that are specific to the most general rules. At the bottom, there are the lines that are specific to the most specific rules. The elements of **inv_line** form the inverted taxonomy of **line**, $\mathcal{P}_{inv_line} = (P_{line}, \geq_{line})$. The logic of the pattern context can thus be summarized by:

$$\mathcal{L}_p = \mathcal{F}_{LIS}(\mathcal{F}_{POSET}(\mathcal{P}_{ar}) \cup \mathcal{F}_{POSET}(\mathcal{P}_{line}) \cup \mathcal{F}_{POSET}(\mathcal{P}_{inv_line}) \cup \mathcal{L}_{base}).$$

Definition 8 (\mathcal{P}_{line}). Let \mathcal{P}_{line} be a couple (P_{line}, \leq_{line}) such that:

- P_{line} is all lines of the program;
- Let l and l' be two lines of P_{line} then $l \leq_{line} l'$ if all association rules that contain l' also contain l .

For instance, r_2 which is not a failure rule, represents the 112 failed executions that execute lines 81, 93, ... of the program. $r_1 \leq_{ar} r_2$ and $r_3 \leq_{ar} r_2$, r_2 has thus

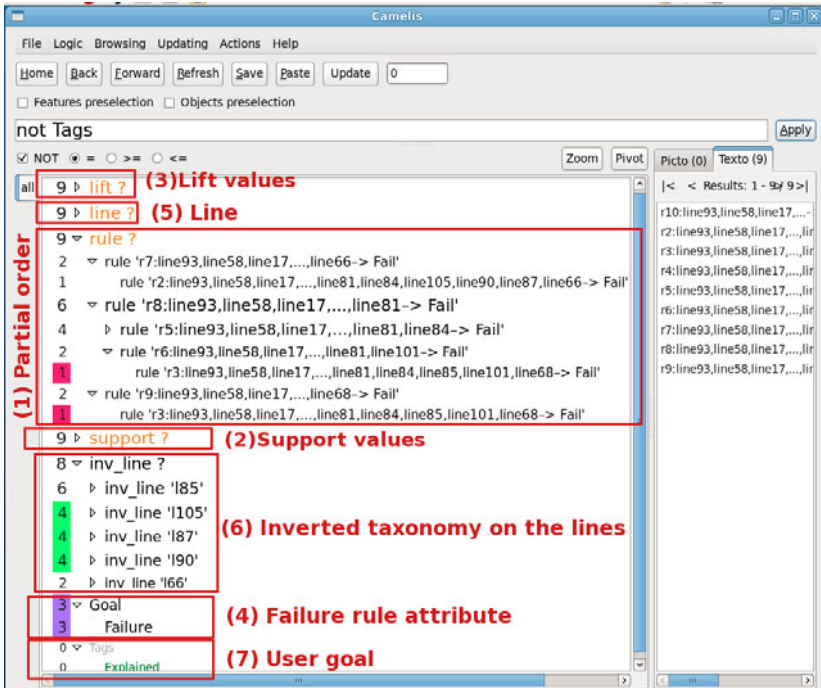


Fig. 3. Camelis with the fault localization context

the specific lines of r_1 and r_3 in its inverted line description, i.e. 90 and 101, plus its specific line 81. Figure 3 shows Camelis with the fault localization context and highlights the different elements of rule descriptions.

Navigation. In order to understand the behaviour of the program when it fails, the user starts by checking the lines that are specific to failure rules. When this is not sufficient to understand why the program fails, the user checks specific lines of a more general rule and so on. It means that the user checks the lines of the patterns in the order of the inverted line taxonomy.

In order to navigate from the lines that specifically belong to the failure rules, the user can select **Failure** in the navigation tree. Then only failure rules appear in the pattern view. The user can then choose one rule among them and display the intent of the rule which is its description. The query is now the description of the selected rule. That description shows the most specific lines describing the rule. The user can select one of those lines as the starting point of the navigation and explore the lines through the inverted order (*inv_line*).

Updating. For the fault localization task, there is only one tag: **Explained** (7). The user explores the lines in the reverse order (*inv_line*). When the user sees a line, *inv_line:l*, which allows him/her to understand a failure, he/she adds the attribute **Explained** to all rules that have *line:l* in their description. Consequently, those rules are no more interesting for the exploration, they are

explained by l . The user can add **not Tags** in the query to avoid seeing the association rules that are already “explained”.

User Goal and Stopping Criterion. For the fault localization task, the user goal is to label all *failure rules*. The exploration thus stops when there is one explanation by failure rule, namely when all failure rules are labelled by **Explained**.

4.6 Discussion

The proposed approach does not depend on the type of patterns. The only requirement is a partial order on the patterns. There is always a natural order between patterns: the inclusion order of their cover in the original database. But constraints may introduce other more relevant orders.

There are several advantages to use the LIS framework. The first advantage is when the number of patterns is high and thus the size of the lattice is also high. LIS tools do not compute the whole lattice a priori but parts of the lattice on demand when it is relevant for the navigation. The second advantage is that the patterns are organized as a lattice. It allows to highlight the dependencies between them whereas other existing approaches, such as total ranking, do not provide that information. In addition, thanks to the logics, the patterns can have a rich description (e.g., integer values, partial order, taxonomy). Finally, when exploring, the user can add information in order to prune the search space and to be helped in its exploration.

The definition of a user goal is not a requirement, indeed, the user may want to navigate through patterns. But when the user has a clear goal, it is important to specify it with care. Indeed, the user goal defines the stopping criterion.

5 Related Work

The method that we propose in this paper can be applied after methods that reduce the number of patterns (e.g., condensed representation [PBT199]), when the set of patterns remains too large to be explored by hand. In addition, unlike top-k ranking methods, our method takes into account the user’s knowledge and highlights the dependencies between patterns. There are some methods that reduce more drastically the number of patterns. Recursive mining [CSK+08] gives control over the number of patterns, but some information is lost, indeed the data are summarized. In order to reduce the number of patterns, one can compute the stable concepts [Kuz07]. That method is interesting when the goal is to compute a general overview of the data with respect to group behaviours, for example [JKN08]. If the user is interested in patterns that are less frequent, the number of patterns remains high. The approach proposed in this paper does not a priori filter out patterns, but gives a data structure (the lattice) to facilitate the exploration of the patterns. In addition, thanks to the updating step and the stopping criterion, the number of patterns that are really explored by the user should be lower than the number of the whole set of patterns. In [CG05],

Garriga proposes to summarize sequential patterns thanks to partial orders; in [MOG09], the authors propose to explore association rules thanks to *rule schemas*; in [MG10], they propose to improve the method by integrating the user knowledge with an ontology. Unlike those methods, our approach is generic and can be applied on sequential patterns but also on association rules. The only requirement is the definition of the partial order on patterns. Richards *et al.* [RM03] have proposed to display the specific rules (Ripple-Down rules) in a lattice thanks to formal concept analysis (FCA) [GW99]. They create an artificial hierarchy over the rules in order to display them in a lattice. Note that to scale up, they have to filter out some nodes of the hierarchy, whereas our method does not a priori prune patterns. Visual data mining [Kei02, SBM08] provides users with a visualization of the patterns in a graphical way. That method is useful when little is known about the data and the exploration goals are vague. The goal of that kind of methods is not the same as the goal of our method. Indeed, our method is used when a lot of patterns are generated and the user has a specific goal and wants accurate details.

6 Conclusion

In this paper we have presented a new way to interactively explore patterns extracted with data mining techniques. The method is based on navigation in a partial order over the set of all patterns. It accommodates several kinds of patterns and the dependencies between patterns are taken into account. It allows users to use their background knowledge to navigate through the partial order until their goal(s) have been reached, without a priori pruning. In addition, the updating features allows the user to be helped by the reduction of the search space while exploring. We have formally defined our approach in the LCA framework, and we have illustrated our method on two different tasks and two different kinds of patterns.

References

- [AFR10] Allard, P., Ferré, S., Ridoux, O.: Discovering functional dependencies and association rules by navigating in a lattice of OLAP views. In: *Concept Lattices and Their Applications*, pp. 199–210. CEUR-WS (2010)
- [AIS93] Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) *Int. Conf. on Management of Data*. ACM Press, New York (1993)
- [AS95] Agrawal, R., Srikant, R.: Mining sequential patterns. In: *Int. Conf. on Data Engineering*. IEEE, Los Alamitos (1995)
- [CC10] Cellier, P., Charnois, T.: Fouille de données séquentielle d'itemsets pour l'apprentissage de patrons linguistiques. *Traitement Automatique des Langues Naturelles* (short paper) (2010)
- [CDFR08] Cellier, P., Ducassé, M., Ferré, S., Ridoux, O.: Formal concept analysis enhances fault localization in software. In: Medina, R., Obiedkov, S. (eds.) *ICFCA 2008*. LNCS (LNAI), vol. 4933, pp. 273–288. Springer, Heidelberg (2008)

- [CG05] Casas-Garriga, G.: Summarizing sequential data with closed partial orders. In: *SIAM International Data Mining Conference (SDM)* (2005)
- [CSK⁺08] Crémilleux, B., Soulet, A., Klema, J., Hébert, C., Gandrillon, O.: *Discovering Knowledge from Local Patterns in SAGE data*. IGI Publishing (2008)
- [DP90] Davey, B.A., Priestly, H.A.: *Introduction to Lattices and Order*, 2nd edn. Cambridge University Press, Cambridge (1990/2001)
- [Fer09] Ferré, S.: Camelis: a logical information system to organize and browse a collection of documents. *Int. J. General Systems* 38(4) (2009)
- [FPSS96] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: an overview. In: *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence (1996)
- [FR04] Ferré, S., Ridoux, O.: An introduction to logical information systems. *Information Processing & Management* 40(3), 383–419 (2004)
- [GK01] Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: *Proc. of the Int. Conf. on Conceptual Structures: Broadening the Base, ICCS 2001*, pp. 129–142. Springer, Heidelberg (2001)
- [GW99] Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer, Heidelberg (1999)
- [JKN08] Jay, N., Kohler, F., Napoli, A.: Analysis of social communities with iceberg and stability-based concept lattices. In: Medina, R., Obiedkov, S. (eds.) *ICFCA 2008. LNCS (LNAI)*, vol. 4933, pp. 258–272. Springer, Heidelberg (2008)
- [KB10] Kontonasios, K., De Bie, T.: An information-theoretic approach to finding informative noisy tiles in binary databases. In: *Proc. of the SIAM Int. Conf. on Data Mining*, pp. 153–164 (2010)
- [Kei02] Keim, D.A.: Information visualization and visual data mining. *IEEE Trans. Vis. Comput. Graph.* 8(1), 1–8 (2002)
- [Kuz07] Kuznetsov, S.O.: On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*. Springer Netherlands ACM (2007)
- [MG10] Marinica, C., Guillet, F.: Knowledge-based interactive postmining of association rules using ontologies. *IEEE Trans. Knowl. Data Eng.* (2010)
- [MOG09] Marinica, C., Olaru, A., Guillet, F.: User-driven association rule mining using a local algorithm. In: *Int. Conf. on Enterprise Information Systems (ICEIS)*, vol. (2), pp. 200–205 (2009)
- [PBT99] Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Beeri, C., Bruneman, P. (eds.) *ICDT 1999. LNCS*, vol. 1540, pp. 398–416. Springer, Heidelberg (1998)
- [PC09] Plantevit, M., Crémilleux, B.: Condensed representation of sequential patterns according to frequency-based measures. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (eds.) *IDA 2009. LNCS*, vol. 5772, pp. 155–166. Springer, Heidelberg (2009)
- [PHL01] Pei, J., Han, J., Lakshmanan, L.V.S.: Mining frequent itemsets with convertible constraints. In: *Int. Conf. on Data Engineering*. IEEE computer society, Los Alamitos (2001)
- [RM03] Richards, D., Malik, U.: Mining propositional knowledge bases to discover multi-level rules. In: Zaïane, O.R., Simoff, S.J., Djeraba, C. (eds.) *MDM/KDD 2002 and KDMCD 2002. LNCS (LNAI)*, vol. 2797, pp. 199–216. Springer, Heidelberg (2003)
- [SBM08] Simoff, S.J., Böhlen, M.H., Mazeika, A. (eds.): *Visual Data Mining*. Springer, Heidelberg (2008)

A Buzz and E-Reputation Monitoring Tool for Twitter Based on Galois Lattices

Etienne Cuvelier and Marie-Aude Aufaure

Business Intelligence Team
Applied Mathematics and Systems Department (MAS)
École Centrale Paris

Etienne.Cuvelier@ecp.fr, Marie-Aude.Aufaure@ecp.fr

Abstract. In the actual interconnected world, the speed of broadcasting of information leads the formation of opinions towards more and more immediacy. Big social networks, by allowing distribution, and therefore broadcasting of information in a almost instantaneous way, also speed up the formation of opinions concerning actuality. Then, these networks are great observatories of opinions and e-reputation. In this e-reputation monitoring task, it is easy to get a set of information (web pages, blog pages, tweets,...) containing a chosen word or a set of words (a company name, a domain of interest,...), and then we can easily search for the most used words. But a harder, but more interesting task, is to track the set of jointly used words in this dataset, because this latter contains the more shared advice about the initial searched set of words. Precisely, the exhaustive discovering of the shared properties of a collection of objects is the main task of the Galois lattices used in the Formal Concept Analysis. In this article we state clearly the characteristics, advantages and constraints of one of the more successful online social networks: Twitter. Then we detail the difficult task of tracking, on Twitter, the most forwarded information about a chosen subject. We also explain how the characteristics of Galois lattices permit to solve elegantly and efficiently this problem. But, retrieving the most used corpus of words is not enough, we have to show the results in an informative and readable manner, which is not easy when the result is a Galois Lattice. Then we propose a visualisation called topigraphic network of tags, which represent a tag cloud in a network of concepts with a topographic allegory, which permits to visualise the more important concepts found about a given search on Twitter.

1 Introduction

Since their appearance, blogs and social networks create a growing interest for observation and modelling of opinions, as illustrated by the special session on this subject of the TREC conferences since their edition of 2006 [1]. Identifying Hot Topics in the Blogosphere was one of the tasks of the 2009 edition of this blog session [2]. Social networks, like Facebook and Twitter, with their sharing and forward features, should also permit to observe the appearance of opinions

practically in real-time and then allow to detect tendencies. For instance [3] uses words expressing emotions in Facebook's status of the American users to synthesize a new index modelling the concept of "Gross National Happiness".

Social networks are therefore ideal places for the observation of opinions, notably regarding a chosen subject, that can be a person (personal branding), an official institution or an industrial operator. In the case of the e-reputation, the observation of the buzz and more particularly of the negative buzz (bad buzz) is important. But monitoring a buzz and/or an e-reputation is not only collect the set of information about a subject, it is also to structure this latter in a understandable way. We proposed a method for this on the most reactive of these networks: Twitter.

This article is organized in the following way: in the section 2 we detail succinctly how the Twitter network works, what are its constraints and its conventions, and what are the implied difficulties of analysis. In the section 3, we recall the basics of the Galois lattices which allow us to solve these difficulties. Finally in the section 4 we display the principles of our tool, as well as results acquired on the dataset of information relating to key word "e-reputation". We end with conclusions and perspectives of improvement.

2 Twitter and Micro-blogging

Twitter was created in 2006 to allow its users to share easily short textual messages called *Tweets*. The system was initially conceived to share tweets via SMS, and then a limit of 140 characters was fixed to these short messages. And even if nowadays the system is mainly used via web applications and mobile phones softwares, this constraint of 140 characters is still true. The basic principles of Twitter are the following:

- a user can , with its Twitter account, generates or forwards an information using a specific field (field "Whats happening?" in figure [1]);
- a user A can follows the tweets of a user B without this latter has to follow the tweets of A in return.

We see immediately that one characteristic of this social network is its asymmetric aspect.

The users which follow a Twitter account A are called his *followers*, while the users which A follows are called its *following*. The set of tweets of the following set from a given account is called his *timeline*. An illustration of each of these elements can be seen in the figure [1].

With its principle of "micro-blogging", Twitter allows to share information very quickly and then allows the diffusion of these information, but also of the opinions related to this latter. The growth of this service is nowadays important and, in April 2010, Twitter counted almost 6 million of recorded users, 300 000 new accounts a day and, on average, 55 million of tweets generated a day [4]. The result of this intense activity is a very big reactivity about the actuality facts, which can be illustrated by the the wikileaks case. In figure [2] we can see a

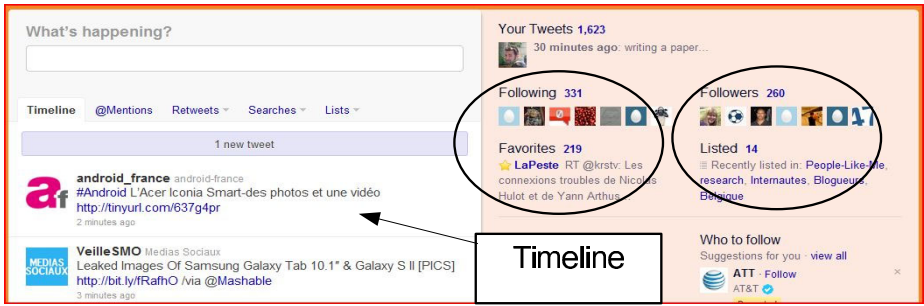


Fig. 1. Web interface for Twitter.com

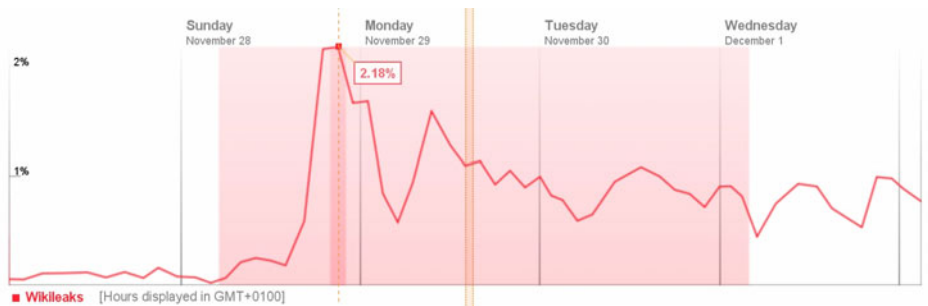


Fig. 2. Evolution of the number of tweets containing the word wikileaks since the publication on 28th of November of the diplomatics files (source: <http://trendistic.com>)

peak of 2% published tweets containing the word "wikileaks" less than 24 hours after the first release of the diplomatic documents on wikileaks^[5]. This reactivity is obviously very interesting for the analysis of the observation of opinions. For instance ^[6] showed that there was a very important correlation between three existent indicators, calculated via daily inquiries and opinions formulated on Twitter regarding these subjects: the first index concerns the trust level of the American consumers, the second one is an opinion polls Obama vs. Mc Cain during the American presidential campaign and the last one was an evaluation of the job of Obama as president. In a more predictive way, ^[7] showed that information circulating through Twitter concerning avian flu, linked to a model of prediction of market, allows to predict more efficiently the opinion concerning the transformation of influenza into pandemic.

The use of Twitter is ruled by some conventions, and we are going to specify the most important for a minimal understanding of the platform. The first one of these conventions is the use of the arobase to name or contact a chosen user. In example below, the user Jules contacts the user Jim naming the user Catherine:

Jules: @Jim see you at 22h00 at @Catherine's home ?

Both users Jim and Catherine will see this tweet in their own time-lines. A second convention is use of the *retweet or RT*. When an user notices in his time-line an information which he wants to share with his followers, he will use the retweet function of the service (web or application) which he uses, as illustrated in the example below:

Jules: Inception is an awesome movie.

Jim: Not for me RT @Jules: Inception is an awesome movie.

Catherine: LOL RT @Jim: Not for me RT @Jules: is an awesome movie.

Even if in most of the cases retweets are preceded by "RT @", other variants and practices coexist, as it was very well analysed in [8]. For instance some users edit the retweet by adding, a (*via*) at the end of this latter, as in following example:

Jules: http://www.google.com is awesome!

Catherine: RT @Jules: http://www.google.com is awesome!

Jim: http://www.google.com is awesome! (via @Jules)

This possibility of edition of a retweet, with the constraint of the 140 characters limit creates something that we call *polymorphism of forwarded information* on Twitter. This polymorphism is illustrated in the figure 3. In this example we see that the initial tweet (tweet N. 0) was retweeted in many ways. We see a first group of retweets (N. 1.1 and N. 1.2) in which the initial tweet is unchanged, and a second group of retweets (N. 2.1 and N. 2.2), where the users changed the retweet slightly. Furthermore one of these retweets of this second group is itself a retweet of retweet (tweet N. 2.2.1). This polymorphism is a real problem when we want to measure the popularity of an information, the number of unchanged retweets is then not a sufficient, because if we limit our count to this, then we will forget the whole set of modified retweets which, in spite of their modifications, carry the same information.

If we do not take into account the stop words and the signs of punctuations (in our example: more, of, and, are, by, us), and restrict our work to the significant words, then we can use a set representation of the information carried, as it can be seen in figure 4. We also can see in this latter how the different inclusions allow us to show the shared part by all tweets, the core of the carried information. Inclusions also define a partial order which can be represented by a Hasse's diagram as in the figure 5. Building such a diagram permits to establish, what are the common words in a group of tweets, as well as the different forms under which the same information was carried. Formal Concept Analysis and Galois Lattices are dedicated to the organization of information under this form.

3 Formal Concept Analysis – Galois Lattices

Formal Concept Analysis [9] is based on *Galois Lattices* [10], [11], which can be used for conceptual classification [12], [13]. A Galois Lattice allows to group, in a

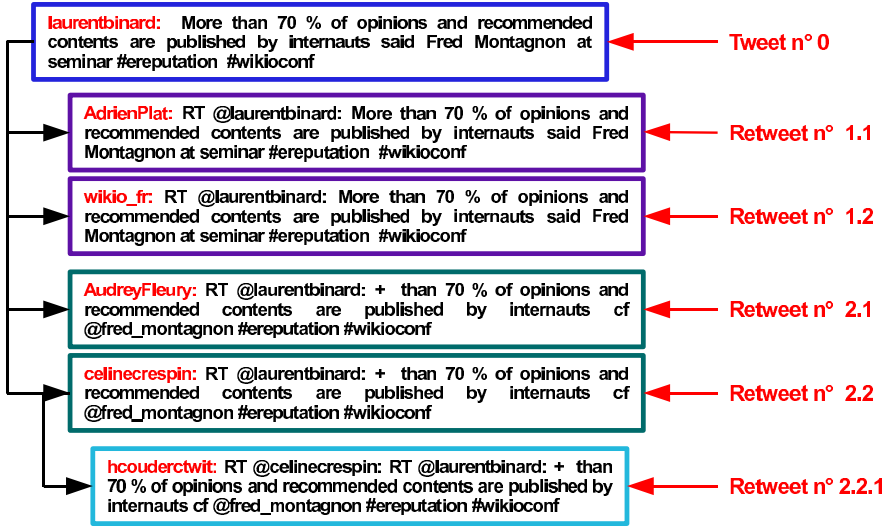


Fig. 3. Illustration of forwarded information’s polymorphism

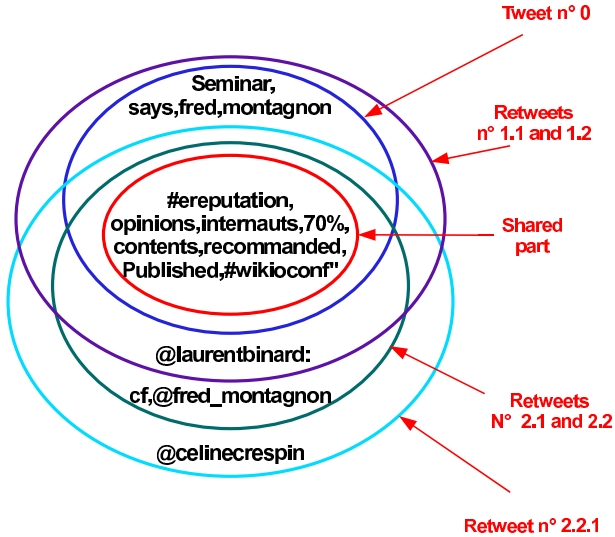


Fig. 4. Set point of view of forwarded information’s polymorphism

exhaustive way, objects in classes, called *concepts*, using their shared properties, and is usually based on a boolean matrix, called the *context matrix* denoted C . Rows of C represent a group of *objects* O , and the columns, a group of *attributes* A used for the description the objects. To introduce this notion of lattice, we will

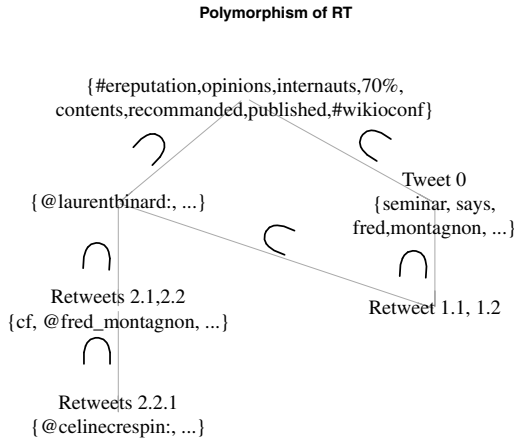


Fig. 5. Polymorphism analysis of information retweet

use the lattice shown in figure 5. The corresponding context matrix can be seen in table 1. The possession of property $a \in A$ by an object $o \in O$ materializes the existence of a relation I between them: aIo . The existence of this relation I between O and A is meant in the matrix of context C by a value "true" (and "false" otherwise) or by any mark (and anything otherwise). The triplet $K = (O, A, I)$ is called one *formal context* or simply a context.

The *intention* of a group $X \subset O$ is the set of attributes owned jointly by all objects of X and, given by the function f :

$$f(X) = \{a \in A | \forall o \in X, oIa\}. \tag{1}$$

Table 1. Table de contexte pour les tweets

Words Tweets	Seminar, says, fred, montagnon	#ereputation, opinions, internauts, 70%, contents, recommended, Published, #wikipioconf	@laurentbinard:	cf, @fred_montagnon	@celinecrespin
0	x	x			
1.1	x	x	x		
1.2	x	x	x		
2.1		x	x	x	
2.2		x	x	x	
2.2.1		x	x	x	x

Conversely the *extension* of a group $Y \subset A$ is all objects which jointly own all attributes of Y and, given by the function g :

$$g(Y) = \{o \in O \mid \forall a \in Y, oIa\}. \tag{2}$$

The couple (f, g) is called a *Galois connexion*.

A *concept* is any couple $C = (X, Y) \subset O \times A$, such as *the objects of X are the only ones to have all attributes of Y* , in other words $X \times Y$ form, except two permutations of O and of A , a maximum rectangle in C :

$$f(X) = Y \ \& \ g(Y) = X. \tag{3}$$

To illustrate this notion of concept, we can notice in table [11](#) that the set $X = \{Tweet\ 1.1, Tweet\ 1.2\}$ gives a concept because $f(X) = \{Seminar, says, fred, montagnon, \#ereputation, opinions, internauts, 70\%, contents, recommended, Published, \#wikioconf\ @laurentbinard:\} = Y$ and $g(Y) = X$, and this concept is then $(\{Tweet\ 1.1, Tweet\ 1.2\}, \{Seminar, says, fred, montagnon, \#ereputation, opinions, internauts, 70\%, contents, recommended, Published, \#wikioconf\ @laurentbinard:\})$, while the set $X' = \{Tweet\ 1.2, Tweet\ 2.1\}$ does not give a concept because $f(X') = \{\#ereputation, opinions, internauts, 70\%, contents, recommended, Published, \#wikioconf\ @laurentbinard:\} = Y'$ and $g(Y') = \{Tweet\ 1.1, Tweet\ 1.2, Tweet\ 2.1, Tweet\ 2.2, Tweet\ 2.2.1\} \neq X'$.

The *Galois lattice* is a *poset* of concepts L with the following partial order \leq :

$$(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 \text{ (or } Y_1 \supseteq Y_2). \tag{4}$$

The Galois lattice is denoted $T = (L, \leq)$ and, its representation is done using a *Hasse diagram*, as in figure [5](#) for the species. Two types of display exist for the labels of concepts, the full labelling and the reduced labelling. For the full labelling, all objects and all attributes of a concept are displayed, while in the reduced labelling, attributes and objects are displayed only once in the lattice. Attributes are displayed the first time they are met when going through the lattice top-down, while it is the contrary for objects, as we can see it in figure [5](#).

The construction of the lattice can be made using, for instance, the Bordat's algorithm [14](#), which compute recursively all the existing concepts starting from the concept $(\emptyset, f(\emptyset))$, computing for each found concept the set of its sub-concepts. A good review of other algorithms for Galois lattices generation can be found in [15](#) which gives also a comparison of performances.

One of the main advantages of the lattice classification is that for a given context table the resulting lattice is unique (no execution instability), and is exhaustive (all existing concepts will be found). In our case this classification is going to allow us to find all the groups of words in a set of given tweets, as seen in figure [5](#).

4 E-Buzz Monitoring

In the following section we propose to analyze a set of tweets, in order to find the most tweeted words or groups of words in the original set. For this we propose the following approach:

1. Getting the tweets including a chosen word or group of words,
2. Cleaning the tweets (suppressing stop words, punctuations,...);
3. Stating the table of context with the tweets as objects and the words as attributes;
4. Building the corresponding Galois lattice;
5. Visualisation of the results.

To illustrate our methods we are going to use it on a set of 50 tweets retrieved from a search on key word “#ereputation”. Here below for the sake of illustration we give the 5 first tweets of this set:

```
Tweet 1: overclub: #ereputation : your opinion on multiple technology
watch solutions... for you which is the best tool?
Tweet 2: AudreyFleury: #eReputation Internauts watch over their
ereputation Strategies http://ow.ly/1a7fWM
Tweet 3: AudreyFleury: RT @laurentbinard: + than 70% of opinions and
recommended contents are published by internauts cf @fred_montagnon
ereputation #wikiocnf
Tweet 4: hcouderctwit: RT @celinecrespin: RT @laurentbinard: + than
70% of opinions and recommended contents are published by internauts
cf @fred_montagnon #ereputation #wikiocnf
Tweet 5: wikio_fr: RT @laurentbinard: At seminar #wikiocnf, Serge
Alleyne, founder of #nomao, announces an presents its local
ereputation solution with #wikiobuzz...
```

Application of step 1 to 4 is easy on such a modest lattice, while the fifth step, the visualisation of the results is less trivial. In figure 6 we display the whole lattice giving to each concept a size proportional to the number of tweets contained. We notice that in spite of its small size (59 found concepts) it is difficult to display all the concepts proportionally to their sizes, and in the same time display clearly their attributes, even when using the reduced labelling. To reduce the number of attributes to be displayed, we can select only the concepts with a relative size (number of objects of the concepts divided by the number of objects in the context table) greater than chosen threshold. In other words, in respect to the notion of buzz we can select the concepts with the more tweeted words. That is what we have done in figure 7 with a threshold of 10%, but it does not increase enough the readability because the attributes of our lattice are, most of the times, groups of words more or less long. Another type of visualization is necessary for these concepts of the more retweeted words. Of course, we can try to display the different words of the found concepts using a classical tag cloud, giving to the tags a size proportional to the corresponding concept, but even if there exists solutions to display the associated tags near from the other one like in 16, we loose in this case the inclusion links between a concept and its sub-concepts. Moreover, as a sub-concept can have many super-concepts, it complicates the grouping task. That is why we propose to display the more important concepts using a network of proportional tags (figure 8), in which the links between the concepts will be materialized by edges. These edges will be

Complete Galois Lattice

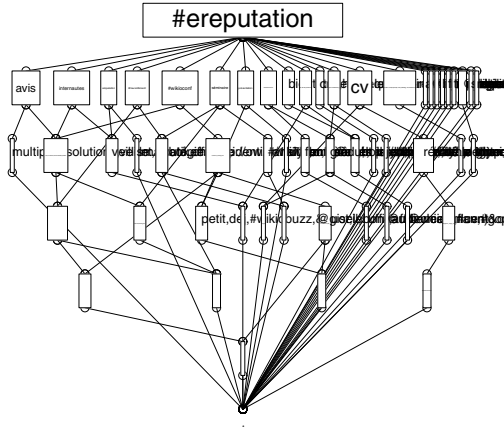


Fig. 6. The Galois lattice for the tweets

Galois Lattice, Concepts >0.1

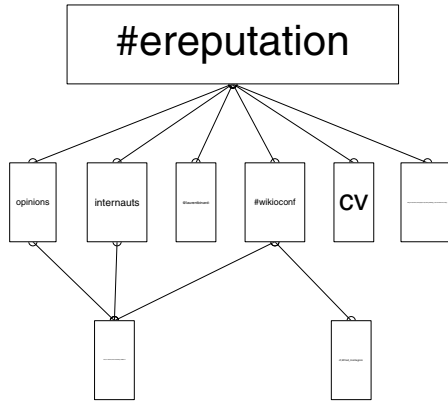


Fig. 7. The concepts containing more than 10% of tweets

directed, going from the concept toward its sub-concepts. For the nodes layout we use the Fruchterman-Reingold method [17], because this technique optimizes the distance between the nodes and allows us to increase the readability of the tags. Finally, to reinforce a reading going from the most general to the most particular, we have decided to add a topographic allegory similarly to the topographic maps

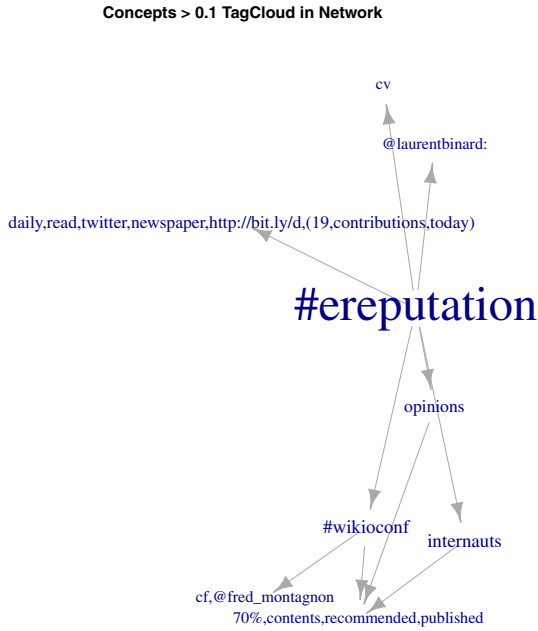


Fig. 8. A network of tags of concepts containing more than 10% of tweets

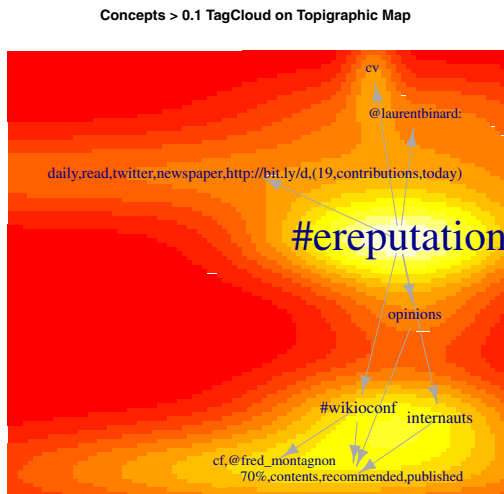


Fig. 9. The topigraphic network of tags of concepts containing more than 10% of tweets

proposed by [18]. We call the result a *topigraphic network of tags*. To do this, for each point of the resulting graphic, we add a level, these levels being pictured using the classical level curves. To compute the level of all the points of the graphic we use a bi-dimensional gaussian probability densities mixture, using as means the centers of the tags and, as standard deviations the width and the height of these tags. Finally, to give a height proportional to the concepts' sizes exactly at the centers of these tags, we normalize the heights of the gaussians multiplying them by the standard deviations and the by the desired heights. The resulting mixture is the topigraphic function T :

$$T(x, y) = \sum_{i=1}^k \frac{s_i}{2\pi} e^{-\frac{(x-x_i)^2(x-y_i)^2}{2l_i^2h_i^2}} \quad (5)$$

where:

- k is the number of displayed concepts,
- x_i and y_i are the coordinates of the i^{th} concept,
- l_i and h_i are the width and the height for the tag of the i^{th} concept,
- s_i is the size of the i^{th} concept.

Of course, as we have changed the volumes under the surfaces our topigraphic function T is not a probability density any more, but this property is not necessary in our case.

The final result can be seen in figure 9. In this figure, from the concept representing the starting key word $\{\#ereputation\}$, we can see that the more important sub-concepts are $\{opinions\}$, $\{internauts\}$ et $\{\#wikiconf\}$, and these three concepts contain also the concept $\{70\%, contents, recommended, published\}$, while only the concept $\{\#wikiconf\}$ contains the concept $\{cf, @fred_montagnon\}$. On the other hand we see three concepts shown independently of the first ones: $\{cv\}$, $\{@laurentbinard:\}$ et $\{daily, read, twitter, newspaper, http://bit.ly/d, (19, contributions, todays)\}$. The main idea of this visualisation is to let the reader's look slide from the "top" (the more general concepts) toward the "valleys" (the less general concepts). The constructions and the displaying of the Galois lattice and of the topigraphic network of tags were made in the R statistical environment [19], using for the lattice part our own package *galois* (to be published on CRAN).

5 Conclusions and Perspectives

In this paper we have presented a new technique for monitoring the buzz on the micro-blogging platform, Twitter. This technique is based on Galois lattices et and proposes as visualisation of the resulting concepts a topigraphic network of proportioned tags. This kind of display, limited to the more important concepts allows us to picture the tags belonging of a concept in a more readable manner than using directly the lattice. The main idea is to make "slip" the reader's

look, from the more general concepts, displayed at the “tops” toward the more particular concepts placed more in the “valleys”, arrows of the network being able to be seen as “lanes” to guide toward linked concepts.

Even if our proposal is only at a prototype stage, some improvement can be considered. The first one is the introduction of some interactivity to allow the user to select the sub-concept he wish to explore. We consider also to “develop” the shortened URLs (bit.ly, is.gd, tinyURL,..) in order to do not count twice a same final URL shortened using two different services. In the stop words suppression step, before stating the table of context we plan to a particular treatment could be reserved for smileys, which are meaningful. In a next step we envisage to use the sentiment analysis to assess the positivity or negativity of the found concepts, which is an interesting notion in the buzz and e-reputation monitoring. Finally the multi-language is an important but exciting challenge for such a tool.

References

1. Ounis, I., de Rijke, C.M.M., Mishne, G., Soboroff, I.: Overview of the trec 2006 blog track. NIST Special Publication 500-272: The Fifteenth Text REtrieval Conference Proceedings (TREC 2006), vol. 272, pp. 17–31 (2006)
2. Macdonald, C., Ounis, I., Soboroff, I.: Overview of the trec 2009 blog track. NIST Special Publication 500-278: The Eighteenth Text REtrieval Conference Proceedings (TREC 2009) (2009)
3. Kramer, A.D.I.: An unobtrusive behavioral model of gross national happiness. In: Proceedings of the 2010 Conference on Human Factors and Computing Systems (CHI 2010) (2010)
4. Bosker, B.: Twitter user statistics revealed (2010), <http://www.huffingtonpost.com/>
5. Razzi, M.: un jour historique (November 28, 2010), <http://www.courrierinternational.com/article/2010/11/29/28-novembre-2010-un-jour-historique>
6. O’Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From tweets to polls: Linking text sentiment to public opinion time series. In: Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, Washington, DC (2010)
7. Ritterman, J., Osborne, M., Klein, E.: Using prediction markets and twitter to predict a swine flu pandemic. In: 1st International Workshop on Mining Social Media (2009)
8. Boyd, D., Golder, S., Lotan, G.: Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In: Proceedings of the 43rd Hawaii International Conference on Social Systems (HICSS) (2010)
9. Wille, R.: Restructuring lattice theory, Ordered sets I. Rival (1980)
10. Barbut, M., Monjardet, B.: Ordre et classification, Algebre et combinatoire, Tome 2. Hachette (1970)
11. Birkhoff, G.: Lattice Theory, vol. 25. American Mathematical Society, New York (1940)
12. Carpineto, C., Romano, G.: Galois: An order-theoretic approach to conceptual clustering. In: Proc. of the 10th Conference on Machine Learning, Amherst, MA, pp. 33–40. Kaufmann, San Francisco (1993)

13. Wille, R.: Line diagrams of hierarchical concept systems. *Int. Classif.* 11, 77–86 (1984)
14. Bordat, J.: Calcul pratique du treillis de galois d'une correspondance. *Mathématique, Informatique et Sciences Humaines* 24, 31–47 (1986)
15. Kuznetsov, S.O., Obedkov, S.A.: Comparing performance of algorithms for generating concept lattices. In: *Concept Lattices-based Theory, Methods and Tools for Knowledge Discovery in Databases (CLKDD 2001)*, Stanford (July 30, 2001)
16. Kaser, O., Lemire, D.: Tag-cloud drawing: Algorithms for cloud visualization. In: *WWW 2007 Workshop on Tagging and Metadata for Social Information Organization*, Banff, Alberta (2007)
17. Fruchterman, T., Reingold, E.: Graph drawing by force-directed placement. *Software - Practice and Experience* 21, 1129–1164 (1991)
18. Fujimura, K., Fujimura, S., Matsubayashi, T., Yamada, T., Okuda, H.: Topigraphy: visualization for large-scale tag clouds. In: *Proceeding of the 17th International Conference on World Wide Web, WWW 2008*, pp. 1087–1088. ACM, New York (2008)
19. R Development Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2010) ISBN 3-900051-07-0

Using Generalization of Syntactic Parse Trees for Taxonomy Capture on the Web

Boris A. Galitsky¹, Gábor Dobrocsi¹, Josep Lluís de la Rosa¹,
and Sergei O. Kuznetsov²

¹ University of Girona, Girona, Catalonia, Spain
bgalitsky@hotmail.com, gadomail@gmail.com,
peplluís@silver.udg.edu

² Higher School of Economics, Moscow Russia
skuznetsov@yandex.ru

Abstract. We implement a scalable mechanism to build a taxonomy of entities which improves relevance of search engine in a vertical domain. Taxonomy construction starts from the seed entities and mines the web for new entities associated with them. To form these new entities, machine learning of syntactic parse trees (syntactic generalization) is applied to form commonalities between various search results for existing entities on the web. Taxonomy and syntactic generalization is applied to relevance improvement in search and text similarity assessment in commercial setting; evaluation results show substantial contribution of both sources.

Keywords: learning taxonomy, learning syntactic parse tree, syntactic generalization, search relevance.

1 Introduction

Nowadays, designing search engines and text relevance systems, it is hard to overestimate the role of taxonomies for improving their precisions, especially in vertical domains. However, building, tuning and managing taxonomies and ontologies is rather costly since a lot of manual operations are required. A number of studies proposed automated building of taxonomies based on linguistic resources and/or statistical machine learning, including multiagent settings [19, 21, 22]. However, most of these approaches have not found practical applications due to insufficient accuracy of resultant search, limited expressiveness of representations of queries of real users, or high cost associated with manual construction of linguistic resources and their limited adjustability.

In this study we propose automated taxonomy building mechanism which is based on initial set of main entities (a seed) for given vertical knowledge domain. This seed is then automatically extended by mining of web documents which include a meaning of a current taxonomy node. This node is further extended by entities which are the results of inductive learning of commonalities between these documents. These commonalities are extracted using an operation of syntactic generalization, which finds the common parts of syntactic parse trees of a set of documents, obtained for the current taxonomy node.

Syntactic generalization has been extensively evaluated commercially to improve text relevance [8, 9], and in this study we apply it for automated building of taxonomies.

Proceeding from parsing to semantic level is an important task towards natural language understanding, and has immediate applications in tasks such as information extraction and question answering [3, 5, 13]. In the last ten years there has been a dramatic shift in computational linguistics from manually constructing grammars and knowledge bases to partially or totally automating this process by using statistical learning methods trained on large annotated or non-annotated natural language corpora. However, instead of using such corpora, in this paper we use web search results for common queries, since their accuracy is higher and they are more up-to-date than academic linguistic resources.

The value of semantically-enabling search engines for improving search relevance has been well understood by the commercial search engine community [2]. Once an 'ideal' taxonomy is available, properly covering all important entities in a vertical domain, it can be directly applied to filtering out irrelevant answers. The state of the art in this area is how to apply a real-world taxonomy to search relevance improvement, where such a taxonomy is automatically compiled from the web and therefore is far from being ideal. It has become obvious that lightweight keyword based approaches cannot adequately tackle this problem. In this paper we address it combining web mining as a source of training set, and syntactic generalization as a learning tool.

2 Improving Search Relevance by Ontologies

To answer a question, natural language or keyword-based, it is beneficial to 'understand' what is this question about. In the sense of current paper this 'understanding' is a preferential treatment of keywords. We use the following definition of a relationship between a set of keywords and its element *is-about* (*set-of-keywords, keyword*).

For a query with keywords $\{a\ b\ c\}$ we understand that query is about b , if queries $\{a\ b\}$ and $\{b\ c\}$ are relevant or marginally relevant, and $\{a\ c\}$ is irrelevant. Our definition of query understanding, which is rather narrow, is the ability to say which keywords in the query are essential (such as b in the above example), so that without them the other query terms become meaningless, and an answer which does not contain b is irrelevant to the query which includes b .

For example, in the set of keywords $\{\textit{computer, vision, technology}\}$, $\{\textit{computer, vision}\}$, $\{\textit{vision, technology}\}$ are relevant, and $\{\textit{computer, technology}\}$ are not, so the query *is about* vision. Notice that if a set of keywords form a noun phrase or a verb phrase, it does not necessarily mean that the head or a verb is a keyword this ordered set is about. Also notice that we can group words into phrases when they form an entity:

$is\text{-}about(\{\textit{vision, bill, gates}\}, \emptyset)$, whereas
 $is\text{-}about(\{\textit{vision, bill-gates, in-computing}\}, \textit{bill-gates})$.

We refer to a keyword as *essential* if it occurs on the right side of *is-about*.

To properly formalize the latter observation, we generalize *is-about* relations towards the relation between a set of keywords and its subset. For query $\{a\ b\ c\ d\}$, if b is essential ($is-about(\{a\ b\ c\ d\}, \{b\})$), c can also be essential when b is in the query such that $\{a\ b\ c\}$, $\{b\ c\ d\}$, $\{b\ c\}$ are relevant, even $\{a\ b\}$, $\{b\ d\}$ are (marginally) relevant, but $\{a\ d\}$ is not ($is-about(\{a\ b\ c\ d\}, \{b,c\})$). Logical properties of sets of keywords, and logical forms expressing meanings of queries, are explored in [8]. There is a systematic way to treat relative importance of keywords via default reasoning [10]; multiple meanings of keyword combinations are represented via operational semantics of default logic.

Taxonomies are required to support query understanding. Taxonomies facilitate the assessments of whether a particular match between a query and an answer is relevant or not, based on the above notion of query understanding via *is-about* relation. Hence for a query $\{a\ b\ c\ d\}$ and two answers (snippets) $\{b\ c\ d\ \dots\ e\ f\ g\}$ and $\{a\ c\ d\ \dots\ e\ f\ g\}$, the former is relevant and the latter is not.

Achieving relevancy using a taxonomy is based on totally different mechanism than a conventional TF*IDF based search. In the latter, importance of terms is based on the frequency of occurrence, and any term can be omitted in the search result if the rest of terms give acceptable relevancy score. In the taxonomy-based search we know which terms *should* occur in the answer and which terms *must* occur there, otherwise the search result becomes irrelevant.

2.1 Building Taxonomy by Web Mining

Our main hypotheses for automated learning taxonomies on the web is that common expressions between search results for a given set of entities gives us *parameters* of these entities. Formation of the taxonomy follows the unsupervised learning style. It can be viewed as a human development process, where a baby explores new environment and forms new rules. Initial set of rules is set genetically, and the learning process adjusts these rules to particular habituation environment, to make these rules more sensitive (and therefore allows more beneficial decision making). As new rules are being accepted or rejected during their application process, exposure to new environment facilitates formation of new specific rules. After the new, more complex rules are evaluated and some part of these newly formed rules is accepted, complexity of rules grows further to adapt to further peculiarities of environment.

We learn new entities to extend our taxonomy in a similar unsupervised learning setting. We start with the seed taxonomy, which enumerates the main entities of a given domain, and relations of these entities with a few domain-determining concepts. For example, a seed for tax domain will include the relationships

$$tax - deduct \quad tax-on-income \quad tax-on-property,$$

where *tax* is a domain-determining entity, and $\{deduct, income, property\}$ are main entities in this domain. The objective of taxonomy learning is to acquire further parameters of existing entities such as *tax - deduct*. In the next iteration of learning these parameters will be turned into entities, so that a new set of parameters will be learned (Fig. 1).

Learning iteration is based on web mining. To find parameters for a given set of tree leaves (current entities), we go to the web and search for common expressions between the search results (snippets) for query formed for current tree paths. For the example above, we search for *tax-deduct*, *tax-on-income*, *tax-on-property* and extract words and expressions which are **common** between search results. Common words are single verbs, nouns, adjectives and even adverbs or multi-words, including prepositional, noun and verb phrases, which occur in **multiple** search results. The central part of our paper, Section 3, explains how to extract common expressions between search results and form new set of current entities (taxonomy leaves).

After such common words and multi-words are identified, they are added to the original words. E.g. for the path *tax - deduct* newly learned entities can be

<i>tax-deduct</i> → <i>decrease-by</i>	<i>tax-deduct</i> → <i>of-income</i>
<i>tax-deduct</i> → <i>property-of</i>	<i>tax-deduct</i> → <i>business</i>
<i>tax-deduct</i> → <i>medical-expense</i> .	

The format here is *existing_entity* → *its parameter (to become a new_entity)*, ‘→’ here is an unlabeled ontology edge.

Now from the path in the taxonomy tree *tax – deduct* we obtained five new respective paths. The next step is to collect parameters for each path in the new set of leaves for the taxonomy tree. In our example, we run five queries and extract parameters for each of them. The results will look like:

<i>tax- deduct-decrease-by</i> → <i>sales</i>
<i>tax-deduct-decrease-by</i> → <i>401-K</i>
<i>tax-deduct-decrease</i> → <i>medical</i>
<i>tax - deduct- of-income</i> → <i>rental</i>
<i>tax – deduct - of-income</i> → <i>itemized</i>
<i>tax – deduct – of-income</i> → <i>mutual-funds</i>

For example, searching the web for *tax-deduct-decrease* allows discovery of an entity *sales-tax* associated with decrease of tax deduction, usually with meaning ‘sales tax’ (italicized and highlighted in Fig.1). Commonality between snippets shows the sales tax should be taken into account while calculating *tax deduction*, and not doing that would *decrease* it.

Hence the taxonomy is built via inductive learning of web search results in iterative mode. We start with the taxonomy seed nodes, then find web search results for all currently available graph paths, and then for each commonality found in these search results we augment each of these taxonomy paths by adding respective leaf nodes. In other words, for each iteration we discover the list of parameters for each set of currently available entities, and then turn these parameters into entities for the next iteration (Fig.2).

The taxonomy seed is formed manually or can be compiled from available domain-specific resources. Seed taxonomy should contain at least 2-3 nodes so that taxonomy growth process has a meaningful start. Taxonomy seed can include, for example, a glossary of particular knowledge domain, readily available for a given vertical domain, like <http://www.investopedia.com/categories/taxes.asp> for tax entities.

- [How to Decrease Your Federal Income Tax | eHow.com](#)
the Amount of Federal **Taxes** Being Withheld; How to Calculate a Mortgage Rate After In-
come **Taxes**; How to **Deduct Sales Tax** From the Federal Income **Tax**
- [Itemizers Can Deduct Certain Taxes](#)
... may be able to **deduct** certain **taxes** on your federal income **tax** return? You can take
these **deductions** if you file Form 1040 and itemize **deductions** on Schedule
A. **Deductions decrease** ...
- [Self Employment Irs Income Tax Rate Information & Help 2008, 2009 ...](#)
You can now **deduct** up to 50% of what has been paid in self employment **tax**. · You are able
to **decrease** your self employment income by 7.65% before figuring your **tax** rate.
- [How to Claim Sales Tax | eHow.com](#)
This amount, along with your other itemized **deductions**, will **decrease** your taxable ... How
to **Deduct Sales Tax** From Federal **Taxes**; How to Write Off **Sales Tax**; Filling **Taxes** with ...
- [Prepaid expenses and Taxes](#)
How would prepaid expenses be accounted for in determining **taxes** and accounting for ... as
the cash effect is not yet determined in the net income, and we should **deduct a decrease**, and
...
- [How to Deduct Sales Tax for New Car Purchases: Buy a New Car in ...](#)
How to **Deduct Sales Tax** for New Car Purchases Buy a New Car in 2009? Eligibility Re-
quirements ... time homebuyer credit and home improvement credits) that are available
to **decrease** the ...

Fig. 1. Search results on Bing.com for the current taxonomy tree path *tax-deduct-decrease*

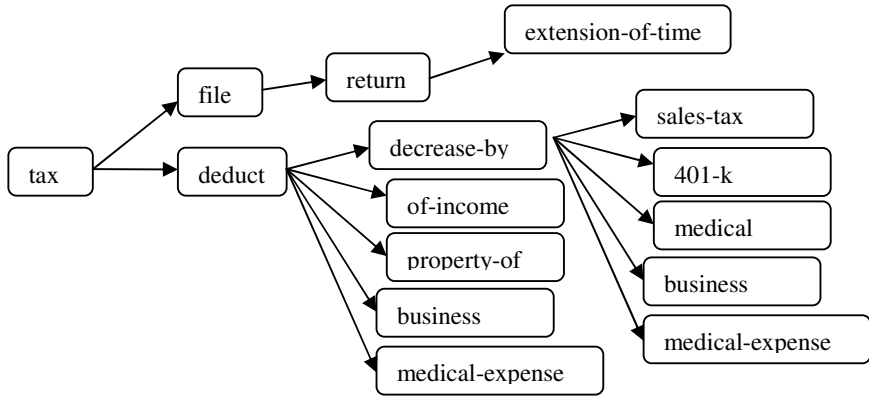


Fig. 2. Taxonomy for tax domain

2.2 Filtering Answers Based on Taxonomy

To use the taxonomy to filter out irrelevant questions, we search for taxonomy path (down to a leaf node if possible) which is closest to the given question in terms of the number of entities from this question. Then this path and leaf node specify most accurate meaning of the question, and constrain which entities *must* occur and which *should* occur in the answer to be considered relevant. If the n-th node entity from the question occurs in answer, then all $k < n$ entities should occur in it as well.

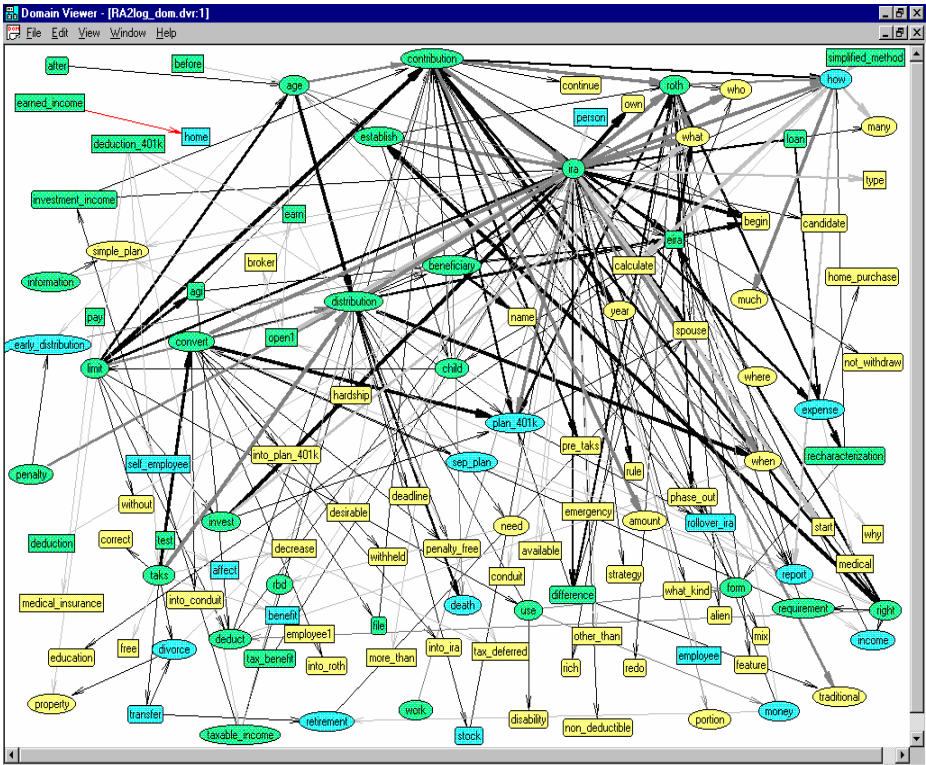


Fig. 3. Visualization of a log graph

For the majority of search applications, *acceptability* condition is easier to apply than the ‘most accurate’ condition: An answer A is acceptable if it includes all essential keywords from the question Q as found in the taxonomy path $T_p \in T$

$$A \subseteq T_p \cap Q.$$

For the best answer we write $A_{best} : \max(\text{cardinality}(A_{best} \cap (T_p \cap Q)))$,

where $S \cap G$ is an operation of finding a maximal path in a graph G whose node labels belong to a set S , so that *is-about*(K, S) for some K .

Examples above illustrate this main requirement. Naturally, multiple taxonomy paths exist. Taxonomies help to solve disambiguation problem. For a question

(Q) "When can I file extension of time for my tax return?"

let us imagine two answers:

(A1) "You need to file form 1234 to request a 4 month extension of time to file your tax return"

(A2) "You need to download file with extension 'pdf', print and complete it to file your tax return".

We expect the closest taxonomy path to be :

(T) tax - file-return - extension-of-time.

tax is a main entity, *file-return* we expect to be in the seed, and *extension-of-time* would be the learned entity, so A1 will match with taxonomy and is an acceptable answer, and A2 is not.

Another way to represent taxonomy is not to enforce it to be a tree, **least general** but only allow one node for each label instead (Fig 3).

3 Syntactic Generalization of Sentences

To measure similarity of abstract entities expressed by logic formulas, the least-general generalization (also called anti-unification) was proposed for a number of machine learning approaches, including explanation-based learning and inductive logic programming. It is the opposite of most general unification, therefore it is also called anti-unification [12]. To form commonality expression between search result snippets (in general, to measure similarity between NL expressions), we extend the notion of generalization from logic formulas to syntactic parse trees of these expressions. If it were possible to define similarity between natural language expressions at pure semantic level, least general generalization of logical formulas would be sufficient. However, in text mining problem, we need to deal with original text, so we apply generalization to syntactic parse trees obtained for each sentence. Rather than extracting common keywords, generalization operation produces a syntactic expression that can be semantically interpreted as a common meaning shared by two sentences, which may underlie a new entity for taxonomy node.

- 1) Obtain parsing tree for each sentence. For each word (tree node) we have <lemma, part of speech , word form> information. This information is contained in the node label. We also have an arc to the other node.
- 2) Split sentences into sub-trees which are phrases of each type: verb, noun, prepositional and others; these sub-trees are overlapping. The sub-trees are coded so that information about occurrence in the full tree is retained.
- 3) All sub-trees are grouped by phrase types.
- 4) Extending the list of phrases by adding equivalence transformations. Generalize each pair of sub-trees for both sentences for each phrase type.
- 5) For each pair of sub-trees perform an alignment of phrases, and then generalize each node of these aligned sentences as sub-trees. For the obtained set of trees (generalization results), calculate the score which is a POS-weighted sum of the number of nodes for all trees from this set (see details in [9]).
- 6) For each pair of sub-trees for phrases, select the set of generalizations with highest score (least general).
- 7) Form the sets of generalizations for each phrase types whose elements are sets of generalizations for this type.
- 8) Filtering the list of generalization results: for the list of generalization for each phrase type, select least general elements from this list of generalization for a given pair of phrases.

For a pair of phrases, generalization includes all *maximum* ordered sets of generalization nodes for words in phrases so that the order of words is retained. In the following example

To buy digital camera today, on Monday

Digital camera was a good buy today, first Monday of the month

Generalization results are the sets of sub-trees {*digital - camera , today - Monday*}, where part of speech information is not shown. *buy* is excluded from both generalizations because it occurs in a different order in the above phrases. *Buy - digital - camera* is not a generalization because *buy* occurs in different sequence with the other generalization nodes.

The result of generalization can be further generalized with other parse trees or generalizations. For a set of sentences, the totality of generalizations forms a lattice: order on generalizations is set by the subsumption relation and generalization score. Generalization of parse trees obeys the associativity by means of computation: it has to be verified and resultant list extended each time new sentence is added. Further details on syntactic generalization can be obtained in [9].

4 Evaluation of Search Relevance Improvement

Evaluation of search included an assessment of classification accuracy for search results as relevant and irrelevant. Since we used the generalization score between the query and each hit snapshot, we drew a threshold of five highest score results as relevant class and the rest of search results as irrelevant. We used the Yahoo search API and applied the generalization score to find the highest score hits from first fifty Yahoo search results (Fig. 4). We then consider the first five hits with the highest generalization score (not Yahoo score) to belong to the class of relevant answers. Third and second rows from the bottom contain classification results for the queries of 3-4 keywords which is slightly more complex than an average one (3 keywords); and significantly more complex queries of 5-7 keywords, respectively.

The total average accuracy (F-measure) for all above problems is 79.2%. Since the syntactic generalization was the only source of classification, we believe the accuracy is satisfactory. A practical application would usually use a hybrid approach with rules and keyword statistic which would deliver higher overall accuracy, but such application is beyond the scope of this paper. Since the generalization algorithm is deterministic, higher accuracy can be also achieved by extending training set.

In this study we demonstrated that such high-level sentences semantic features as *being informative* can be learned from the low level linguistic data of complete parse tree. Unlike the traditional approaches to *multilevel* derivation of semantics from syntax, we explored the possibility of linking low level but detailed syntactic level with high-level pragmatic and semantic levels *directly*.

Table 1. Evaluation of classification accuracy

Type of search query	Relevancy of Yahoo search, %, averaging over 10	Relevancy of resorting by generalization, %, averaging over 10	Relevancy compared to baseline, %
3-4 word phrases	77	77	100.0%
5-7 word phrases	79	78	98.7%
8-10 word single sentences	77	80	103.9%
2 sentences, >8 words total	77	83	107.8%
3sentences,>12 words total	75	82	109.3%

Can Form 1040 EZ be used to claim the earned income credit

You can change the ordering of the table by clicking on column-headers.

[First result](#) [Previous result](#) [Next result](#) [Last result](#)

ORIGINAL-RANK ▾	SYNTACTIC-MATCH SCORE	TAXONOMY-SCORE	TITLE & ABSTRACT
16	3.3	1	2010 Form W-5 Use Form W-5 if you are eligible to get part of t
3	3.3	4	Earned Income Credit Can Form 1040EZ be used to claim the earned i
2	3.3	4	Can Form 1040EZ be used to claim the earned i Can Form 1040EZ be used to claim the earned i
0	3.3	4	Other EITC Issues Question: Can Form 1040EZ be used to claim th
20	3.0	0	Line by Line Tips for Form 1040-EZ (Year 2008) Prepare your 2008 tax returns on Form 1040-EZ
5	3.0	0	Line by Line Tips for Form 1040-EZ (Year 2009) Prepare your 2009 tax returns on Form 1040-EZ
17	2.9	1	FREE 1040EZ - FREE Federal 1040EZ - Federal 10 Now, as an individual, you may wonder whether y
27	2.8	0	2007 Form W-5 I expect to have a qualifying child and be able t
19	2.8	1	2008 Form W-5 I expect to have a qualifying child and be able t

Fig. 4. Sorting search results by syntactic generalization vs taxonomy-based for a given query

We selected Citizens Advise Services as another application domain where taxonomy improves relevance of recommendations easy4.udg.edu/isac/eng/index.php [17,18]. Taxonomy learning of the tax domain was conducted in English and then translated in Spanish, French, German and Italian. It was evaluated by project partners using the tool in Fig 5, where to improve search precision a project partner in a particular location modifies the automatically learned taxonomy to fix a particular case, upload the taxonomy version adjusted for a particular location and verify the improvement of relevance.

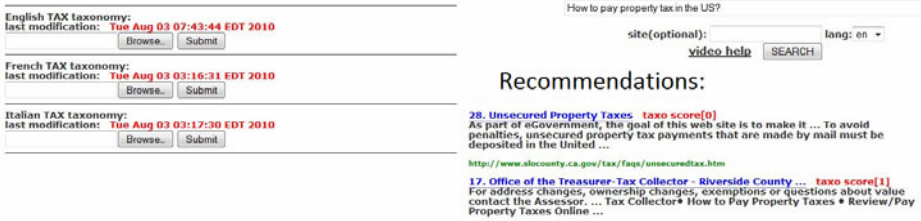


Fig. 5. A tool for manual adjustment of taxonomy for providing Citizens Recommendation Services, <http://box.cs.rpi.edu:8080/wise/taxo.jsp>

4.1 Commercial Evaluation of Text Similarity Improvement

We subject the proposed technique of taxonomy-based and syntactic generalization-based techniques in the commercial area of news analysis at AllVoices.com. The task is to cluster relevant news together, by means of text relevance analysis. By definition, multiple news articles belong to the same cluster, if there is a substantial overlap of involved entities such as geo locations and names of individuals, organizations and other agents, as well as relations between them. Some of these can be extracted by entity taggers, and/or by using taxonomies, and some are handled in real time using syntactic generalization (Fig. 7, oval on the right). The latter is applicable if there is a lack of prior entity information.

In addition to forming a cluster of relevant documents, it is necessary to aggregate relevant images and videos from different sources such as Google image, YouTube and Flickr, and access their relevance given their textual descriptions and tags, where the similar taxonomy and syntactic generalization-based technique is applied (Fig. 6).

Precision of text analysis is achieved by site usability (click rate) of more than nine million unique visitors per month. Recall is accessed manually; however the system needs to find at least a few articles, images and videos for each incoming article. Usually, for web mining and web document analysis recall is not an issue, it is assumed that there are a high number of articles, images and videos on the web for mining.

Precision data for the relevance relation between an article and other articles, blog postings, images and videos is presented in Table 2 (the percentages are normalized taking into account the decreased recall). Notice that although the taxonomy-based method on its own has a very low precision and does not outperform the baseline of the statistical assessment, there is a noticeable improvement of precision in hybrid system. We can conclude that syntactic generalization and taxonomy-based methods (which also rely on syntactic generalization) use different sources of relevance information, so they are indeed complementary to each other.

The objective of syntactic generalization was to filter out false-positive relevance decision, made by statistical relevance engine designed following [21,22]. The percentage of false-positive news stories was reduced from 29 to 13 (about 30000 stories/month viewed by 9 million unique users), and the percentage of false positive image attachment was reduced from 24 to 18 (about 3000 images and 500 videos attached to stories monthly).

Fireworks Likely Caused 3,000 Ark. Bird Deaths

Relevance Verifier Results PASSED

Fox | about 14 hours ago

Hide Delete

Dead birds lie on the ground after being thrown off the roof of a home by a worker in Beebe, Ark. Ark. -- Celebratory fireworks likely sent thousands of discombobulated blackbirds into such a tizzy that they crashed into homes, cars and each other...

4 and 20 blackbirds, and 3,000, dead in the sky

Relevance Verifier Results FAILED

The Boston Globe | about 16 hours ago

Hide Delete

Celebratory fireworks likely sent thousands of discombobulated blackbirds into such a tizzy that they crashed into homes, cars and each other before plummeting to their deaths in central Arkansas, scientists say. Still, officials acknowledge it's...

Mass La. bird deaths puzzle investigators

Relevance Verifier Results PASSED

Hide Delete

Relevance Verifier Results

Decision: PASSED

Final Score: 7.630000000000003

Breakdown:

- **Rule:** infrequent noun is found0
Logs: oupee
Score: 0.7
- **Rule:** frequent noun is found4
Logs: dead
Score: 0.2
- **Rule:** frequent noun is found3
Logs: mile
Score: 0.2
- **Rule:** frequent noun is found2
Logs: estimated
Score: 0.2
- **Rule:** frequent noun is found1
Logs: birds
Score: 0.2
- **Rule:** frequent noun is found0
Logs: determine
Score: 0.2
- **Rule:** nouns phrases from image tried
Logs: [Pointe Coupee Parish, red-winged blackbirds starlings La, deaths red-winged blackbirds starlings La]
Score: 0.0
- **Rule:** synt match result
Logs: np [[NNS-birds], [JJ-dead NNS-birds]] vp [[IN- NP-* IN-in NP-*]]
Score: 2.1
- **Rule:** string and keyword similarity
Logs: High
Score: 1.1308178713196471
- **Rule:** category
Logs: different categs or no categ available
Score: 0.0
- **Rule:** attempted to find People's names
Logs: [Georgia]
Score: 0.0
- **Rule:** found common geolocation city
Logs: 228
Score: 0.7

Fig. 6. Explanation for relevance decision while forming a cluster of news articles for the one on Fig.6. The circled area shows the syntactic generalization result for the seed articles and the given one.

Table 2. Improvement the precision of text similarity

Media/ method of text similarity assessment	Full size news articles	Abstracts of articles	Blog posting	Comments	Images	Videos
Frequencies of terms in documents	29.3%	26.1%	31.4%	32.0%	24.1%	25.2%
Syntactic generalization	17.8%	18.4%	20.8%	27.1%	20.1%	19.0%
Taxonomy-based	45.0%	41.7%	44.9%	52.3%	44.8%	43.1%
Hybrid (taxonomy + syntactic)	13.2%	13.6%	15.5%	22.1%	18.2%	18.0%

5 Related Work and Conclusions

For a few decades, most approaches to NL semantics relied on mapping to First Order Logic representations with a general prover and without using acquired rich knowledge sources. Significant development in NLP, specifically the ability to acquire knowledge and induce some level of abstract representation such as taxonomies is expected to support more sophisticated and robust approaches. A number of recent approaches are based on shallow representations of the text that capture lexico-syntactic relations based on dependency structures and are mostly built from grammatical functions extending keyword matching [15]. On the contrary, taxonomy learning in this work is performed in a vertical domain, where ambiguity of terms is limited, and therefore fully automated settings produce adequate resultant search accuracy. Hence our approach is finding a number of commercial applications including relevancy engine at citizens' journalism portal AllVoices.com and search and recommendation at Zvents.com.

Usually, classical approaches to semantic inference rely on complex logical representations. However, practical applications usually adopt shallower lexical or lexical-syntactic representations, but lack a principled inference framework. A generic semantic inference framework that operates directly on syntactic trees has been proposed. New trees are inferred by applying entailment rules, which provide a unified representation for varying types of inferences. Rules are generated by manual and automatic methods, covering generic linguistic structures as well as specific lexical-based inferences. The current work deals with syntactic tree transformation in the graph learning framework (compare with [4, 16]), treating various phrasings for the same meaning in a more unified and automated manner.

Traditionally, semantic parsers are constructed manually, or are based on manually constructed semantic ontologies, but these are too delicate and costly. A number of supervised learning approaches to building formal semantic representation have been proposed [6]. Unsupervised approaches have been proposed as well, however they applied to shallow semantic tasks [14]. The problem domain in the current study required much deeper handling of syntactic peculiarities to build taxonomies. In terms of learning, our approach is closer in merits to unsupervised learning of complete formal semantic representation. Compared to semantic role labeling [7] and other forms of shallow semantic processing, our approach maps text to formal meaning representations, obtained via generalization.

There are a number of applications of formal concepts in building natural language taxonomies. Formal framework based on formal concept lattices that categorizes epistemic communities automatically and hierarchically, rebuilding a relevant taxonomy in the form of a hypergraph of epistemic sub-communities, has been proposed in [23]. The study of concepts can advance further by clarifying the meanings of basic terms such as "prototype" and by constructing a large-scale primary taxonomy of concept types [11]. Based on concept structures, two secondary concept taxonomies and one of conceptual structures has been built, where the primary taxonomy organizes much data and several previous taxonomies into a single framework. It suggests that many concept types exist, and that type determines how a concept is learned, is used and how it develops. [1] provides a tool to facilitate the re-use of existing knowledge structures such as taxonomies, based on the ranking of ontologies.

This tool uses as input the search terms provided by a knowledge engineer and, using the output of an ontology search engine, ranks the taxonomies. A number of metrics in an attempt to investigate their appropriateness for ranking ontologies has been applied, and results were compared with a questionnaire-based human study.

The use of syntactic generalization in this work is two-fold. Firstly, it is used off-line to form the node of taxonomy tree, finding commonalities between search results for a given taxonomy node. Secondly, syntactic generalization is used online for measuring similarity of either two portions of text, or question and answer, to measure the relevance between them. We demonstrated that merging taxonomy-based methods and syntactic generalization methods improves the relevance of text understanding in general, and complementary to each other, because the former uses pure meaning-based information, and the latter user linguistic information about the involved entities. Naturally, such combination outperforms a bag-of-words approach in horizontal domain, and also, according to our evaluation, outperforms a baseline statistical approach in a vertical domain.

Acknowledgements. This research is funded by the European Union project Num. 238887, *a unique European citizens' attention service (iSAC6+)* IST-PSP, the ACCIÓ Catalan Government *grant ASKS - Agents for Social Knowledge Search*, the Spanish MCINN (Ministerio de Ciencia e Innovación) project IPT-430000-2010-13 project *Social powered Agents for Knowledge search Engine (SAKE)*, and the CSI-ref.2009SGR-1202.

References

1. Alani, H., Brewster, C.: Ontology ranking based on the analysis of concept structures. In: K-CAP 2005 Proceedings of the 3rd International Conference on Knowledge Capture (2005)
2. Heddon, H.: Better Living Through Taxonomies. Digital Web Magazine (2008), http://www.digital-web.com/articles/better_living_through_taxonomies/
3. Allen, J.F.: Natural Language Understanding, Benjamin Cummings (1987)
4. Chakrabarti, D., Faloutsos, C.: Graph Mining: Laws, Generators, and Algorithms. *ACM Computing Surveys* 38(1) (2006)
5. Dzikovska, M., Swift, M., Allen, J., de Beaumont, W.: Generic parsing for multi-domain semantic interpretation. In: International Workshop on Parsing Technologies (IWPT 2005), Vancouver BC (2005)
6. Cardie, C., Mooney, R.J.: Machine Learning and Natural Language. *Machine Learning* 1(5) (1999)
7. Carreras, X., Marquez, L.: Introduction to the CoNLL-2004 shared task: Semantic role labeling. In: Proceedings of the Eighth Conference on Computational Natural Language Learning, pp. 89–97. ACL, Boston (2004)
8. Galitsky, B.: Natural Language Question Answering System: Technique of Semantic Headers. In: Advanced Knowledge International, Australia (2003)
9. Galitsky, B., Dobrocsi, G., de la Rosa, J.L., Kuznetsov, S.O.: From Generalization of Syntactic Parse Trees to Conceptual Graphs. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 185–190. Springer, Heidelberg (2010)

10. Galitsky, B.: Disambiguation Via Default Rules Under Answering Complex Questions. *Intl. J. AI. Tools* 14(1-2) (2005)
11. Howard, R.W.: Classifying types of concept and conceptual structure: Some taxonomies. *Journal of Cognitive Psychology* 4(2), 81–111 (1992)
12. Plotkin., G.D.: A note on inductive generalization. In: Meltzer, Michie (eds.) *Machine Intelligence*, vol. 5, pp. 153–163. Edinburgh University Press, Edinburgh (1970)
13. Ravichandran, D., Hovy, E.: Learning surface text patterns for a Question Answering system. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, PA (2002)
14. Lin, D., Pantel, P.: DIRT: discovery of inference rules from text. In: *Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2001*, pp. 323–328 (2001)
15. Durme, B.V., Huang, Y., Kupsc, A., Nyberg, E.: Towards light semantic processing for question answering. In: *HLT Workshop on Text Meaning* (2003)
16. Kapoor, S., Ramesh, H.: Algorithms for Enumerating All Spanning Trees of Undirected and Weighted Graphs. *SIAM J. Computing* 24, 247–265 (1995)
17. De la Rosa, J.L., Rovira, M., Beer, M., Montaner, M., Gibovic, D.: Reducing Administrative Burden by Online Information and Referral Services. In: Reddick, C.G. (ed.) *Citizens and E-Government: Evaluating Policy and Management*, pp. 131–157. IGI Global, Austin (2010)
18. López Arjona, A.M., Rigall, M.M., de la Rosa i Esteva, J.L., Regàs, M.M.R.I.: POP2.0: A search engine for public information services in local government. In: Angulo, C., Godo, L. (eds.) *Artificial Intelligence Research and Development*, vol. 163, pp. 255–262. IOS Press, Amsterdam (2007)
19. Kozareva, Z., Hovy, E., Riloff, E.: Learning and Evaluating the Content and Structure of a Term Taxonomy. In: *Learning by Reading and Learning to Read AAAI Spring Symposium*, Stanford CA (2009)
20. Liu, J., Birnbaum, L.: What do they think? Aggregating local views about news events and topics. In: *WWW 2008*, pp. 1021–1022 (2008)
21. Liu, J., Birnbaum, L.: Measuring Semantic Similarity between Named Entities by Searching the Web Directory. *Web Intelligence*, 461–465 (2007)
22. Kerschberg, L., Kim, W., Scime, A.: A Semantic Taxonomy-Based Personalizable Meta-Search Agent. In: Truszkowski, W., Hinchey, M., Rouff, C.A. (eds.) *WRAC 2002. LNCS*, vol. 2564, pp. 3–31. Springer, Heidelberg (2003)
23. Roth, C.: Compact, evolving community taxonomies using concept lattices *ICCS 14*, July 17-21, Aalborg, DK (2006)

A.N. Prior's Ideas on Tensed Ontology

David Jakobsen, Peter Øhrstrøm, and Henrik Schärfe

Department of Communication and Psychology, Aalborg University
Kroghstræde 3, 9220 Aalborg East, Denmark
{davker, poe, scharfe}@hum.aau.dk

Abstract. A.N. Prior's work with Peirce's philosophy and investigations into the formalisation of temporal ontology give rise to some important questions regarding time and existence. Some answers to these questions are considered in this paper, which deals mainly with A.N. Prior's ideas on time and existence. The focus is on Prior's analysis of problems concerning the ontology of objects which exist for a limited time only. This work led him to some interesting ideas concerning tensed ontology. In particular, the paper discusses Prior's own contribution to tensed ontology, which may be seen as a further development of Leśniewski's ontology. Finally, the paper presents some challenges regarding the significance of tensed ontology in philosophy and information architecture.

Keywords: Ontology, A.N. Prior, time and existence, tense-logic, tensed ontology.

1 A.N. Prior on Time and Existence

Ontology is in this paper, conceived as the systematic study of being, the study of how existing entities are related, and how they can be presented. Time is obviously relevant in the context of ontology. Firstly, one wonders how the various temporal notions, such as tenses, instants and durations, are best related in an ontological framework. The attempt to represent the ontology of temporal notions was, in fact, included in the works of Jacob Lorhard, who originally coined the word ontology (see [21, 22, 23]). Secondly, the ontology of objects existing for a limited time only, raises the question of how to theoretically handle the fact that physical objects may come into being at one time and cease to exist at a later time? How should a future or a past object fit into a logical description of everything existing?

In dealing with the first of these two questions, the founder of modern tense-logic, A.N. Prior (1914-69), presented four theories on the relations between the "A-notions" (e.g. past, present, future) and the "B-notions" (e.g. before, after, simultaneous with). This is regarded as one of Prior's most important contributions to the understanding of the fundamental ideas of time itself (see [19, 20]). As it turns out, this work is closely related to the basic questions regarding the general assumptions of temporal logic (see [4]).

This paper focuses on the second of the two aspects of temporal ontology mentioned above; i.e., we shall present and discuss Prior's analysis of important positions regarding time and existence. Prior's analysis is interesting, not only from a

historical point of view, but also in the context of modern ontology, as an important philosophical and conceptual background. According to Prior the study of time and existence is an important, but also rather complicated, field within the study of tense-logic. In fact, he stated that this is “the untidiest and most obscure part of tense-logic” [9:172].

Prior's ideas on time and existence were hard to understand for the contemporary philosophers who surrounded Prior. Jack Smart, an Australian philosopher of science and personal friend of Prior, who differed in his fundamental belief on time, struggled to follow Prior as they corresponded together in 1954 about ontological issues concerning time and existence.¹ Where Prior believed in a dynamic view on time, Smart believed in a static theory of time; and argued, already in 1949, against the passage of time (see [16]). In a letter dated Nov. 15, 1954, Smart states that he struggles to follow Prior in his development of tense-logic.

Prior had send Smart some pages with logical formalism about which Smart says, “[it] fills me with admiration of your brainpower,” and Smart reports that he “follows [Prior's] theory (with difficulty) and still can't get a grasp of, as you say, why it works.” From the next letter, dated Nov. 17, 1954, it is evident that Smart seems to be struggling with Prior's idea that the notion of ‘sempiternal’ cannot be kept out of what, he called, quantified tense-logic. Smart writes:

To say that Prior is sempiternal is to say he has always existed and always will, and this is false because he was born in 1914. Do you mean to say that ‘Prior didn't exist before 1914’ can't be translated into quantified tense logic? Surely not. So what do you mean by saying that the sempiternality of Prior is implied by quantified tense logic? [Letter from Smart Prior Nov. 17, 1954, The Prior Collection, Box 3]

From this quote one sees how Smart finds it hard to understand what quantified tense-logic is incapable of handling, specifically in comparison to Prior's tense-logic formalism. Prior argues that it should be accepted, “there exists now an x , such that it has been the case that x is not alive”. On the other hand, it does not seem to be acceptable that “it has been the case that there exists an x , such that x is not alive”. The problem is how these two statements, (E1) and (E2), should be represented formally in terms of Prior's tense-logical formalism. One option would be:

$$(E1') \exists x: P \sim a(x)$$

$$(E2') P \exists x: \sim a(x)$$

where $a(x)$ stand for the statement “ x is alive”, and where $\exists x:$ is assumed to stand for “there is (now) an x such that”. What Prior and Smart seem to struggle with in their letters apparently has to do with the following interpretation of P , which we may call the assumption of temporal quantification:

(TQ) Pq stands for “there is some time before now at which q is the case”. It can formally be expressed as $\exists t < t_0: q$

¹ There is a large number of letters from their correspondence kept in the Prior collection, Bodleian Library, Oxford.

However, using (TQ) on (E1-2) we find

$$(E1'') \exists x: \exists t < t_0: \sim a(x)$$

$$(E2'') \exists t < t_0: \exists x: \sim a(x)$$

The problem is that (E1'') and (E2'') will be logically equivalent, given standard quantification theory. Therefore if one want (E1'') one must also accept (E2''), i.e. that x exists even when he is not alive. For this reason, Prior finds arguments like the one above, including what we have called (TQ), leads to the sempiternality for beings in time. It turns out there are several other related arguments to consider concerning time and existence. Prior points-out that some of these problems had, in fact, already been discussed by earlier logicians, such as Peirce. Prior analysed these earlier findings and contributed significantly to a deeper understanding of the problems.

2 The Logic of Past and Future Objects

C.S. Peirce's work strongly influenced Prior's philosophy. In Prior's account of "Tense and Truth in the History of Logic" included in his *Time and Modality* [8], he devoted a substantial part to the discussion of Peirce's ideas. Among other things he considered Peirce's examination of the future suicides in New York. Peirce wrote:

Again, statisticians can tell us pretty accurately how many people in the city of New York will commit suicide in the year after next. None of these persons have at present any idea of doing such a thing, and it is very doubtful whether it can properly be said to be determinate now who they will be, although their number is approximately fixed. [7:4.172]

The point is, although Peirce believes it is true that someone in New York during the year-after-next will commit suicide, it is, nevertheless not true to say that any specific individual in New York will commit suicide in that same time period. According to Peirce's view this may be explained in the following way:

A certain event either will happen or it will not. There is nothing now in existence to constitute the truth of its being about to happen, or of its being about not to happen, unless it be certain circumstances to which only a law or uniformity can lend efficacy. [7:6.368]

The point is that there is nothing now in existence on which one could base the truth of the claim that a particular individual is going to commit suicide in the year-after-next. There may be, however, sociological laws on which we can base the claim that someone in New York will commit suicide in the year-after-next. The underlying logic of this kind of reasoning allows the combination of the following two statements:

P1. It is true that at some time in the future there is an individual New Yorker who commits suicide.

P2. It is false that there is an individual New Yorker who at some time in the future is going to commit suicide.

In principal, P1 posits that, at some time in the future, a New Yorker commits suicide. Clearly, it could be that this person is not now a New Yorker; in which case, it is obvious why P2 should hold. The same point is illustrated in the following examples:

P3. It is true that, at some time in the future, there is an individual who is elected as the first female president of USA.

P4. It is false that there is an individual who, at some time in the future, is going to be elected as the first female president of USA.

We may hold that P3 is true because of some sociological law or regularity, and still it may be the case that no existing female is a future American president. In short, it seems we should be looking for a logical and ontological position according to which we may reject the following implication:

If in the future, there exists a x such that p ,
then there exists a x such that in the future p .

However, it turns-out that if the kind of reasoning presented here is transformed into a formal, tense-logical system, then one can prove that this implication is a logical law (in fact it is one of the so-called Barcan formulae) – given that a few apparently straight-forward assumptions are taken for granted.

In the following we present the standard, tense-logical setup and discuss what possible positions can be taken when addressing this conceptual problem, regarding time and ontology.

3 The Barcan Formulae

According to Prior, a theory regarding objects whose existence is limited in time must be formulated within the framework of a tense-logic, based on the primitive tense-operators H and G ; its axiomatisation is often formulated in terms of the derived operators P and F (defined as $\sim H\sim$ and $\sim G\sim$, respectively). A very fundamental system has been named K_t -where the 'K' is probably in honour of Saul Kripke. This tense-logic can be presented as an axiomatic system, with the following axiom schemes (see [9:176; 5:17 ff.]):

- (A1) p , where p is a tautology of the propositional calculus
- (A2) $G(p \supset q) \supset (Gp \supset Gq)$
- (A3) $H(p \supset q) \supset (Hp \supset Hq)$
- (A4) $PGp \supset p$
- (A5) $p \supset GPP$

In this system, p and q are arbitrary, well-formed formulas. All axioms are said to be immediately provable, while other theses can be proved by inference. In K_t , Modus Ponens is the basic rule of inference:

- (MP) If $\vdash p$ and $\vdash p \supset q$, then $\vdash q$.

In addition we have two rules, introducing tense-operators:

$$\begin{array}{ll} \text{(RG)} & \text{If } \vdash p, \text{ then } \vdash Gp. \\ \text{(RH)} & \text{If } \vdash p, \text{ then } \vdash Hp. \end{array}$$

K_t is often presented as the most basic tense-logical system, in the sense that it is hard to imagine a simpler system. If this is, in fact, the case, all other tense-logical systems should be defined as extensions of K_t . This means that they can be introduced by the addition of further axioms to the above list, (A1-5), and by the addition of more operators.

Within K_t , it is possible to prove a number of interesting theorems. One of them is:

$$\text{(T6)} \quad H(p \supset q) \supset (Pp \supset Pq)$$

As the system has been presented so far, K_t is propositional. However, in order to deal with objects existing within a limited time, we have to add quantification over objects. This means that we not only need a propositional logic, but also a tense-logical predicate calculus. The classical quantification over objects can be brought in through the following two rules, in which, the universal quantifier is introduced:

$$\begin{array}{ll} \text{(PI1)} & \text{If } \vdash \phi(x) \supset \beta, \text{ then } \vdash \forall x: \phi(x) \supset \beta. \\ \text{(PI2)} & \text{If } \vdash \alpha \supset \phi(x), \text{ then } \vdash \alpha \supset \forall x: \phi(x), \text{ for } x \text{ not free in } \alpha. \end{array}$$

These rules correspond to the following rules, expressed in terms of the existential quantifier:

$$\begin{array}{ll} \text{(\Sigma1)} & \text{If } \vdash \phi(x) \supset \beta, \text{ then } \vdash \exists x: \phi(x) \supset \beta, \text{ for } x \text{ not free in } \beta. \\ \text{(\Sigma2)} & \text{If } \vdash \alpha \supset \phi(x), \text{ then } \vdash \alpha \supset \exists x: \phi(x). \end{array}$$

Prior demonstrated that, in this system of tense-logic with quantification, one can establish the following proof:

$$\begin{array}{lll} (1) & Gq \supset Gq & \\ (2) & \forall x: Gq \supset Gq & (1 \text{ and PI1}) \\ (3) & H(\forall x: Gq \supset Gq) & (2 \text{ and RH}) \\ (4) & P \forall x: Gq \supset PGq & (3, \text{MP and T6}) \\ (5) & P \forall x: Gq \supset q & (4 \text{ and A4}) \\ (6) & P \forall x: Gq \supset \forall x: q & (5 \text{ and PI2}) \\ (7) & G(P \forall x: Gq \supset \forall x: q) & (6 \text{ and RG}) \\ (8) & GP \forall x: Gq \supset G \forall x: q & (7, \text{MP, and A2}) \\ (9) & \forall x: Gq \supset G \forall x: q & (8 \text{ and A5}) \\ (10) & F \exists x: q \supset \exists x: Fq & (9) \end{array}$$

(10) is one of the so-called Barcan formulae first discovered by Ruth Barcan. The other traditional Barcan formula,

$$(11) \quad P \exists x: q \supset \exists x: Pq$$

can be demonstrated in a similar manner.

It should be emphasized that the propositions used above are, in fact, predicates applied to individual name-variables. This means that a more precise way of expressing the content of the Barcan formulae would be the following:

$$(10') \quad F\exists x:q(x) \supset \exists x:Fq(x)$$

$$(11') \quad P\exists x:q(x) \supset \exists x:Pq(x)$$

If we assume that 'existence' is understood as 'present existence', as opposed to past or future existence, then the acceptance of (10') and (11') give rise to a problem: If an object with the property q will exist in the future, or if it existed in the past, then the object exists now with the property corresponding to Pq (or Fq). In this way, it appears that past or future existence implies present existence. Clearly, if we want to make a clear distinction between what actually exists (now), as opposed to what does not exist now (although it might have existed earlier or might come into existence later); then the Barcan formulae certainly lead to a problem – at least the existential quantifier in (10') and (11') does not seem adequate to provide such a distinction.

4 Quantifying over Possible Objects

The Barcan formulae don't cause a problem for tense-logic if (10') and (11') are accepted on the grounds that the kind of quantification involved in these formulae do not refer to actual existence, but only to possible existence. This position was defended by Niko Cocchiarella, who accepts the use of individual name-variables, even when these names are what Prior called, "now empty". Prior presented this view in the following way:

For instance, in the tensed predicate calculi of Cocchiarella it is boldly ruled that x , y , and z are the particular individuals they are even before and after they exist, and he has quantifiers over the whole bunch of them at all times. Identifiable individuals thus conceived can of course come into existence, and be brought into existence too ..." [9:158]

As Prior pointed out, Cocchiarella's metaphysics may be conceived as involving the assumption of a kind of "waiting room" for possible individuals which do not yet exist (see [9:158]). These individuals may come into existence by themselves, or they may be brought into existence by somebody else. When these individuals have been in the "existence room" for some time, they may cease to exist; i.e., they may be transferred into another room (or they may be in the same "waiting room", where they were kept before the coming into existence).

Obviously, these metaphorical statements regarding various "rooms" are not fully satisfactory from a philosophical point of view. It turns out that some additional machinery is needed to deal precisely with the notion of present existence. In fact, Cocchiarella introduced an undefined, restricted quantifier to define what 'x actually exists' means. (See [9:159]).

It is worthwhile to consider the metaphysical issues behind whether or not one wants to allow non-existent, as well as existent, individuals to be values of bound variables. In Prior's metaphysics the reasons for his dissatisfaction with a view of

logic which quantifies over merely possible entities, are tied-up with two, distinct views on time and existence. The first is the view that

[the real and the present] are closely connected; indeed on my view they are one and the same concept, and the present similar *is* the real considered in relation to two particular species of unreality, namely the past and the future. [10:245]

That the present is the real does not, by itself, entail that there aren't *facts* about singular future or past objects. And, hence, it doesn't necessarily make quantifying over singular, future or past, objects problematic. It only becomes so if one adds a premise that either limits the ontology to only contain facts about actual objects, or to contain a certain essential fact about actual objects that future and past objects necessarily cannot have.

Prior's second metaphysical commitment is that in every possible world in which *x* exists, there is a proposition about '*x*,' in a demonstrative mode. An example of the demonstrative Prior uses in *Past, Present and Future* (1967) is G.E. Moore's "This exists." The meaning of "*x* exists" is thus given by the demonstrative "This exists".

This commitment makes it necessary that *x* must exist in order for there to be facts about *x*. This commitment resonates deeply with Prior's understanding of what a sentence is, and what it means to name something in a proposition. In an unpublished paper called 'Names of names', Prior says about the use of demonstratives that

... a word like 'This' or 'That', say in 'That is a unicorn', has no meaning unless the object to which it is applied is actually present and in some way indicated while the word is being used. The purpose of the word is not so much to function as a subject on its own, "representing" the thing (as words are sometimes said to do); it is rather as it were to bring the thing bodily into the sentence, so that the predicate is attached not so much to another word as to the thing itself. [Bodleian Library, The Prior Collection, Box 6]

In order for there to be truth or facts at all about *x*, like the property of being identical with *x*, *x* must *be present* to be dragged, as it were, into a sentence. In fact, he stated that the function of 'this' is to "drag some bit of the world right into what is being said" [11:147] In this regard Prior acknowledges his debt to Peirce, with regard to this view of sentences and demonstratives, quoting the following:

The subjects are the indications of the things spoken of, the predicates, the words that assert, question or command whatever is intended. Only, the shallowness of syntax is manifest in its failing to recognise the impotence of mere words, and especially of common nouns, to fulfil the function of a grammatical subject. Words like *this*, *that*, *lo*, *hallo*, *hi there*, have a direct, forceful action upon the nervous system, and compel the hearer to look about him; and so they, more than ordinary words, contribute towards indicating what the speech is about. [11: 147]

What it means to bring objects "bodily into the sentence", is arguably a bit obscure; but it is quite clear that such a view of reality must, given a tensed view of time where only the present is real, reject quantifying over possible beings. Plantinga [14] argues that Prior's view on existence is problematic. How are we to understand the

statement: “ x might not have existed”, if we can't quantify over possible objects? It turns out that Prior cannot take it to mean

(S1) It could have been that (it is not the case that (x exist))

S1 can be shown to be a necessarily false proposition, given Prior's view (and, arguably, also Peirce's), and a straightforward understanding of “ p is possible” as “ p is possibly true”. This can be seen if we translate S1 in terms of possible worlds:

(S1') There is a possible world, in which it is the case that x does not exist

For how could x be brought “bodily into a sentence” in a world in which x doesn't exist? In consequence, Prior cannot represent the statement “ x might not have existed” in this way. Instead he takes it to mean

(S2) It is not the case that (it is necessary that (*this* x exist))

In other words, the idea is that one may take “ x might not have existed” to mean “it is possible, that it is not the case that *this* x exists”. The claim then is that there aren't any facts about x , which would make the existence of x necessary.

Clearly the distinction between (S1) and (S2) is somewhat tricky and problematic. As we shall see below, it will require at least some additions to the formalisation. Obviously, it can still be debated whether Prior's solution should be accepted; or whether we should rather prefer a quantification over possible individuals, as suggested by Plantinga [14].

5 Asymmetry between the Past and the Future

One of the solutions to the problem we are facing is based on the introduction of an asymmetry between the past and the future operators. In fact, according to Prior, A.J. Kenny maintained that the naming of past individuals is easier (i.e. less problematic) than the naming of future individuals. This may be related to the view that the future is indeterminate, whereas the past is determined (see [9: 172]). Given this view it is likewise unproblematic to refer to the qualities and the properties of past object in the present discussion; whereas references to future objects in the present discussion are more problematic. For this reason one may see past objects as available in the present discussion; whereas the future objects are not available in the same sense. This means that we may want a tense-logic which accepts (11'), but rejects (10'). But how can such asymmetry be obtained, given the proof presented in section 3? The point is that, although the Barcan formulae may hold for the F operator included in the K_t system, there may be other, and perhaps more relevant future operators for which the formulae do not hold. In fact, the future operator in Prior's so-called Peircean tense-logic is different from $\sim G\sim$ and is designed to deal with the idea of an indeterministic and undetermined future. In this system the Barcan formulae will not be valid in terms of this Peircean future operator. In fact, K_t theorems, like $p \supset HFP$, are rejected in this system.

Although this possible solution, based on the asymmetry of the past and the future, has some attractions to it. It is, however, still an open question whether it will, in fact, solve all the philosophical problems generated by the Barcan formulae and similar results.

6 Towards a Theory of Tensed Ontology

Inspired by Leśniewski's ideas, Prior worked with a theory he called tensed ontology. According to this theory, we should not allow facts about individuals. This means that in expressions such as ϕa , the a should, in general, not stand for an individual name, but, rather, for a common name. Prior points-out that, if this approach is taken, there will be "no Russellian individual name-variables at all, bound *or* free, but only devices for referring to individuals obliquely" ([9: 173]). In this way, it is, in fact, possible to keep standard tense-logic, as well as standard quantification theory, without running into the problems mentioned above, in relation to Barcan's formulae. However, Prior also showed that the price for doing so would be the acceptance of a distinction between those operators forming complex propositions and those forming complex predicates, from which one can form complex propositions. Prior considered the following statements

- (12) 'For some a (it will be that (the a is a b))'
- (13) 'It will be that (for some a (the a is a b))'
- (14) 'For some a , the a is a *thing-that-will-be-a-b*'

In the theory of tensed ontology suggested by Prior, (12) and (13) will be equivalent. Formally, this equivalence is similar to Barcan's formula concerning future existence. However, in this case the equivalence will not be philosophically problematic. Instead, the important point is that (13) and (14) are not equivalent. This means that it does not follow from the claim that an object of a certain kind will exist at some future time that there actually exists an object which is going to be an object of the particular kind in question.

In terms of the formalism used in Prior's tensed ontology the above statements can be formulated in the following way:

- (12') $\exists a: F(\mathcal{E}(a,b))$
- (13') $F(\exists a: \mathcal{E}(a,b))$
- (14') $\exists a: \mathcal{E}(a,f(b))$

Here the statement $\mathcal{E}(a,b)$ is read "the a is a b ". One very special term is V , which stands for "object". The statement $\mathcal{E}(a,V)$ then means "the a is an object". The statement $\mathcal{E}(a,V)$ means that ' a ' actually (presently) exists. According to Prior, this form should be conceived as equivalent to $\exists b: \mathcal{E}(a,b)$, i.e., as the statement "there is some common noun b , such that the a is a b ". In this way it appears that he comes close to the idea that existence can be understood as a predicate. In fact, this idea wasn't foreign to Prior. In "On some proofs of the existence of God", published by Kenny in 1976, in *Papers in Logic and Ethics*, Prior refutes a version of the ontological argument for God's existence referring to a point that, in essence, is the

same he gives against the Barcan formulae. From his correspondence with Henrik von Wright, we can reasonably assume that the paper was prepared for a meeting of *The Philosophical Society* in 1956. In a letter to Prior, dated 18 May 1956, von Wright argues that Prior's point "about the fallacious commutation of quantifier-like/operators in Anselm's proof [is] very nice"; and he stated that the ontological argument for God's existence "cannot be refuted without saying something about the status of existence as a predicate" [Bodleian Library, The Prior Collection, Box 6]. It seems that Prior disagreed. The problem Prior saw in the ontological argument for God's existence wasn't connected to whether existence is a predicate or not. Actually, he was not too certain about the precise nature of the concept of (present) existence:

We take it for granted nowadays that we have Existence properly tied up and put in a bag, but I don't know. [12: 61]

In Prior's opinion there is still a lot to do in order to understand the notion of existence in a satisfactory manner. However, he believed that it would be useful to develop the ε -calculus as much as possible. He pointed-out that we, in order to do so along with f , would need some other predicate forming operator: n (not), p (past), h (has always been), g (will always be). This means that we may, for instance, form an this expressions like $\varepsilon(a,nh(V))$, which stands for "the a is a thing-that-has-not-always-been-an-object", (i.e., that the a is something that has come into being.) However, things can quickly become complicated. For instance, we observe that the formalism suggested here would allow expression like $\varepsilon(a,n(V))$ and $\varepsilon(a,pn(V))$ as standing for "the a is a thing-that-is-not-an-object" and "the a is a thing-that-has-been-not-an-object", respectively. Intuitively, such statements must be false, since there cannot be anything which is a thing-that-is-not-an-object. However, if this is accepted, we have to conclude that $\varepsilon(a,nh(V))$ and $\varepsilon(a,pn(V))$ are not equivalent. This means that although the counterpart for propositional tense-operators fulfills the rule $\sim H \equiv P\sim$, we do not have $nh \equiv pn$ for tense-logical predicate operators. There are axiomatic systems dealing with the ε -calculus ([9: 164 ff]; but it is still an open question how precisely a calculus of tensed ontology with tensed predicate operators should look like. This will be a task for further research. However, anyone who would study this problem in detail will certainly benefit from Prior's preliminary study in the field; which, at least, indicates that it is meaningful to seek a formal calculus of tensed ontology. One obvious challenge for further development of the ε -calculus is bringing a relevant version of Prior's Q-system into the account ([8: 41 – 54] and [9: 54 - 58]). In this way it may be possible to deal with the very simple but very important fact that the set of storable nouns (and other expressions in formal language) is growing as time passes, i.e., many nouns and other expressions which are now storable were not storable earlier.

7 Time and Existence in Modern Ontologies

In modern ontology, the question of tensed ontologies is rarely addressed. Instead, it is assumed the matter of tense is to be treated as a linguistic phenomenon rather than an ontological one. In consequence, time is typically formally described in terms of instants and intervals, with emphasis placed on measurements and order,

corresponding to the B-notions of before, after and simultaneous with. A prototypical example of this is the *W3C Time Ontology in OWL*, which considers exactly two subclasses of *TemporalEntity*, namely: *Instant* and *Interval* [6]. Relations between *TemporalEntities* are describes in terms of properties of the temporal objects. This framework allows for handling of durations, including the 13 Allen/Hayes distinctions [1]. But the ontology is silent about the notions of Past, Present, and Future and will therefore be forced to represent things that are not yet, on a par with what is not the case.

In systems such as the *Universal Networking Language effort (UNL)*, emphasis is placed on the formal representation of natural language utterances [18]. Here, a distinction is made between *Time Attributes* and *Time Relations*. Attributes are functions that take a single argument (predication), and may be considered as absolute (past, present, future, recent, remote), or as relative (anterior, posterior), while *Time Relations* link more than one node and allow to expressing the following relationships: time from, time to, from to, and duration. In this way it becomes possible to represent both A- and B-logical perspectives on temporal phenomena. The focus on linguistic expressions will, however, inevitably lead to conflict between natural language semantics and formal ontology. Thus, the UNL system addresses the matter of the semantics of its terms to contextual definitions.

A third route to take is to acknowledge the distinction between actual and non-actual entities, and to describe the consequences of the choices that must necessarily be made regarding their representation. In [2] Pierre Grenon has argued that the metaphysics of time is relevant as a philosophical and conceptual background in the study of formal ontology. He has demonstrated that there are two possible doctrines on which a system of formal ontology can be based:

1. eternalism i.e. the view that “all things exist on a par for all eternity” (p.2),
2. presentism i.e. “ the doctrine according to which all entities which exist, exist at the present time” (p.44).

Grenon has stressed that these doctrines are meta-ontological. This means that the choice between them has to be made outside the scope of formal ontology itself. On the other hand, it is obvious that the choice between the two basic doctrines has a very clear impact on how we should deal with crucial problems in formal ontology such as the formal references to non-actual entities (p.8) and the formal representation of the realm of continuants in time (p.44).

Grenon has also pointed-out that his theory might be expanded, making use of some “apparatus of a tense-logical sort” (p.8) and perhaps even of the idea of branching time (p.60). In this way he has indicated that the further development of temporal ontology might benefit from the study of the kind of temporal logic founded by A.N. Prior. The relevance of Prior's temporal logic within formal ontology is also evident from in the *Temporal Database Entries for the Springer Encyclopedia of Database Systems* edited by Christian S. Jensen and Richard T. Snodgrass [3]. This work introduces a number of important aspects and perspectives of conceptual background which may interest researchers in computer science who want to construct temporal databases.

8 Conclusion

As we have seen, the question of eternalism versus presentism is a theme within the modern study of formal ontology. When dealing with the philosophical and conceptual aspects of this question, Prior's analysis can certainly be useful. Prior explained how a logical analysis like his can be used by researchers working with questions on time within specific fields:

The logician must be rather like a lawyer... in the sense that he is there to give the metaphysician, perhaps even the physicist, the tense-logic that he wants, provided that it be consistent. He must tell his client what the consequences of a given choice will be ... and what alternatives are open to him; but I doubt whether he can, qua logician, do more. [9: 59]

As we have seen, Prior demonstrated how eternalism and presentism are both possible answers to the basic philosophical question regarding the relation between time and existence. However, in both cases there is a price to pay:

- 1) Eternalism. This solution requires that one accepts the idea of time as a tapestry, with past, future and present individuals existing on a par. Such an ontology goes against our intuitions that becoming is a fundamental part of reality. On the other hand, on this view, one has no problem with quantification over non-present individuals, hereby also avoiding the "waiting room problem".
- 2) Presentism. According to this solution, we must restrict quantification to nouns, or otherwise deal with the "waiting room problem". If one will not accept quantifying over none present individuals as possibilities, then the challenge consists of developing a calculus corresponding to Prior's ε -calculus. As shown above this approach gives rise to a number of formal and conceptual problems.

An analysis of the modern ontologies suggests that we are compelled to choose between 1) and 2). The present study indicates that researchers in modern ontology can benefit greatly by closely examining the insightful work of those working in this field in the 1950s and 1960s – especially Prior.

Acknowledgments. We are grateful to Dr. Mary Prior and to Bodleian Library for giving us access to the Prior collection in Oxford. We also want to thank Claude Louis Chappuis for useful comments on an earlier version of this paper.

References

1. Allen, J.F., Hayes, P.: A Common-Sense Theory of Time. In: Proc. of the Ninth Int. Joint Conf. on Artificial Intelligence, pp. 528–531 (1985)
2. Grenon, P.: Spatio-temporality in Basic Formal Ontology SNAP and SPAN, Upper-Level Ontology, and Framework for Formalization, PART I, Final Version, November 2003, Universität Leipzig, Faculty of Medicine, Institute for Formal Ontology and Medical Information Science (IFOMIS), (May 2003) ISSN: 1611-4019

3. Jensen, C.S., Snodgrass, R.T. (eds.): Temporal Database Entries for the Springer Encyclopedia of Database Systems: A TIMECENTER Technical Report, Aalborg University & University of Arizona, USA (2008), <http://timecenter.cs.aau.dk/TimeCenterPublications/TR-90.pdf>
4. Kachi, D.: Tensed Ontology Based on Simple Partial Logic. In: Proceedings of the Ninth International Symposium on Temporal Representation and Reasoning (TIME 2002), pp. 141–145 (2002)
5. McArthur, R.P.: Tense Logic. Reidel, Dordrecht (1976)
6. OWL Time, <http://www.w3.org/TR/owl-time>
7. Peirce, C.S.: Collected Papers. In: Weiss, P., Burks, A., Hartshorne, C. (eds.), vol. 8. Harvard University Press, Cambridge (1931-1958)
8. Prior, A.N.: Time and Modality, Oxford (1957)
9. Prior, A.N.: Past, Present and Future. Clarendon Press, Oxford (1967)
10. Prior, A.N.: The Notion of the Present. In: Prior, M. (ed.) *Studium Generale*, vol. 23, pp. 245–248 (1970)
11. Prior, A.N.: *Objects of Thoughts*, Oxford (1971)
12. Prior, A.N.: On Some Proofs of the Existence of God. In: Geach, P.T., Kenny, A.J.P. (eds.) *Papers in Logic and Ethics*. University of Massachusetts Press (1976)
13. Prior, A.N.: *Papers on Time and Tense*. In: Hasle, P., et al. (eds.) Oxford University Press, Oxford (2003)
14. Plantinga, A.: On Existentialism. *Philosophical Studies* 44, 1–20 (1983)
15. Smart, J.J.C.: Letter to A.N. Prior. The Prior Collection, Bodleian Library, box 3, Oxford (September 3, 1958)
16. Smart, J.J.C.: The river of time. *Mind* 58(232), 483–494 (1949)
17. Von Wright, G.H.: Letter to A.N. Prior. The Prior Collection, Bodleian Library box 3, Oxford (September 3, 1958)
18. UNL Web, <http://www.unlweb.net/wiki/index.php/Time>
19. Øhrstrøm, P., Hasle, P.: *Temporal Logic - From Ancient Ideas to Artificial Intelligence*. Kluwer Academic Publishers, Dordrecht (1995)
20. Øhrstrøm, P., Schärfe, H.: A Priorean Approach to Time Ontologies. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 388–401. Springer, Heidelberg (2004)
21. Øhrstrøm, P., Andersen, J., Schärfe, H.: What Has Happened to Ontology. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 425–438. Springer, Heidelberg (2005)
22. Øhrstrøm, P., Uckelman, S.L., Schärfe, H.: Historical and Conceptual Foundation of Diagrammatical Ontology. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 374–386. Springer, Heidelberg (2007)
23. Øhrstrøm, P., Schärfe, H., Uckelman, S.L.: Jacob Lorhard's Ontology: A 17th Century Hypertext on the Reality and Temporality of the World of Intelligibles. In: Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 74–87. Springer, Heidelberg (2008)

Crowdsourced Knowledge: Peril and Promise for Conceptual Structures Research

Mary Keeler

mkeeler@uw.edu

Abstract. Recent efforts to create natural-language, question-answering systems for the World Wide Web exploit the vast availability of information in Web resources as a knowledge base. Researchers who develop these systems explicitly assume that what is most often repeated in that knowledge base is the truth. Their assumption implicitly relies on a fallacy in classical logic: “proof by assertion” (or proof by repeated assertion). This paper considers the implications—both hazardous and hopeful—of the Most Often Repeated (MOR) assumption, and suggests Peirce’s “economy of research” (EOR), in his evolutionary view of logic, as a promising alternative to MOR for truth-finding in the increasing complexity of crowdsourced knowledge.

1 Introduction

“Crowdsourcing” on the World Wide Web (WWW) is becoming a powerful methodology for everything from finding new stars in the galaxy [1] to revolutionizing governments [2]. Employers can even crowdsource a labor force for “human intelligence tasks” [3]. In the article “Everyone’s an Expert: The Crowdsourcing of History” in *Data Conservation Laboratory News*, M. Ojala reports: “With sites like *Wikipedia* relying on expertise provided by a vast community of internet users, crowdsourced knowledge is now something people rely on every day.” Ojala points out that the crowdsourcing phenomenon predates the Internet, and can be as simple as yelling a question in a large crowd, which may by chance return the correct answer—if you are prepared to sift through responses. “Crowdsourcing is now changing the way we think about knowledge, expertise is no longer the exclusive domain of experts” [4]. Social media now make it possible to correct an article’s facts or write an argument in a blog post. The “Flickr Commons” project [5] incorporates this power in the identification of items in the photo collections of 30 participating institutions (including the Library of Congress, the Getty Research institute, and the British Museum) to crowdsource the expertise of ordinary people. Validating answers to questions on social media is also a public process, which these institutions have learned is time consuming, but they conclude that “greater understanding of our shared past” makes the endeavor worthwhile [4].

Meanwhile, the vast availability of information on the WWW has inspired the goal of “open domain question answering systems” (QA), capable of responding to a natural language question with a short natural language response. The QA technology race to develop more efficient use of “crowd-sourced data” began with systems that integrated information retrieval (IR) with natural language processing (NLP). Contenders are featured at Text Retrieval Conferences (TREC), the annual meeting

sponsored by the National Institute of Standards and Technology (NIST). Since 1999, TREC's question-answering track has challenged researchers to develop standardized evaluation metrics for question-answering systems, with the goal "to foster research on systems that retrieve answers rather than documents in response to a question" [6]. Researchers who developed the most famous QA system, IBM's Watson, recognize that the real challenge is to engage humans effectively in the process. Even less ambitious QA system developers explicitly make use of human participation, usually in "voting" processes. Examination of these processes reveals a variety of methods, all of which rely on a "most often repeated" (MOR) assumption. When the fallacy of MOR is brought to developers' attention, they simply respond: "What else can we do?" This paper argues that Peirce's pragmatic methodology for the "economy of research" (EOR) could be an effective alternative to MOR, which could be developed in Conceptual Structures research *as a game* to engage humans more effectively in the process.

2 Open Domain Question Answering Systems

The development of QA systems began in specialized IR systems and has integrated NLP techniques as those have advanced. Ideally, a QA would be able to answer a broad range of questions, but the unstructured nature of the WWW poses a significant challenge.

2.1 Evolution of QA Research

Traditional QA systems (since the 1960s) were interfaces to relational databases [7]. Text understanding programs came along in the seventies [e.g., 8], for parsing natural language text to create a knowledge base augmented by knowledge-engineered structures such as a situational script, a frame, or a plan. These programs produced paraphrases or answered questions to demonstrate their understanding of a text, but the extensive knowledge engineering required limited their domains to certain topics.

By 2000, researchers began to develop QA systems using resources on the WWW to create knowledge bases from which to generate answers (e.g., FAQ Finder, START, etc.). Generally, these attempted to find answers using search engines that rely on what Google indexes, and limited their scope to answering fact-based questions. Their motivation was to replace the exploding number of keyword-based lists of links with accurate answers to natural language queries on any subject. Systems were designed to scale, answering an increasing number of questions by adding more knowledge.

2.2 Sample of Recent QA Research Projects

TrueKnowledge [9], for example, "answers millions of questions [9] per month, asked by real Internet users," and increases its knowledge base typically by adding "knowledge about knowledge" [10].

MULDER [11] reformulates a question to create a specific search, then extracts the answer from search page summaries, based on information from its question classifier. Developers find that this "federated approach" suffers from the possibility

that an answer is no longer available, and its distributed approach suffers from the possibility that an answer is unreliable.

[I]t is hard to know if the answer is correct. One way of measuring the correctness of an answer is to look for multiple occurrences of an answer in different documents. The voting for an answer by multiple sources lends credibility to an answer, and allows possible answers to be ranked. In this way MULDER attempts to use redundancy as a possible alternative to structured knowledge bases and understanding. [11: 8]

MULDER's answer selection module uses the "voting procedure" to pick the best answer among candidates, first ranking them by closeness to keywords, then "clustering" similar answers together. A final ballot is cast for all clusters, the cluster with the highest score wins, and the final answer is chosen from the top candidate in this cluster, to achieve several corrections:

- **Reducing noise.** Random phrases that occur by chance are likely to be eliminated from further consideration by clustering.
- **Allowing alternative answers.** Many answers have alternative acceptable forms. ... Clustering collects all of these dissimilar entries together so that they have collective bargaining power and get selected as the final answer.
- **Separating facts from fiction.** The Web contains a lot of misinformation. MULDER assumes the truth will prevail and occur more often, and clustering embeds this ideal. [11: 9-10]

Another QA system, DART (Discovery and Aggregation of Relations in Text) was based on the conjecture, "there is a largely untapped source of general knowledge in texts, lying at a level beneath the explicit assertional content." This knowledge consists of relationships implied to be possible in the world, or, under certain conditions, implied to be normal or commonplace in the world. The system attempts "to derive a broad range of general relationships from texts, rather than some predetermined specific kinds of facts; [it uses] general phrase structure coupled with compositional interpretive rules to obtain general propositional information, rather than employing specialized extraction patterns targeted at specific relationships." Project developers concluded in 2009, "There's still a gap between constructing these things, and finding out how they can be significantly useful" [12].

2.3 Current State of the Art

Work on many QA systems (including MULDER and DART) has been suspended, perhaps because they reached their research paradigm potential, and their developers have consolidated efforts to work on a few promising systems. The Fall 2010 issue of *AI Magazine* covered six of the "most interesting" QA systems still under development: Project Halo's Digital Aristotle, Cyc's Semantic Research Assistant, IBM's Watson, the University of Washington's TextRunner, and Cambridge University's True Knowledge. The article explains why QA continues to be pursued as a fundamental capability.

The QA problem extends beyond AI systems to many analytical tasks that involve gathering, correlating, and analyzing information in ways that can

naturally be formulated as questions. Ultimately, questions are an interface to systems that provide such analytic capabilities, and the need to provide this interface has increased dramatically over the past decade with the explosion of information available in digital form. ... [Users expect] specific answers to specific questions by understanding, synthesizing, and reasoning about the underlying data, knowledge, and text documents. [They] also expect it to be able to provide user-specific explanations or justifications relating the context of the question to what the user currently knows, as well as informing the user of its confidence, especially in cases where the confidence in an answer is low. ... recently, with advances in knowledge-based systems, natural language understanding, machine learning, and text understanding across the web, we may be on the threshold of finding combinations of these techniques that achieve a convincing and useful question-answering capability that is radically different from what is available in the market today.

The most ambitious system, IBM's Watson, debuted as a contestant on the U.S. television quiz show *Jeopardy*, winning the game against two previously top-scoring humans. Chief Architect D. Ferrucci explains that the real challenge in finding correct answers is to improve the efficiency—not of finding documents, but of gathering knowledge from them, a process in which he says humans must be engaged.

The human is doing the deep reasoning, but without the technology can't get the information to reason over. Formal reasoning systems can help, but we need to be careful about what role humans really need to and want to play. We need to advance the paradigm from known-item keyword research, because it's a pipe dream that authors are going to tag up the stuff; also, because the users' view is different from authors' (in vocabulary, language, etc.), the annotation problem becomes much worse. The hypothesis here is that open semantics must *emerge*. The right analysis for the job will likely be a best-of-breed combination integrating across many dimensions. [13]

Ferrucci here refers to the Emerging Component Community at CMU [14]. Watson depends on interoperable components in a very large-scale system for assigning semantics that weighs evidence and determines the probability that an answer is right. Watson doesn't have the deep understanding or common sense that humans do; it "sees" only words and their relations.

Even less ambitious QA system developers still make use of human participation, usually programmers in a "weighted voting process" [Note 1]. Examination of these processes reveals a variety of methods that cautiously rely on the "most often repeated" (MOR) assumption that any voting involves (in ballot counting). Developers of these systems explicitly make this truth assumption, resigning themselves to the position: "What else can we do?" First of all, we can consider the perils of MOR.

3 The Most Often Repeated (MOR) Problem

A query to *Answers.com* [15] finds that the *Oxford Dictionary of Proverbs* attributes this quotation to Charles Haddon Spurgeon: "It is well said in the old proverb, 'a lie

will go round the world while truth is pulling its boots on.” *Answers.com* also explains: “Most say Mark Twain but Charles Hadden Spurgeon did in 1885.” Meanwhile, *IWise* (“Wisdom on Demand”) quotes Winston Churchill: “A lie gets halfway around the world before the truth has a chance to get its pants on” [16]. Other Web sources mention Shakespeare as the possible source of a similar quotation. Claiming truth by simply repeating an assertion (identified by Aristotle as babbling) is a fallacy in classical logic theory, which has become a master rhetorical strategy in modern politics and advertising. A quick Web search suggests that either Hitler or Minister of Propaganda Goebbels is credited with the “Big Lie Theory,” used to exploit German media during the Nazi regime: “the bigger the lie, the greater the likelihood that people would believe it.” Meanwhile, “A lie, repeated often enough, will end up as truth” is most frequently attributed to Lenin [17]. And recently, S. Bales’ research on *Framing* demonstrates that “the first person to ‘frame’ an argument most often succeeds in capturing the public, and that changing a frame is extraordinarily difficult” [18]. One of the biggest current political lies, about “weapons of mass destruction” (often repeated, and still believed by most), was finally exposed by its source [19].

3.1 MOR in Web Journalism

The WWW is now the medium of choice to propagate the MOR strategy. With the explosion of sources for knowledge on the Web, new sites have emerged that claim to find the truth—or the lies. Truth-finding can be especially complex in political issues. For example, results of a *Fox News* survey, published 19 August 2010, indicate that nearly one in five Americans thinks Obama is Muslim [20]. Searching for “Obama is Muslim” on Google retrieves “About 44,400,000 results (in 0.13 seconds).” And *PolitiFact.com*’s “Truth-o-Meter” selects a “Lie of the Year” in the U.S., which for 2010 was “A government takeover of health care” [21]. Other sites include *FactCheck.org*, a project of the Annenberg Public Policy Center, and *Truth-out.org*, “progressive journalism and commentary on the web.” Of course, these “liberal” sources have been matched by “conservative truth-finders,” such as *Conservapedia*, “An encyclopedia with articles written from a conservative viewpoint,” and *ConservativeTruth.org*, “The Antidote to Liberal News Media.”

Journalists have begun to study how political “memes” that are blatantly false get started and circulate on the Web, by following an increasing array of political blogs [22]. Reporters are especially aware of the MOR hazard as they negotiate the complexities of finding truth on the Web; some are self-critical of their industry. For example, A. Huffington says, “the WikiLeaks controversy has found a great deal of the media once again on the wrong side of the secrecy debate. As Harvard’s John Perry Barlow tweeted: ‘We have reached a point in our history where lies are protected speech and the truth is criminal.’” Here, I quote Huffington from the *Guardian* newspaper (5 February 2011) [23], but the headline was repeated in at least 25 news sources on the Web, that day: “Traditional papers didn’t know how to handle WikiLeaks” [24]. Of course, what those documents contain is too complex for most traditional media production methods, and it is easier to repeat a concise conclusion. “When distant and unfamiliar and complex things are communicated to great masses of people, the truth suffers a considerable and often a radical distortion. The complex

is made over into the simple, the hypothetical into the dogmatic, and the relative into an absolute,” warned W. Lippmann in 1955 [25].

3.2 Truth about Lies

To help sort out the complexities there are sites like *RationalWiki.org*, which announces its purpose:

- Analyzing and refuting pseudoscience and the anti-science movement.
- Documenting the full range of crank ideas.
- Explorations of authoritarianism and fundamentalism.
- Analysis and criticism of how these subjects are handled in the media. [26]

At this Website is Carl Sagan’s essay “The Fine Art of Baloney Detection,” with the instructions: “Together, the set of warning signs for common fallacies constitutes what Sagan calls a ‘baloney detection kit.’” Types of fallacy are listed, with a definition of each, and an example found on the current Internet. (It does not specifically list the MOR fallacy, but “begging the question” is a generalization.)

Recently, various social media have taken up the challenge of truth-seeking by crowdsourcing, beyond the “walls” of Wikis. *Quora* on Facebook calls itself, “A continually improving collection of questions and answers created, edited, and organized by everyone who uses it” [27]. And *TruthOrFiction* claims, “Be among the first to know about new eRumors, viruses, Internet hoaxes ... and more” [28]! This site features “eRumors” (such as emails representing “lies told by Barack Obama”), analyzes their content and then issues a “Fight the Smears” invitation for public response, through voting. We all know that many “universal truths” are no more than conjectures or simply rumors. “Most often repeated” is a perilous methodology for measuring truth, especially in the evolution of crowd-sourced knowledge, which will rely on QA systems as they become common Web utilities.

4 Peirce’s Economy of Research (EOR)

Why should we look to Peirce for help with the MOR problem? Consider R. Burch’s appraisal of Peirce in the online *Stanford Encyclopedia of Philosophy* (SEP) [29]: “given his lifelong ideas and goals as a scientist-philosopher, he probably would have found the current practical importance of his ideas entirely to be expected.” For example: “He would not be in the least surprised to find that the topic of constructing ‘ontologies’ is in vogue among computer scientists.... He would not find in the least alien many contemporary analytic discussions of the notion of similarity; he would be right at home among them.” Burch finishes his entry with a synoptic account of the many contemporary, practical and even crucial uses of Peirce’s ideas (for industry, business, intelligence organizations, and the military) in the development of algorithms at the core of what is known as “Social Network Analysis.”

4.1 EOR and Scientific Methodology

Although Peirce scholars have neglected the topic (perhaps due to difficulty of access to his later writings [29][30]), “the Economy of Research” threads through the

evolution of Peirce's work on logic, as Burch's SEP entry indicates [29][30]. EOR begins as a part of abduction, which Peirce closely identified with his pragmatism (saying that pragmatism "is nothing else than the question of the logic of abduction") [CP 5.196 (1903)]. Then, as logic takes on a more general significance in his later development of sign theory, the role of economy extends through his systematic procedure for seeking the truth in science, integrating all stages of inference: abduction, deduction, and induction [CP 7.220]. M. Fisch, editor of an edition of Peirce's writings, explains that as a practising scientist Peirce "gradually gave up conceiving science as a mode of apprehension by a single knower, or as systematized knowledge, and came to conceive it as a mode of life common to any community of investigators, and to conceive a particular science as a social group pursuing the same or closely related inquiries" [31: xxv]. We know that he concluded: "the whole service of logic to science, whatever the nature of its services to individuals may be, is of the nature of an economy" [CP 7.220 FN (1901)].

Peirce's EOR considers "the relations between the utility and the cost of diminishing the probable error of our knowledge ... how, with a given expenditure of money, time, and energy, to obtain the most valuable addition to our knowledge" [CP 7.140 (1879)]. "In effect," says Burch, "the economics of research is a cost/benefit analysis in connection with states of knowledge. Although this idea has been insufficiently explored, Peirce himself regarded it as central to the scientific method and to the idea of rational behavior" [29]. The method begins in *abduction*, to form a hypothesis from conjectures about something uncertain that would, if true, reduce our uncertainty. In the next step of the method, *deduction*, we predict what would be the consequences from the provisionally adopted hypothesis, if it were true. In the third step of *induction*, we experiment to test for evidence of whether those predicted consequences actually occur. This method then enters either of two "feedback loops," as Burch explains. If we do find evidence for the deduced (predicted) consequences occurring, "then we loop back to the deduction stage," to predict further consequences of our hypothesis and to experimentally test for them again. If the deduced consequences (our predictions) do not occur, "then we loop back to the abduction stage and come up with some new hypothesis" that explains both our original uncertainty and any new uncertainties uncovered in the course of testing the first, failed, hypothesis. "Then we pass on to the deduction stage, as before ..." [29]. In [32-34], we discuss this methodology in more detail.

Obviously, this "self-corrective" procedure could be made more efficient at each stage, and that is precisely what technology has done for scientific research. How could EOR respond to the MOR problem?

4.2 MOR as a Confidence Game

How would Peirce assess the MOR methodology in measuring truth? It has a low cost, simply a tally of similar phrase occurrences. Naïvely, it could be considered a form of induction in scientific methodology (reasoning from a sample of particular instances to general conclusions). Naïve, because a central problem of philosophy is to explain induction (thought evolving from particular experiences to general concepts)—how is it possible—in other words, how can we know anything? In his detailed work on the logic of relatives Peirce explains the problem of induction as *not*

to be fooled by observed regularity: “It is true that there is a difference between an *accidental* and an *essential* regularity. But the difference does not manifest itself in the existential facts themselves. The problem of how an accidental regularity can be distinguished from an essential one is precisely the problem of inductive logic” [CP 3.605, c.1903]. Knowledge (or the truth we draw from experience) is a *habit of regularity in thinking and representing that has proven to be reliable in the long run*. That reliability depends on distinguishing accidental from essential regularity—a habit that does not come naturally or automatically, but needs to be cultivated [32] [33] [34]. Peirce concludes, “But in induction a habit of probity is needed for success: a trickster is sure to play the confidence game upon himself. And in addition to probity, industry is essential [CP 1.576 (1902)].

P. Skagestad interprets this habit as “intellectual virtue,” which cannot be specified in the clearly formulated rules of scientific methodology, but only in terms of the ideal end we aim for in logic and in science, which is *truth*.

I take Peirce to have in mind something like the following. The formal rules of inference do not and cannot include rules for how they themselves are to be applied. The canons of induction may be applied with a conscious intent of deception—or with the less conscious intent of self-deception—and if they are so applied they will not lead the inquirer towards the truth. Similarly, the requirement of explanatory power will be easily fulfilled if applied by an investigator who is satisfied with shallow or trivial explanations; the requirement of economy will be easily fulfilled to the satisfaction of a sloppy and careless mind, and so forth. The self-correctiveness of any set of rules for scientific inference will always depend on the mental and moral character of the investigator applying the rules. [35: 193]

We devise explanations from related propositions and, as R. Hilpinen’s analysis points out, because Peirce defined the truth of a proposition as “the utterer’s ability to defend it successfully against the interpreter’s attack, ... this analysis of quantifier phrases gives quantified sentences correct truth-conditions and is essentially similar to modern game-theoretical interpretation of quantifiers” [36:268; CP 2.328].

4.3 EOR as a Crowdsourcing Knowledge Game?

J. McGonigal argues in *Reality is Broken: Why Games Make Us Better and How They Can Change the World*, that we can use games to build stronger communities and to collaborate at vast scales. In a chapter entitled “The Engagement Economy,” she reports that it has already been done.

On June 24, 2009, more than twenty thousand Britons joined forces online to investigate one of the biggest scandals in British history—investigations that led to the resignation of dozens of parliament members and ultimately inspired sweeping political reform. How did these ordinary citizens make such a big difference? They did it by playing a game. [37: 219].

When leaked documents revealed that dozens of members of parliament (MPs) had been filing illegal expense claims, adding up to tens of thousands of pounds, the public was outraged and demanded a full account. The government then “dumped the data” of more than a million expense forms and receipts in the unuseful format of

scanned image files. Editors of the *Guardian*, who knew their reporters could never sort through the data, decided to develop a game to crowdsource the public's help. *Investigate Your MP's Expenses* became "the world's first massively multiplayer investigative journalism project," and a great success.

Just three days into the game, it was clear that the crowdsourcing effort was an unprecedented success. More than 20,000 players had already analyzed more than 170,000 electronic documents. Michael Andersen, a member of the Nieman Journalism Lab at Harvard University and an expert on Internet journalism, reported at the time: "Journalism has seen crowdsourcing before, but it's the scale of the *Guardian's* project—170,000 documents reviewed in the first 80 hours, thanks to a visitor participation rate of 56 percent—that's breathtaking. [37: 222]"

For comparison, McGonigal reports "roughly 4.6 percent of visitors to Wikipedia make a contribution to the online encyclopedia." The difference in motivation is the sense of moral "self-correctiveness" that a democratic system requires, but usually fails to facilitate. In fact, the Obama administration has recently approached Microsoft to create a game to explain the difficult realities of spending cuts [38], another complex issue for the "Serious Games Initiative," where developers focus on "the growing application of videogames and videogame technologies for purposes outside of commercial entertainment," for use in "education, training, health and public policy" [39].

McGonigal's account tells us of a game to engage collaborators for the purpose of improving a situation by constructing knowledge that would not be possible without their research—and the technology to make that research possible. An EOR game would augment not only their access to evidence but also their constructive reasoning; see for example [33][34]. Peirce's pragmatic theory of research defines "habits of thinking" as beliefs, and distinguishes reasoning from believing as the self-corrective thinking required in learning to improve habits of thought.

The pragmatic role of logic is to improve, or economize, reasoning in evolving effective habits of thought, or knowledge. Knowledge evolution conceived in the three stages (abduction, deduction, and induction) by no accident corresponds to natural evolutionary theory, in stages of diversity, selection, and adaptation. As Burch says, in Peirce's view, "The world, as it were, evolves by abducing, deducing, and inducing itself," and the evolution of knowledge is an extension of natural evolution [29]. Peirce revised his ideas many times and developed an intricate, mathematical form of EOR [e.g., *CP* 7.139-157 (1879)], all to maximize the reduction of indeterminacy in evolving knowledge by applying logic in research.

5 Perspectives and Promise

Peirce's motivation for his EOR was to improve the scientific method, which he came to regard as a method for improving reasoning in general. Our natural intuitions and practical reasoning have evolved for our survival in the natural environment; but we now live in an increasingly symbolic (or virtual) environment, as we describe in [40]. Peirce also came to understand the natural human urge to generalize as a sort of "nominalist game," explained in [33]. When we successfully use the scientific method for reasoning, we learn "to play the game of explaining how the world is"—

while remembering that it *is a game* (of finding out which representations are probably true) [Note 3]. Many of us need to be engaged, each contributing views, to make the game work in improving knowledge. Like the QA system Watson, we can predict but are never sure of the truth—unlike systems that conveniently assume truth to be what is MOR, we need not be fooled by a confidence game. Predicting is always subject to risk, but even more so in our highly complex, symbolic world, where we can no longer depend on “common sense” and instincts that are vulnerable to other common fallacies, as N. Taleb identifies:

Narrative Fallacy: Our need to fit a story or pattern to a series of connected or disconnected facts. The statistical application of this is data mining.

The fallacy of silent evidence: Looking at history, we do not see the full story, only the rosier parts of the process.

Confirmation error: You look for instances that confirm your belief, your construction (or model)—and find them.

Round-up fallacy: Confusing absence of evidence for evidence of absence.

Ludic Fallacy: A manifestation of the Platonic fallacy in the study of uncertainty; basing studies of chance on the narrow world of games and dice ... the bell curve (Gaussian) ... is the application of the ludic fallacy to randomness. [41:302-304]

Statistical regress (or circularity of statistics): We need the data to tell us what probability distribution to assume; but we need a probability distribution to tell us how much data we need. This causes a severe regress argument (which is somewhat shamelessly circumvented by resorting to the Gaussian and its kin).

Taleb summarizes the weaknesses of instinctive reasoning:

We tend to use different mental machinery—so called modules—in different situations: our brains lack a central all-purpose computer that starts with logical rules and applies them equally to all possible situations ... By the mental mechanism I call naïve empiricism, we have a natural tendency to look for instances that confirm our story and our vision of the world—these instances are always easy to find. You take the past instances that corroborate your theories and you treat them as *evidence* ... Even in testing a hypothesis, we tend to look for instances where the hypothesis proved true. Of course we can easily find confirmation; all we have to do is look, or have a researcher do it for us. I can *find confirmation* for just about anything, the way a skilled London cabbie can find traffic to increase the fare ... Seeing white swans does not confirm the nonexistence of black swans... It is misleading to build a theory from observed facts ... Contrary to conventional wisdom, our body of knowledge does not increase from a series of confirmatory observations [41:54-56].

Knowledge representation technology could give us access to what Taleb says “our brains lack,” what QA systems exploit to succeed where humans fail. But, as *Financial Times* reporter Richard Waters cautions, in gaining that supplementary cognition, “we should not outsource the brain too much” [42]. Like the human brain, Watson weighs evidence and determines the probability that an answer is correct; but its evidence is entirely symbolic, so the “weighing” is of *virtual evidence*—

symbolically represented evidence (it can't even know if it's playing the MOR confidence game). The natural brain often fails us in our symbolic evolution where, meanwhile, new evidence-weighting capabilities are emerging that we have only begun to comprehend. For instance, *Patch* developers claim they “will create a premier global, national, local, and hyper-local content group ... [to] mark a seminal moment in the evolution of digital journalism and online engagement” [43]. And *TED* curator C. Anderson claims his global online video lectures will power “crowd accelerated innovation” [44]. *TED* has already spawned *JoVE*, “Journal of Visualized Experiments—a peer reviewed, indexed journal devoted to the publication of biological research in a video format” [45]. While these developments are stunning, they can be misused, even unintentionally. “A fallacy is a pitfall of reasoning that exhibits a general and recurring tendency to deceive and to deceive successfully, to trick even the entirely serious and honest arguer” [46:6]. The MOR assumption is already instituting a misleading rule for establishing truth in QA systems; but, as their developers say, what else can we do?

Instead of the MOR confidence game, we need an EOR crowdsourcing game for investigating truth in the complex evolution of knowledge, using Web technology. In his last published work, A. Burks (another editor of Peirce's writings) explains pragmatism as a logical theory of evolution, in which “inquiry is an intellectual process of adapting to the environment” by learning and discovery. “Thus Peirce saw that evolution is creative in producing a sequence of ever more complicated organisms and societies of these organisms, and that Darwinism did not fully explain this creativity” [47: 501, 506]. Although Peirce's work predated the 20th-century development of genetics as the mechanism to describe how biological evolution proceeds, his logical theory could provide a mechanism to describe how knowledge evolution proceeds. Unfortunately, he observed, we lack the form of logical representation to study that evolution, as thought “in continual mutation ... thinking in its own movement is presumably of the same nature as that which we represent by arguments and inferences, but not so representable in consequence of a defect in that method of representation” [*CP* 2.27 (1902)].

J. Sowa says Peirce was right when he called his Existential Graphs “the logic of the future” [48: 444; *MS L* 224 (1909)]. Our future work must bring Conceptual Structures technology to the task of representing knowledge as virtually living, with logic as the “genetics” of its virtual evolutionary process that engages human creativity on the WWW. Already, “Wikipediolics” (people who are addicted to editing Wikipedia articles) explain that “Wikipedia is a massively multiplayer online role-playing game” [33: 228-29]. Such “MMORPG players” need much better methodology than the “MOR confidence game” in the complex pursuit of truth in crowdsourced knowledge. Collaboratively, we can improve our investigative capabilities, both in logical precision and in comprehensive perspective — to facilitate citizen engagement in global democracy.

6 Notes

General Note: For all *CP* references, *Collected Papers of Charles Sanders Peirce*, 8 vols., edited by Arthur W. Burks, Charles Hartshorne, and Paul Weiss (Cambridge:

Harvard University Press, 1931-58). *MS* references are to Peirce's manuscripts in the Houghton Library, Harvard University.

[1] Peter Clark, now at Vulcan's Project Halo, explains: [T]here are a whole variety of methods for weighing up evidence, many of which essentially come down to weighted voting. Basically the evidence is a set of features about the hypothesis answer – the programmer has to decide what those features are, and the typical approach is to throw in as many different features as possible. Then the challenge is to determine the right weights on those features so you give more weight to more informative features. If the programmer included an irrelevant feature, it will end up with zero weight so there's no penalty for that. To find the weights, people use machine learning: there are algorithmic solutions for finding the "optimal" weights, i.e., the weight values that produce the highest score on the annotated training data. Then you're done! [personal email 2/10/11]

[2] Burch explains: "Peirce came ever more clearly to see that there are three distinct and mutually incommensurable measures of imperfection of certitude. Only one was probability. The other two he called "verisimilitude" (or "likelihood") and "plausibility". Each of the three measures was associated with one of his types of argument. Probability he associated with deduction. Verisimilitude he associated with induction. And plausibility he associated with abduction. [29]

References

1. GalaxyZoo, <http://www.galaxyzoo.org/>
2. WordPress. Crowdsourcing Revolution in Egypt, <http://siliconcowboy.wordpress.com/2011/02/02/crowdsourcing-revolution-in-egypt/>
3. Mechanical Turk, <https://www.mturk.com/mturk/welcome>
4. Ojala, M.: Data Conservation Lab News (2009), <http://www.dclab.com/crowdsourcinghistory.asp>
5. Flickr Commons, <http://www.flickr.com/commons>
6. Voorhees, E.: Overview of the TREC 2001 Question Answering Track. National Institute of Standards and Technology (2001), <http://www.ai.mit.edu/people/jimmylin/papers/Voorhees01.pdf>
7. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural language interfaces to databases—an introduction. *Natural Language Engineering* 1(1), 29–81 (1995)
8. Shank, R., Colby, K. (eds.): *Computer Models of Thought and Language*. W.H. Freeman, San Francisco (1973)
9. TrueKnowledge. The Internet Answer Engine, <http://www.trueknowledge.com>
10. Tunstall-Pedoe, W.: Knowledge: Open-Domain Question Answering Using Structured Knowledge and Inference, *AI Magazine*, Fall (2010), http://findarticles.com/p/articles/mi_7446/is_201010/ai_n56441011/
11. Kwok, C., Etzioni, O., Weld, D.: Scaling question answering to the Web. In: *Proceedings of the Tenth International World Wide Web Conference (WWW 2010)* (2001), <http://www.cs.washington.edu/homes/weld/papers/mulder-www10.pdf>
12. Clark, P., Harrison, P.: Large-Scale Extraction and Use of Knowledge from Text. In: *Proceedings of the Fifth International Conference on Knowledge Capture*, pp. 153–160. ACM, New York (September 2009)

13. Ferrucci, D.: Lecture in Yorktown (recording), <http://www-943.ibm.com/innovation/us/watson/watson-for-a-smarter-planet/building-a-jeopardy-champion/how-watson-works.html>
14. Common Component Metadata Analysis Engine, <http://uima.lti.cs.cmu.edu/index.html>
15. Answers.com, <http://www.answers.com/>
16. IWISE. Wisdom on Demand, <http://www.iwise.com/91fzT>
17. Thinkexist.com. "Finding Quotations was never this Easy!", <http://www.quotationspage.com/>, <http://www.quotationspage.com/>
18. Framework Institute, <http://www.frameworksinstitute.org/mission.html>
19. BBC World News; and see The PIPA Knowledge Networks Poll BBC World News. Iraqi defector 'Curveball' Janabi denies WMD claims, (February 16, 2011), <http://personaldemocracy.com/pdfleaks>
20. Fox News Survey, <http://www.foxnews.com/politics/2010/08/19/nearly-americans-thinks-obama-muslim-survey-shows/>
21. PolitiFact, <http://politifact.com/truth-o-meter/article/2010/dec/16/lie-year-government-takeover-health-care/>
22. Tomasky, M.: Guardian, <http://www.guardian.co.uk/commentisfree/michaeltomasky/2011/feb/01/healthcare-vinson-decision>
23. Huffington, A.: Report on "A Symposium on WikiLeaks and Internet Freedom". Personal Democracy Forum (12/11/10), http://www.huffingtonpost.com/2010/12/03/wikileaks-online-presence_n_791699.html
24. UTVNews, <http://www.u.tv/News/>
25. Lippmann, W.: Goodquotes.com, <http://www.goodquotes.com/quote/walter-lippmann/when-distant-and-unfamiliar-and-comple>
26. Sagan, C.:
<http://www.positiveatheism.org/writ/saganbd.htm#BALONEY>,
<http://www.positiveatheism.org/writ/saganbd.htm#BALONEY>
27. Quora, <http://www.quora.com>
28. TruthorFiction,
<http://www.truthorfiction.com>,
<http://www.truthorfiction.com/rumors/o/obama-lies.htm>
29. Burch, R.: In: Zalta, E. (eds.): Stanford Encyclopedia of Philosophy, <http://plato.stanford.edu/>
30. Wible, J.: Science Bought and Sold: Essays in the Economics of Science. University of Chicago Press, Chicago (2002)
31. Peirce, C. Studies in Logic. In: Eschbach, A. (ed.) Foundations of Semiotics Series, John Benjamins B.V (1983), pp. 126-181. Originally published as A Theory of Probable Inference, In: The Johns Hopkins Studies in Logic (ed.) Peirce, C. Little Brown and Co (1883); intended as Essay XIV of the Search for a Method (1893)
32. Keeler, M., Pfeiffer, H.: Building a Pragmatic Methodology for KR Tool Research and Development. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 314–330. Springer, Heidelberg (2006)
33. Keeler, M.: Revelator Game of Inquiry: A Peircean Challenge for Conceptual Structures in Application and Evolution. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 443–459. Springer, Heidelberg (2007)
34. Keeler, M., Majumdar, A.: Revelator's Complex Adaptive Reasoning Methodology for Resource Infrastructure Evolution. In: Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 88–103. Springer, Heidelberg (2008)
35. Skagestad, P.: The Road of Inquiry. Columbia University Press (1981)
36. Hilpinen, R.: On C.S. Peirce's Theory of the Proposition: Peirce as a Precursor of Game Theoretical Semantics. In: Freeman, E. (ed.) The Relevance of Peirce, pp. 264–270 (1983a)

37. McGonigal, J.: *Reality is Broken: Why Games Make us Better and How They Can Change the World*. Penguin Press (2011)
38. Bailey, K.: Microsoft, Government Working Together on Deficit Reduction Game, <http://www.escapistmagazine.com/forums/read/7.188356-Microsoft-to-Make-Budget-Balancing-Game>
39. Taleb, N.: *The Black Swan: the Impact of the Highly Improbable*. Random House (2010)
40. Serious Games Initiative, <http://www.seriousgames.org/>
41. Keeler, M.: Learning to Map the Virtual Evolution of Knowledge. In: Croitoru, M., Ferre, S., Lukose, D. (eds.) *Proceedings of the 18th International Conference on Conceptual Structures* (2010)
42. Waters, R.: Information Technology: Mind Games. *Financial Times* (February 16, 2011)
43. Bakalis, A.: AOL, Patch's Parent Company, Buys Huffington Post, <http://hollywood.patch.com/articles/aol-patches-parent-company-buys-huffington-post>
44. Anderson, C.: How web video powers global innovation, http://www.ted.com/talks/chris_anderson_how_web_video_powers_global_innovation.html
45. JoVE. *Journal of Visualized Experiments*, <http://www.jove.com/>
46. Woods, J., Walton, D.: *Argument: The Logic of Fallacies*. McGraw-Hill, New York (1982)
47. Burks, A.: Logic, Learning, and Creativity in Evolution. In: Houser, N., Roberts, D., Van Evra, J. (eds.) *Studies in the Logic of Charles Sanders Peirce*, pp. 497–534. Indiana U. Press (1997)
48. Sowa, J.: Matching Logic Structure to Linguistic Structure. In: Houser, N., Roberts, D., Van Evra, J. (eds.) *Studies in the Logic of Charles Sanders Peirce*, pp. 418–444. Indiana U. Press (1997)

Evaluating the Transaction Graph through a Financial Trading Case Study

Ivan Lauanders

BT Innovate & Design, PO Box 200, London, United Kingdom
ivan.lauanders@bt.com

Abstract. The Transaction Graph is argued as leading to a better understanding of the concepts and relations in enterprise transactions and their semantics with business rules. This design process begins with an analysis of case study narrative through the Transactional Use-Case, using a Financial Trading case study in this evaluation. The lexicon or dictionary of words and their underlying concepts are explored in order to model transactions. Peirce logic (i.e. Peirce's visual Extential Graphs as extended by Sowa and Heaton) is then applied to the business rules to visualise the inferences in Conceptual Graphs and therefore any further possible Transaction Graph refinements. Model automation provides validation checking removing errors in semantics and syntactics. The paper demonstrates the reuse and refinement of a generic Transaction Model highlighting inference and logical operations with business rules. Additionally the paper demonstrates the reuse of the automated Transaction Model's ontology coupled with its conceptual catalogue.

1 Introduction

This paper evaluates the Transaction Graph based on Sowa's theory of Conceptual Graphs (CG) and Hill's Transaction Agent Modelling (TrAM) theory on multi-agent systems [6, 7, 11, 13] through a Financial Trading case study. The paper starts with a case study narrative, using a Financial Trading (FT) case study adapted from a version by Said Tabet and Gerd Wagner, and reproduced as appropriate in table 1. TechRules Advisors (TRA Inc.) is a fictitious asset management firm. The firm buys and sells numbers of shares of securities and manages its clients' assets. Portfolio managers create and manage accounts. The FT case study described in table 1 has been selected as it provides a benchmark example which has been previously worked through UML by Said Tabet and Gerd Wagner. The Transaction Graph provides an automated framework to formalise narrative within business transactions in order to model and explore through inference and logical operations with business rules [10].

The design process begins with a Transactional Use-Case (TUC) analysis reusing the generic Transaction Model (TM) providing a focus on the Resources, Events and Agents (REA) within the enterprise transactions [4, 6, 10]. The generic TM ontology coupled with its conceptual catalogue are reusable design artefacts which can be applied to different domains assisting the transaction design in a new domain though the sharing and reuse of a specification of concepts and relations that have already

been tested in a previous domain; thus ontology. Reuse of the automated ontology and conceptual catalogue allow a Transaction Graph to be produced and tested more rapidly. This ontology reuse works with three principal objectives [3]:

1. It must represent a conceptualisation that can be shared and reused;
2. The ontology must represent all of the applications within a domain and not be specific to one type;
3. It must contain all of the required information to permit knowledge to be explicitly stated (providing a declaration of terms, achieved through the conceptual catalogue), together with rules and constraints to facilitate the inference of new knowledge (canonical use and conceptual definitions).

2 The Financial Trading Case Study

Businesses need words and sentences to express transactions and rules as they continue to sign written agreements, follow regulations, verbalise business policies, and capture service knowledge [13]. Narrative text capturing and describing enterprise transactions is often the first piece of information a designer will be able to work with. Table 1 provides the narrative text for the Financial Trading case study to be analysed. This case study is concise in detail and not over complicated in that it does not provide supporting stakeholder detail allowing a designer to work with a small number of significant business transaction concepts.

Table 1. Financial Trading case study

Case Study "Financial Trading" © Said Tabet and Gerd Wagner

TechRules Advisors (TRA Inc.) is a fictitious asset management firm. The **firm buys and sells numbers of shares of securities and manages its clients' assets. Portfolio managers create and manage accounts.**

A portfolio is owned by a legal entity and is managed by a portfolio manager who works for an investment firm. A portfolio is described by a creation date and a value. It consists of a number of positions. Each position holds an asset and is described by a quantity and an acquisition date. The value of a portfolio is the total value of all the securities held in the portfolio. There are three different categories of assets:

- real estate, cash, and securities.

Real estate and cash are described by a name. Securities are described by:

- a security ID, a name, a price.

There are three categories of securities: options, bonds, stocks.

They are issued by a legal entity that is called *issuer*, which can be:

- a company, a municipality, an agency, a government.

There are many reasons that motivate issuers to issue securities (repay debts, raise capital, etc). Issuers (and the securities they have issued) can be affected positively or negatively by market events such as **upgrades or downgrades by credit rating agencies**. Certain issuers are classified as restricted by portfolio owners and investment firms. Orders (for buying or selling assets) are placed in the interest of a portfolio. An order is **placed by a trader or by a portfolio manager**.

After an initial review of the case study a designer (e.g. an enterprise architect) would start by identifying the main stakeholders (agents central to transactions) and examine the key transactional facts (shown in bold) within the FT case study narrative text. Asking the analysis questions “What are the economic resources?”, “How do the economic events operate with those resources?”, “Who are the agents?”, and “Why are they transacting, what are the business goals?” These questions are central to bringing focus to the initial Transactional Use-Case (TUC). In subsequent design iteration steps, the designer needs to ask deeper questions such as what are the relations between those business concepts described in the narrative text? What are the definitions of those concepts used and what are their typical uses (canonical use)? What do the words mean and what form to apply to those words and concepts in the context of this case study domain? To address these questions Automated TrAM identifies three distinct parts to the early requirements capture stage; namely Model Fundamentals, Model Visualisation and Model Automation. Each part can require several design iterations to make a formal judgement and produce a Transaction Graph represented in CG, therefore capturing the intentions, communications, conversations, and negotiations involved in the business transactions.

3 Model Fundamentals

The process begins with Transactional Use-Case (TUC) analysis providing a focus on those use-cases relevant to the main transactions. TUC analysis aims to capture those high level interactions between economic resources, economic events, agents and business goals in an informal diagram form as shown in Fig. 1. The checks initiating TUC include identifying the stakeholders, writing the initial use-cases, defining actor goals and setting use-case priority. The TUC diagram focuses on the key transactions and their associated actors (agents). Good use-case and TUC practice would assume capturing the high level components (typically no more than six components in a TUC diagram) of the business transaction. Seeking and identifying key facts needed to capture the semantics with the transaction. A high level of abstraction for both TUC and CG is important at this stage as refinement and specialisation will occur when applying business rules.

3.1 Capture Transactional Use-Case

The first part of the analysis involves capturing the high level transactional facts from the case study narrative, those transactional facts that are key to the enterprise system. Central to that capture is the ‘What’ (Economic Resources?), ‘How’ (Economic Events?), ‘Who’ (agents?), and ‘Why’ (business goals). Identifying the inside agents and outside agents as per Geerts and McCarthy [4]. Starting this process as follows:

The ‘What’ - TRA inc. transacts with the economic resources; real estate, cash, securities (options, bonds, and stocks).

The ‘How’ - The firm buys and sells numbers of shares of securities and manages its clients' assets. Portfolio managers create and manage accounts.

The ‘Who’ - Clients, issuer, trader, credit rating agency.

Identifying what their tasks provide as produced in Table 2:

Table 2. Agent types and allocated tasks

Agent Type	Allocated Tasks
Clients	buy and sell assets
Issuer	issue securities
Trader	place buy and sell orders
Credit Rating Agency	upgrade and down grade securities

The ‘Why’ - The business goals, TRA inc. provide a financial service and therefore derive profit on the size and number of transactions. Fast and efficient transactions with no transactional error at a competitive transactional cost equates to good quality service. Interestingly these business goals are not outlined in the case study narrative and so have been deduced from the narrative.

After capturing the “who” and the “what” parts from the case study narrative illustrate the “how” part in the form of a TUC diagram as shown in Fig. 1. The firm buys and sells numbers of shares of securities and manages its clients' assets. Portfolio managers create and manage accounts. From this description of logic portfolio managers would seem to be central to orchestrating FT transaction in this case study and so a good starting point for the centre of the TUC diagram. This is not a specific rule but process logic is typically described through verbs referring to actions and states. In the case of the FT case study transitive verbs (verbs that take a direct object) for example the firm buys and sells numbers of shares of securities, portfolio managers create and manage accounts. Capturing the logic described in the case study narrative into the transactional use-case diagrams provides a visual outline as follows:

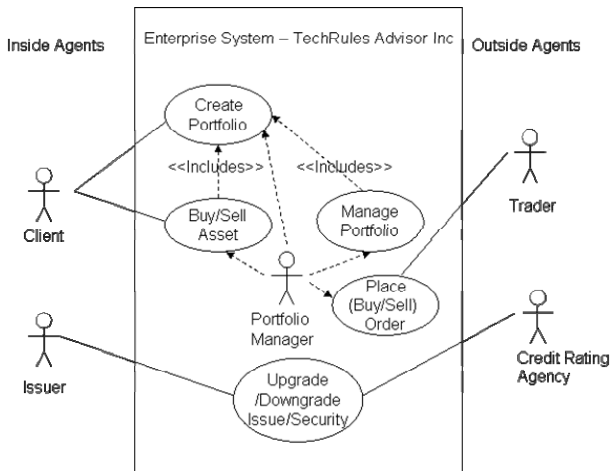


Fig. 1. Financial Trading Transaction Use-Case

It should be noted that this is not a complete TUC but a first iteration. Fig. 1 illustrates and starts by capturing six key components within the enterprise system boundary “TechRules Advisers Inc”. The box around those six components denotes the system boundary. Outside of the system boundary agents show interaction with those components. The TUC shown in Fig. 1 is less formal than CG for the analysis, less formal in the sense that it contains ambiguity in terms of the meaning of the words denoting the components. The words may have different meanings to different enterprise designers, for example what is meant by the word ‘portfolio’ in the context of the FT case study? What are the limitations of transaction on a portfolio? A designer can’t be expected to establish those facts from an initial TUC. The case study needs a supporting set of business rules to refine those initial high level concepts, however before refining the focus needs to stay on the high level concepts and capture and visualise how they relate in a given context.

3.2 Transform Transactional Use-Case into Conceptual Graphs

The second part of the analysis involves transforming the less formal TUC into CG and then into the form of Hill's generic Transaction Model (TM) [6], starting with the central concept and working outwards. For example, portfolio manager as a concept relates to a portfolio as a manager. Relating concepts to each other starts to provide a formal model between the words provided in the narrative text. For example the TUC in Fig. 1 uses the concept “issue security” which could mean many things in different contexts such as “security clearance” or “security notice”. Even in the correct context the concept “security” could have several meanings and have deeper consequences. For example a debt “security” which could be bought and sold and carries the same type of interest applied to balance or draft.

It is not until the word is related within context that the semantics within the transaction design become specified, in the case of the FT case study it is more obvious that we mean “security” as in a financial product to transact upon. Let us consider how we surface meaning for the concept “Portfolio Manager” in the FT case studying in terms of the words (lexicons). A lexicon or dictionary relates the word of a language to their grammatical category and their underlying concepts [13]:405-425. For example:

portfolio. mass noun; Portfolio

manager. count noun; Manager

portfolio manager. count noun; Portfolio_Manager

Graphs for the first part of this TUC are as follows:

manager. links [Portfolio_Manager] to Portfolio. Example: *The manager of a portfolio is a portfolio manager.*

[Portfolio_Manager]<-manager-[Portfolio].

creator. links [Portfolio_Manager] to Portfolio. Example: *The creator of a portfolio is a portfolio manager.*

[Portfolio_Manager]<-creator-[Portfolio]

This graph is not a definition but it provides a canonical basis in the context of the FT case study. Sowa outlines that an explicit type definition is not always possible [13]:405-425. This is valuable in that it formally specifies the relationships between

concepts in a given domain. Model automation discussed in section 5 ensures the designer provides either a definition and or canonical use of those concepts and relations if possible removing the syntactic errors [10]. The next step is to transfer each of those concepts and their relationships identified in the TUC into CG formally representing the logic captured as shown in Fig. 2:

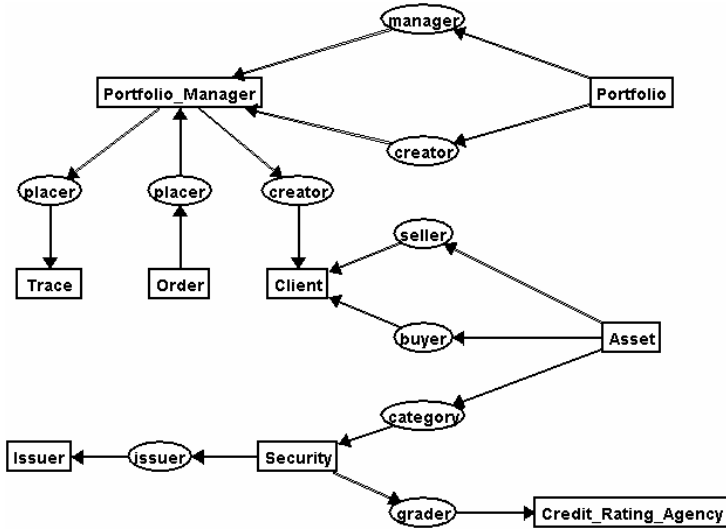


Fig. 2. Transforming the TUC into CG

Fig. 2 shows those concepts transferred and maximally joined using the logic extracted through TUC analysis in Fig. 1. Maximally joined graphs are joined on the most common or maximally extended, projection [13]. The corresponding concepts and relations in the overlapping regions of each graph are specialised so that the result contains the most specific object for each node involved. For example in the maximal join of Portfolio Manager and Portfolio graphs shown in Fig. 2 we see that Portfolio Manager has two relationships to a Portfolio, that of manager and creator of a portfolio.

A Type Hierarchy as illustrated in Fig. 3 provides the design artefact that enables the inheritance of properties for concepts and relations to be examined. A type hierarchy is used to support the inheritance of properties from supertypes to subtypes of concepts within the TM. Subtypes inherit the properties of supertypes for example in Fig. 3 the subtypes “Asset” inherits the properties of the supertype “Economic_Resource”. Fig. 3 also shows relations supertypes and their subtypes, providing clarification for example on the supertype “Economic_Event” in that “Sale” is a subtype of “Economic_Event”. The subtype relation is used to relate concepts by a partial order for example, $A \leq B$ means that A is a subtype of B so in the context of “Sale” \leq “Economic_Event”, “Pay” \leq “Economic_Event”, “Dispose_Recommendation” \leq “Economic_Event”. Automation of a type hierarchy within an Amine ontology provides refinement in that the automation demands resolution to subtypes used in an ontology in that subtypes must be declared and subsequently defined [10].

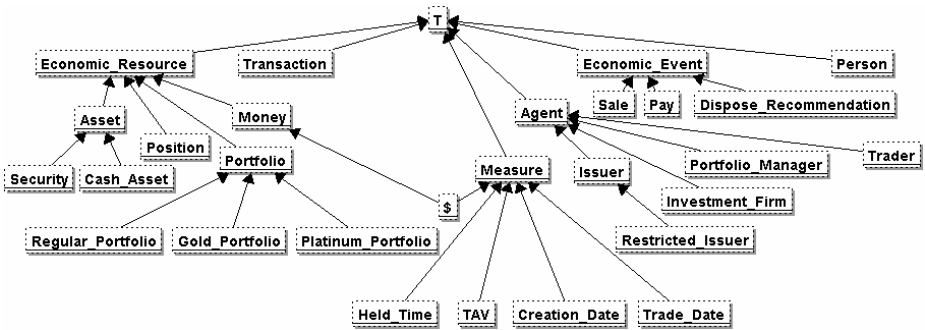


Fig. 3. Financial Trading Type Hierarchy through Automation

The analysis of the type hierarchy is then confirmed through the definition and canonical use of the super and subtypes adding to the conceptual catalogue.

3.3 Integrate Conceptual Graphs with the Generic Transaction Model

TRA Inc. sell (and buy) assets on behalf of their clients. The assets come from an issuer. Examining one of the transactions in detail shows the balance within the transaction to sell assets. Our case study and therefore transaction design is interested in one specific investment company TRA Inc. A subsequent step to include the fact of being specific in terms of TRA Inc is to refine the TM though specialising the Investment Company, this produces the TM as illustrated in Fig. 6.

An important aspect of developing the generic TM into the Transaction Graph for TRA Inc. involves iteration reviewing design output at each stage looking for inconsistencies and then resolving those inconsistencies.

4 Model Visualisation

Model visualisation uses Peirce's visual Extential Graphs as extended by Sowa and Heaton and applied by Polovina and Hill [5, 11, 13]. This Peirce logic visualises the inferences in CG and therefore any further possible refinements within the Transaction Model. Hill outlines the use of Peirce logic as a manual technique in TrAM [6]. This research works with the original technique which defines that once a TM is complete, queries are then created and used to test the TM applying the case study business rules and Peirce Logic. The intention of model visualisation is to show the contexts of knowledge elements, inference is performed through negation with an attempt to reduce those contexts [11]. For example consider *Rule 1*:

Rule 1: Securities issued by a 'restricted' issuer must NOT be bought.

Consider if to apply an IF-THEN or OR rule and apply the logic to the rule rewriting as follows:

IF a security is issued by a 'restricted' issuer **THEN** it must **NOT** be bought

In Peirce logic 'if-then' can be rewritten as **not** (Graph 1 **and not** Graph2).

Transferring *Rule 1* graphically in Peirce showing the referent link between **not** (Graph 1 **and not** (not Graph 2)). Is as follows:

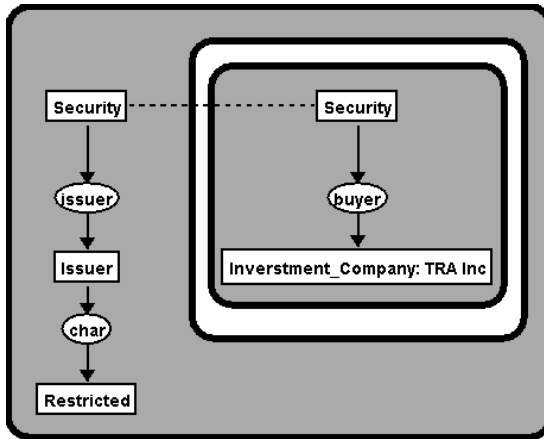


Fig. 4. Example Rule1 inference

not (Graph 1 **and not** (**not** Graph 2)). maps to the Peirce logic form **not** (Graph 1 **and** Graph 2).

Applying Peirce logic we know that:

IF A THEN B is true

IF NOT B THEN NOT A is also true

Hence:

IF a security is issued by a 'restricted' issuer THEN it must NOT be bought

Implies:

IF a security is bought from an issuer THEN the issuer is NOT restricted.

While the example may seem trivial on the surface the semantics and consequences of trading a “security” can be far from trivial and can benefit from deeper understanding in a business domain. For example in rule 1 establishing regulated activities behind trading securities and their levels of restriction would inform the canonical use of the concept “Restricted”.

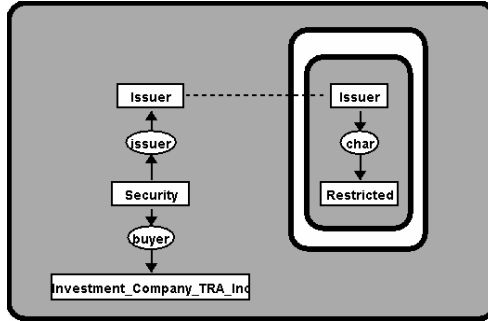
Fig. 5 (c) shows *Rule 1* after applying inference and logical operations. Then combining the visualisation illustrated in Fig. 4 and Fig. 5 into the TM provides the following:

The model visualisation step works though each of the business rules further specialising the TM to finally produce a Transaction Graph as illustrated in Fig. 7.

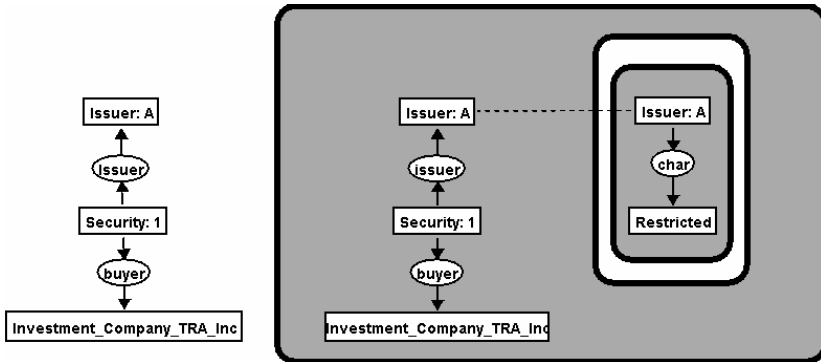
4.1 Rule Refinement

Model visualisation aims to refine the model through applying business rules to the TM. Model visualisation uses Peirce Logic to perform inference to highlight further refinements, however at this point in time there are no known software tools capable of performing Peirce logic. It is however possible to automate and test parts of the inference process such as testing a projecting between *inner* and *dominate* graphs.

(a) Graphs showing referent between **not** (Graph 1 **and not** (**not** Graph 2)). Map as follows:



(b) Specialisation of projecting graph and referent passing through link:



(c) After deiteration of the projecting graph and double negation Rule 1 becomes:

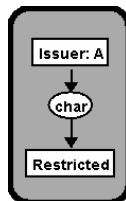


Fig. 5. Example rule 1 in Peirce logic

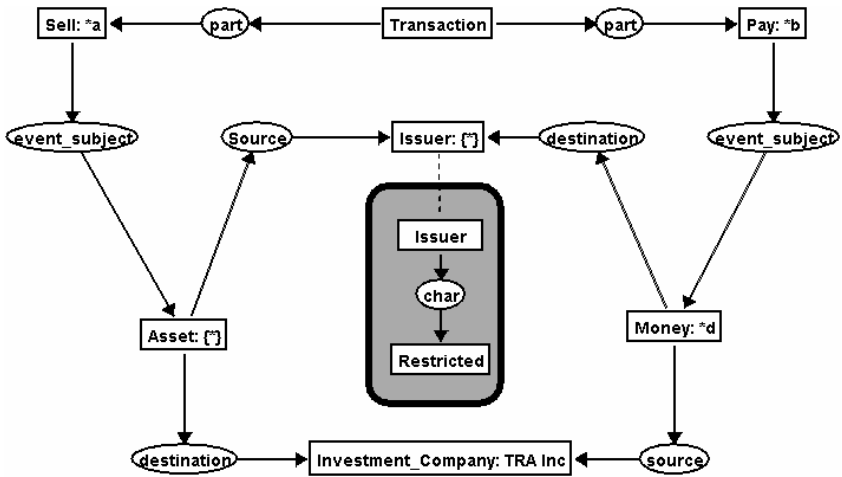


Fig. 6. Rule 1 combined into the FT TM

It is also possible to be able to capture business rule in CG in both CharGer and Amine but again there are differences in the translations between CharGer and Amine as demonstrated in this section [2, 9].

The following list provides a complete set of the business rule set for the FT case study:

Table 3. Business rule set for Financial Trading case study

1. Securities issued by a 'restricted' issuer must NOT be bought.
2. An asset must NOT be sold if it has been in the portfolio for less than 30 days.
3. The total asset value (TAV) is the sum of the market value of all positions.
4. The value of cash assets must be less than or equal to 10% of the total asset value.
5. A portfolio is rated *platinum*, if TAV is greater than 1 Mio \$.
6. It is rated *gold*, if TAV is less than 1 Mio \$ and greater than 100.000 \$.
7. It is rated *regular*, if TAV is less than 100.000\$.
8. If there is a downgrade for a security held in a portfolio, the portfolio owner must be sent a "dispose recommendation". This advises the owner that they should sell the security.
9. An order placed in the interest of a portfolio must not refer to more than one asset held in a position of that portfolio.
10. The trade date of an order placed in the interest of a portfolio must be after the date that portfolio was created.
11. An order must not be placed both by the trader and by the portfolio manager.

Applying rule refinement to the financial trading case study exemplar results in the Transaction Graph illustrated in Fig. 7. Financial trading rules are identified in the numbered boxes overlaid on the Transaction Graph as shown below:

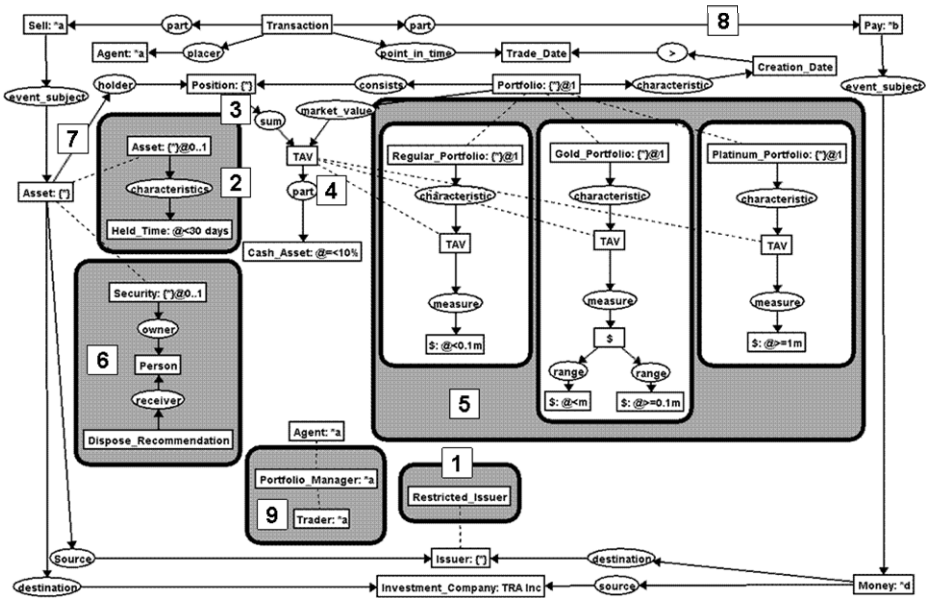


Fig. 7. Financial Trading Transaction Graph with Rule Refinement

The translation of business rules into CG using CharGer and Amine illustrates further difficulties when working with different CG tools for rule refinement. For example, *Rule 4* "The value of cash assets must be less than or equal to 10% of total asset value." Shown as follows first in CharGer and then secondly in Amine:

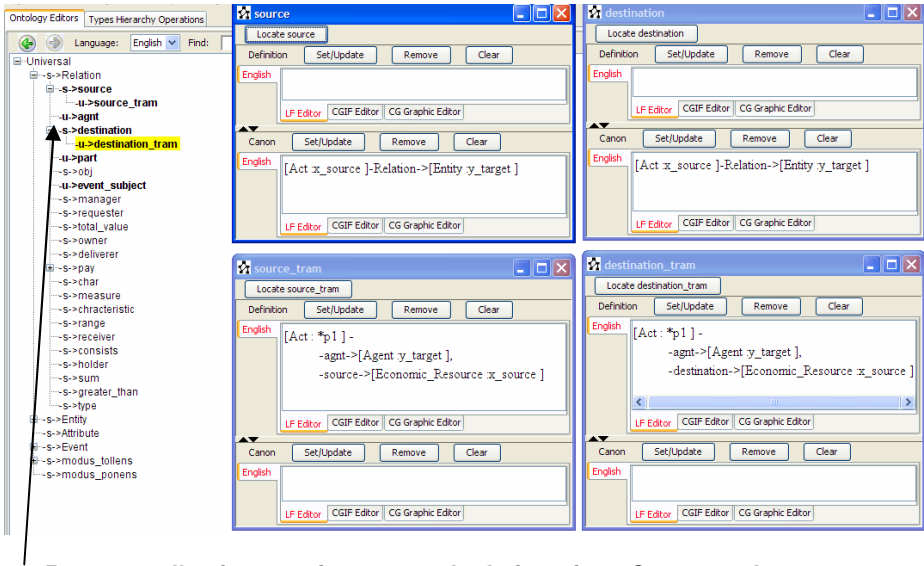
```
[Cash_Asset: @=<10%]<-part-[TAV].
[Cash_Asset]-equallessthan->[TAV]<-percentage[Integer:10]
```

CharGer is accurate to the ISO standard [1] while Amine tackles the practicalities of CG operations. One of the difficulties is that CharGer works with symbols and values in a different way to Amine, for example CharGer can express rules using the symbols @, @<, @>, and @=< where as in Amine the syntax has to be explicit and to define the value or the cardinality. The adoption of standard CGIF by CG tools will enable interoperability between them to be achieved, and in turn lead to the interoperability between CG tools and other tools [8]. CharGer version 3.6 [2], allows for the export of CG in standard CGIF but had no support for reading a CG written in standard CGIF. Amine V7.0 [9] now allows for the import of type hierarchy from CharGer.

5 Model Automation

A model should be able to answer questions in place of the actual system therefore providing design verification. Model automation transfers the paper based analysis into the software based Transaction Graph ontology. Automation provides the

mechanism for exploring and testing the semantics of concepts used with the transaction. The automated ontology does not use Peirce logic as we are at present unable to compute *deiteration* and *double negation*. Automating the ontology does however provide validation checking removing semantic and syntactic errors through CG operations [10]. Automation also provides the basis for practical reuse of the TM, adding definitions and canons to concepts in order to build the TM.



Reuse a collective set of conceptual relations for a Conceptual Catalogue. Starting with Sowa’s ‘Conceptual Catalogue’ [11]:405-425.

Fig. 8. Reuse of Conceptual Catalogue in Amine

Fig. 8 shows the conceptual relations reusing and building upon a conceptual catalogue for the FT Ontology. Source and Destination from Fig. 6 are shown in terms of their definition and canonical use within the TM. Hill and Polovina [7] provide a detailed build up of the generic CGs for the TM in linear form.

In summary model automation [10] comprise for the following:

- *Creating a Conceptual Catalogue:* In order to compute an ontology and to describe the forms to apply to words and concepts used in the FT business transactions it is necessary to further develop a conceptual catalogue. Sowa’s published conceptual catalogue provides a starting set which needs extending and modifying to fulfil business transaction for the FT case study [13].
- *Verify the semantics and logic captured in the TM:* Having built an initial TM ontology, the next step of the automation process is to use Amine’s CGOperational interface to develop the generic TM ontology into a Transaction Graph, testing adding facts through *projection* and *maximal join*. Relatively simple CG operations rapidly become complex and are therefore prone to error if handled manually.

6 Concluding Remarks

Evaluating the Transaction Graph through a ‘Financial Trading’ (FT) case study has improved the understanding of the concepts and relations in FT transactions and their semantics within the context of the case study. The Transaction Graph focuses on early model verification, examining the semantics of concepts and relations in transactions through the Transaction Graph ontology, in order to deepen the understanding of how those transactions are specialised with business rules. Conceptual Graphs are unambiguous, applying CG through the TM provides the means to reduce ambiguity and to clarify semantic of specific terms used through a conceptual catalogue. For example clarifying the tasks to be carried out by a ‘portfolio manager’ informs the definition for a ‘portfolio manager’ within FT transactions.

Whilst the evaluation is specific to FT it demonstrates reuse of an automated generic TM, reusing the TM ontology including the TM parts of a conceptual catalogue produced and tested in a different domain [6, 10]. The example of reuse described in this paper is however limited in that it only uses a small conceptual catalogue including those concepts and relations used in the generic TM. The evidence suggests that to derive significant value from ontology reuse a far greater conceptual catalogue needs to be developed through a larger sample of case studies.

Automation of the TM through Amine [9] makes it possible to more rapidly create ontology to explore business transactions such as FT. The automation facilitates the capture of canonical use and definition in that they need resolving in Amine for those concepts and relations to compute. Building the FT ontology provided validation checking removing semantic and syntactic errors through automated CG operations. The Transaction Graph and resulting artefacts provide a transaction design specification for the eventual enterprise architecture that does not impose a particular implementation but servers to complement architecture framework that lack a semantic requirements gathering stage.

Acknowledgements. This work has been assisted by the generous efforts of Lynne Dawson. Lynne has questioned and examined each step of the logic described in this paper. I would like to thank Gerd Wagner for allowing for the use of an adapted version of the FT case study in this research. The FT case study was first introduced at the REVERSE summer school in Malta in 2005.

References

1. Delugach, H.S.: Information technology – Common Logic (CL) – A Framework for a Family of Logic-Based Languages (2009), http://standards.iso.org/ittf/PubliclyAvailableStandards/c039175_ISO_IEC_24707_2007E.zip. (May 2, 2009)
2. Delugach, H.S.: CharGer: A graphical conceptual graph editor (2002), <http://glotta.ntua.gr/StateoftheArt/CGs/CharGer.pdf> (May 2, 2009)
3. Gruber, T.R.: A Translation Approach to Portable Ontology Specification. Knowledge Acquisition: special issue - Current Issues in Knowledge Modelling 5(2), 199–220 (1993)

4. Geerts, G.L., McCarthy, W.E.: Database Accounting Systems. In: Williams, B.C., Spaul, B.J. (eds.) *Information Technology Perspectives in Accounting: an integrated approach*, pp. 159–183. Chapman & Hall, London (1991)
5. Heaton, J.: *Goal Driven Theorem Proving using Conceptual Graphs and Pierce Logic*. PhD edn, Loughborough University (1994)
6. Hill, R., Polovina, S., Shadija, D.: Transaction Agent Modelling: From Experts to Concepts to Multi-Agent Systems. In: Scharfe, H., Hitzler, P., Ohrstrom, P. (eds.) *ICCS 2006. LNCS (LNAI)*, vol. 4068, pp. 247–259. Springer, Heidelberg (2006)
7. Hill, R., Polovina, S.: An Automated Conceptual Catalogue for the Enterprise. In: Eklund, P., Haemmerlé, O. (eds.) *Supplementary Proceedings of 16th International Conference on Conceptual Structures (ICCS 09): Conceptual Structures: Knowledge Visualization and Reasoning*, Toulouse, France. *CEUR-WS*, vol. 354, pp. 99–106 (July 2008)
8. Kabbaj, A., Launders, I., Polovina, S.: Interoperability through standard CGIF notation: The case of Amine Platform. In: *Supplementary Proceedings of the 17th International Conference on Conceptual Structures, ICCS 2009* (2009)
9. Kabbaj, A., About Amine Platform (2010), <http://amine-platform.sourceforge.net/about.htm> (March 2010)
10. Launders, I., Polovina, S., Hill, R.: The transaction pattern through automating TrAM. In: *17th International Conference on Conceptual Structures, Aachen, CEUR-WS* (2009)
11. Polovina, S., Hill, R.: A Transactions Framework for Effective Enterprise Knowledge Management. In: Akhgar, B. (ed.) *Proceedings of the 15th International Workshops on Conceptual Structures*, pp. 221–225. Springer, London (2007)
12. Ross, R.G.: More on the If-Then Format for Expressing Business Rules (2011), <http://www.brcommunity.com/b588.php> (April 21, 2011)
13. Sowa, J.F.: *Conceptual structures: Information Processing in Mind and Machine*. Addison-Wesley, London (1984)

Integration of the Controlled Language ACE to the Amine Platform

Mohammed Nasri¹, Adil Kabbaj², and Karim Bouzoubaa³

¹ EMI, Rabat, Morocco,

mohammed.nasri@gmail.com

² INSEA, Rabat, Morocco, B.P. 6217

akabbaj@insea.ac.ma

³ EMI, Rabat, Morocco

Karim.bouzoubaa@emi.ac.ma

Abstract. This paper presents the integration of the controlled language ACE (Attempto Controlled English) to Amine platform. Since the parser engine of ACE (ACE Parser Engine or APE) generates a DRS structure (Discourse Representation Structure), we have developed a mapping from DRS to CG (DRS2CG) which produces a CG equivalent to the DRS produced by APE. Through this mapping and this integration of ACE into Amine, Amine users can use controlled language to express their knowledge or specifications, instead of having to express them in CG directly.

Keywords: Natural language processing controlled language, ACE, DRS, CG, and DRS to CG mapping.

1 Introduction

This paper presents the integration of the controlled language ACE [9, 11] into the Amine platform [1, 2]¹. Amine is an open source software dedicated to the development of intelligent systems and agents. Amine provides Conceptual Graphs (CG) as its main knowledge representation formalism [7, 8].

Instead of formulating knowledge directly in CG, it would be easier and more convenient to formulate it in natural language. For this purpose, Amine should be extended by adding a Natural Language Processing component.

Given the numerous complexities and ambiguities of natural language, many researchers have chosen to focus on one subset of natural language², conventionally called "controlled language". Among these researchers is the Attempto group³.

Controlled Natural Languages are subsets of natural languages whose grammars and dictionaries have been restricted in order to reduce or eliminate both ambiguity and complexity.

¹ <http://sourceforge.net/projects/amine-platform>

² <http://sites.google.com/site/controllednaturallanguage/>

³ <http://attempto.ifi.uzh.ch>

In our study, we opt for the controlled language ACE (Attempto Controlled English). ACE is a rich subset of Standard English designed to serve as knowledge representation language. ACE allows users to express texts precisely, and in the terms of their respective application domain [12, 13].

The Attempto group worked on various research and projects based on ACE such as the syntactic-semantic parser APE (ACE Parser Engine) [9] which extracts the ACE sentence semantics and represents it in the DRS formalism (Discourse Resource Structure) [10, 4].

In order to integrate ACE/APE with Amine, we developed a mapping between DRS and CG called DRS2CG. A sentence in ACE would be handled by APE generating a DRS structure from which DRS2CG generates a CG (Figure 1).



Fig. 1. Natural Language Processing Component in Amine Platform

This paper is organized as follows: Section 2 introduces briefly the controlled language ACE and DRS formalism, section 3 presents DRS2CG mapping for simple and composite sentences with some examples, the fourth one describes some applications of this mapping. The article ends with a conclusion and some future works.

2 ACE and DRS

2.1 ACE

ACE (Attempto Controlled English) is a language specifically designed to write specifications. It is a controlled natural language, i.e. a subset of English with a domain specific vocabulary and a restricted grammar in the form of a small set of construction and interpretation principles. ACE allows representing specifications in simple or compound sentences. It also helps to formulate questions or queries.

Any ACE instruction consists of sentences, a sentence consists of words or other sentences.

Each sentence is characterized by a function that defines its role in a given context (like subject, complement, etc.) and a form defined as: negative sentence, positive sentence, verbal sentence, etc.

A set of optional elements called "modifiers" can be added to a sentence so as to give additional information; ACE accepts noun modifiers (adjectives, relative sentences, propositional phrases, possessive nouns and appositions) and verb modifiers (adverbs and prepositional phrases).

In ACE, users can build composite sentences from simple sentences, or composite phrases from simpler phrases, with the help of so-called constructors. ACE provides four different types of constructors: coordinators, subordinators, quantifiers and negators.

ACE supports questions and command sentences too and is able to treat modality sentences, like: admissibility, possibility, recommendation, provability and necessity.

2.2 DRS

The ACE parser translates an ACE text unambiguously into a DRS representation [9]. It consists of a set of elements: “object”, object “properties”, possession “relations”, “predicate”, “modifier_adv” / “modifier_pp” to give more information to the predicate, “has_part” to specify that an object belongs to a group and “query”.

Due to space limitation, we give briefly the structure of each element of DRS in the following parts, for more details, please refer to DRS technical report [10]:

- Object: Object (Ref, Noun, Quant, Unit, Op, Count)
- Property
 - Property (Ref1, Adjective, Degree)
 - Property (Ref1, Adjective, Degree, Ref2)
 - Property (Ref1, Adjective, Ref2, Degree, CompTarget, Ref3)
- Relation: Relation (Ref1, of, Ref2)
- Predicate
 - Predicate (Ref, Verb, SubjRef)
 - Predicate (Ref, Verb, SubjRef, ObjRef) for a transitive verb
 - Predicate (Ref, Verb, SubjRef, ObjRef, IndObjRef) for a ditransitive verb
- Modifier
 - modifier_adv (VerbRef, Adverb, Degree) for verb modifier
 - modifier_pp (VerbRef, preposition, ObjRef) for noun modifier
- Has_part: has_part (GroupRef, MemberRef)
- Request: Query (Ref, QuestionWord)

Here are some examples of structures with the DRS corresponding ACE sentences:

- *Example 1:* « a card is green. »:
 - object (A, card, countable, na, eq, 1), property (B, green, pos), predicate (C, be, A, B).
- *Example 2:* « A customer has at least 2 cards that are valid. »:
 - object (A, customer, countable, na, eq, 1), object (B, card, countable, na, geq, 2), property (C, valid, pos), predicate (D, be, B, C), predicate (E, have, A, B).
- *Example 3:* « A customer enters a card quickly and manually in a bank in the morning. »:
 - object (A, customer, countable, na, eq, 1), object (B, bank, countable, na, eq, 1), object (C, card, countable, na, eq, 1), predicate (D, enter, A, C), modifier_pp (D, in, E), modifier_pp (D, in, B), modifier_adv (D, manually, pos), modifier_adv (D, quickly, pos), object (E, morning, countable, na, eq, 1).

3 DRS2CG Mapping

3.1 Simple Sentences

DRS2CG mapping uses the DRS structure so as to extract all the semantic elements and generate the corresponding conceptual graph. As mentioned by John Sowa, "Kamp's DRS notation is isomorphic to Peirce's existential graphs (EG), and conceptual graphs are a typed version of EGs" [6].

DRS mapping to CG is done as follows:

- Mapping of Object construct :

DRS: Object(Ref,Noun,Quant,Unit,Op,Count)

CG:

[Noun] -quant->[Quant]

-count->[number]<-Op-[integer : "Count"]

In a single object case (Quant=countable, op=eq et Count =1) we generate the concept [noun].

- Mapping of Property construct :

DRS: property(Ref,Adjective,Degree)

CG:

[Object(referenced by Ref)]-property->[Adjective]

DRS: property(Ref1,Adjective,Degree,Ref2)

CG:

[Adjective] -

<-property - [Object(referenced by Ref1)],

-target->[Object(referenced by Ref2)],

-opComp->[Degree],

DRS: property(Ref1,Adjective,Ref2,Degree,CompTarget,Ref3)

CG:

If CompTarget is subj, this corresponds to:

[Adjective] -

<-property - [Object(referenced by Ref1)],

-target->[Object(referenced by Ref2)],

-opComp->[Degree],

<- property-[Object(referenced by Ref3)]

Otherwise

[Adjective] -

<-property - [Object(referenced by Ref1)],

-target1->[Object(referenced by Ref2)],

-opComp->[Degree],

-target2->[Object(referenced by Ref3)]

- Mapping of Relation construct :

DRS: relation(Ref1,of,Ref2)

CG:

[Object(referenced by Ref1)] - poss ->[Object(referenced by Ref2)] if Res2 refers to an object, and

[Object(referenced by Ref1)] - attr -> [Object(referenced by Ref2)] if ref2 refers to an attribute.

- Mapping of Predicate construct :

DRS: predicate(Ref,Verb,SubjRef)

CG:

[Object(referenced by SubjRef)]<-Agent-[Verb]

DRS: predicate(Ref,Verb,SubjRef, ObjRef)

CG:

[Verb] –

-Obj->[Object(referenced by ObjRef)]

-Agent->[Object(referenced by SubjRef)]

DRS: predicate(Ref,Verb,SubjRef, ObjRef, IndObjRef)

CG:

[Verb]-

-Agent ->[Object(referenced by SubjRef)],

-Obj->[Object(referenced by ObjRef)],

-Dest ->[Object(referenced by IndObjRef)]

- Mapping of Modifier construct:

DRS: modifier_adv(VerbRef,Adverb,Degree)

CG:

[Verb(referenced by VerbRef)]->manner->[Adverb]

DRS: modifier_pp(VerbRef,preposition,ObjRef)

CG:

[Verb(referenced by Ref)]-loc/time ->[preposition]

- Mapping of Has_part construct:

DRS: has_part(GroupRef,MemberRef)

CG: [Group] <- partOf - [Member]

- Mapping of Query construct:

DRS: query(Ref,QuestionWord)

CG: [Location : x], [Time : x], [Manner : x] or [Object : x]

The mapping of DRS structure to CG is based on these correspondences. Here is an example of the mapping of a simple sentence.

ACE: John gives Mary at least 2 cards in the bank carefully.

CG:

[give : *p3] -

-agent->[Person :John],

-obj->[card : *p1] -

-quant->[countable],

-count->[number : *p2]<-greaterThanORequalTo-[number : 2];

-dest->[Person :Mary],

-loc->[bank],

-mann->[carefully]

The next section presents the DRS2CG mapping for the composite sentences.

3.2 Composites Sentences

DRS2CG mapping also handles ACE composite sentences. Recall that ACE builds composite sentences using four constructors: coordination, subordination, quantifiers and negators.

In addition to simple and composite sentences, DRS2CG handles questions, commands, possibility, necessity, admissibility, recommendation and provability ACE sentences.

- Coordination: ACE distinguishes between two types of coordination: conjunction “AND” and disjunction “OR”.

In the conjunction case, DRS2CG generate the following CG:

[A] – and ->[B]

And in the disjunction case:

[A] – or ->[B]

“A” and “B” are the CGs corresponding to coordinated sentences (“A and B”, “A or B”).

- Subordination: Subordination means combining elements of unequal syntactic status. In APE, two types of subordination are supported: relative sentences and if-then sentences.

The relative sentence is treated as a separate sentence conjuncted to the rest of the sentence. The ACE sentence “John who is a customer waits” is treated as: “John is a customer and John waits”. In this case, DRS2CG interface is based on the conjunction principle so as to generate the CG.

However, the if-then sentence is treated as an implication element that must contains two elements, the first one represents the antecedent and the second the consequence.

The corresponding CG to an if-then statement generated by DRS2CG is:

[Ant] – imply ->[Conseq]

“Ant” and “Conseq” are the CGs corresponding to Antecedent and the Consequence sentences.

- Quantification: Quantified sentences are used to make assertions about the number of persons or objects that are involved in an event or state, the quantification is translated to an if-then sentence, for example the quantification “Every card is valid.” is translated to “if there is a card then this card is valid.”.

In this case, DRS2CG interface is based on the if-then principle so as to generate the CG (previous part).

- Negation: The negators are used to express that something is not happening, not true or not the case. APE maps a negative sentence to a Negation element in which the sentence is mapped.

Our mapping DRS2CG generates the following CG for a negative sentence:

[Prop] – property ->[veracity: “false”]

“Prop” is the CG corresponding to the sentence on which we are applying the negation.

- Question: DRS2CG supports Yes/No, Where, When, Who, which and How questions. APE maps a question sentence to a Question element in which the sentence is mapped:

[Prop] – property ->[question: questionTypesSet]

questionTypesSet is a set of question types, example : {who, what}.

The same mapping is used in the next sentences types: Command, Possibility, Necessity, Recommendation, Admissibility and Provability.

- Command: The DRS2CG generated CG for a command sentence is:

[Prop] – property ->[Command]

- Possibility: The DRS2CG generated CG for a possibility sentence is:

[Prop] – property ->[Possibility]

- Necessity: The DRS2CG generated CG for a necessity sentence is:

[Prop] – property ->[Necessity]

- Recommendation: The DRS2CG generated CG for a recommendation sentence is:

[Prop] – property ->[Recommendation]

- Admissibility: The DRS2CG generated CG for a admissibility sentence is:

[Prop] – property ->[Admissibility]

- Provability: The DRS2CG generated CG for a provability sentence is:

[Prop] – property ->[Provability]

Here are some examples of the mapping of composite sentences.

- Example 1 (Conjunction): “John eats an apple and Mary waits.”

CG:

```
[cg : [eat : *p1 ] -
      -agent->[Person :John ],
      -obj->[apple]
]-and->[cg : [Person :Mary ]<-agent-[wait]]
```

- Example 2 (If-then): “if the code is valid then the machine accepts the card.”

CG:

```
[ant : [code]-property->[valid]
]-imply->[conseq : [accept : *p1 ] -
               -agent->[machine],
               -obj->[card]
]
```

- Example 3 (Quantification): "Every card is valid."

CG:

[ant : [card : *c1]

]-imply->[conseq : [card : ?c1]-property->[valid]]

- Example 4 (Question): "Is the card valid?"

CG:

[prop : [card]-property->[valid]

]-property->[question : yes_no]

- Example 5 (Necessity): "A customer must enter a card."

CG:

[prop : [enter : *p1] -

-agent->[customer],

-obj->[card]

] -property->[necessity]

- Example 6 (Provability): "A card is not provably valid."

CG:

[prop : [prop : [card]-property->[valid]

]-property->[provability]

] -property->[vericity : "false"]

4 ACE2CG Integration into the Amine Platform

Currently, four actions have been realised toward a full use of ACE2CG and its integration with Amine Platform: a) the development of an ACE2CG Graphical user interface, b) the formulation of Conceptual Structures in ACE, c) the development of a simple Text Analysis and Question/Response System, d) Prolog+CG rules generator from specifications written in ACE. These actions are described in the next sections.

4.1 ACE2CG Graphical User Interface

We have developed a graphical interface which illustrates ACE/APE and ACE2CG mapping (figure 2). This GUI allows user to load his ontology, then edit his ACE text. The activation of the button "Do Syntactico-Semantic Analysis" activates the ACE/APE which produces a DRS structure that is mapped to an equivalent CG by ACE2CG. The result (a CG) is shown in the bottom panel (Figure 2).

4.2 Conceptual Structures in ACE

Amine uses CG to represent knowledge. Also, Amine enables the construction and edition of ontologies composed of conceptual structures (concept type definition, relation type definition, canon, individual, schema/situation, and rules). With ACE/APE and DRS2CG, Amine offers now the possibility to express knowledge and conceptual structures directly in controlled natural language (ACE), instead of forcing user to express his knowledge in CG according to a specific notation.

The first application of ACE2CG (ACE-DRS2CG) interface was its integration into Amine Platform and the formulation of conceptual structures in ACE instead of CG.

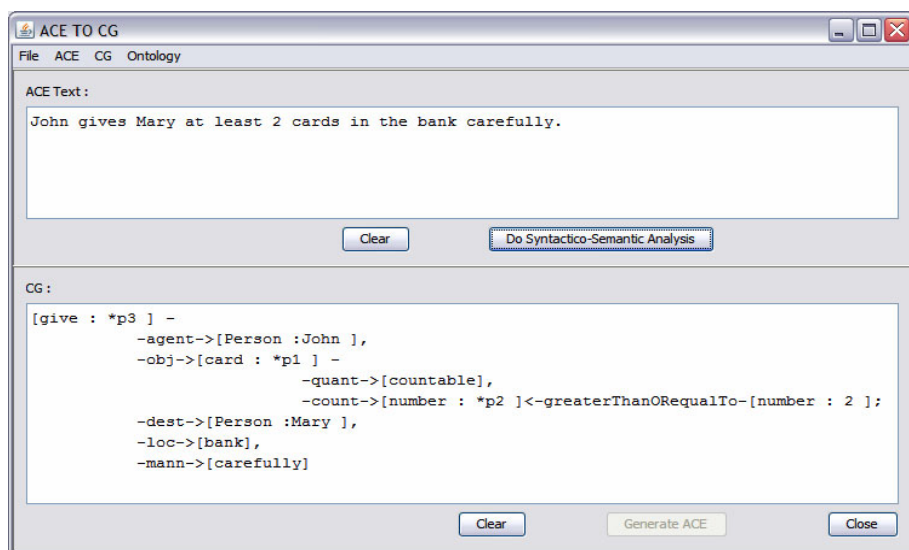


Fig. 2. ACE2CG GUI using sample

A new tab (ACE Editor) has been added to allow user editing text in ACE language instead of CG formalism.

Amine users are therefore able to formulate conceptual structures using Linear Form, CGIF Form [5], Graphical representation or ACE controlled language. Here are some examples that illustrate this new possibility:

- Type Definition and Canon:

We can edit the canon or definition of the type in ACE, on switching the tab, the CG corresponding to ACE definition is generated in different representations (figures 3.a, 3.b, 4.a and 4.b).

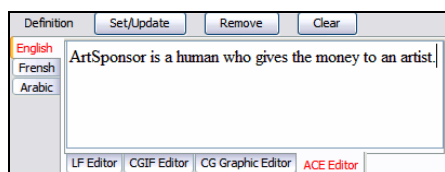


Fig. 3. a: Definition of ArtSponsor in ACE

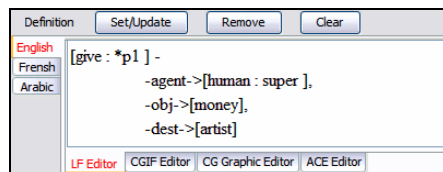


Fig. 3. b: Definition of ArtSponsor in CG LF

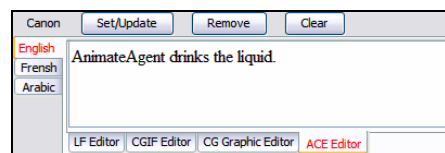


Fig. 4. a: Canon of AnimateAgent in ACE

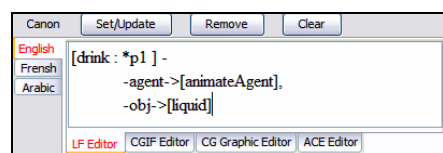


Fig. 4. b: Canon of AnimateAgent in CG LF

- Rules:

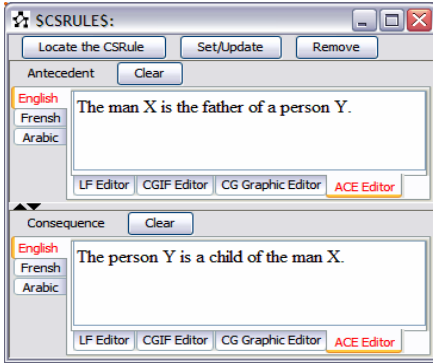


Fig. 5. a: Rule in ACE

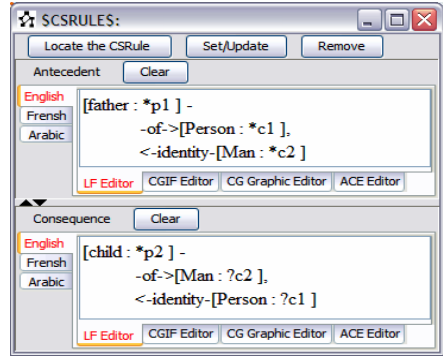


Fig. 5. b: Rule in CG LF

- Situations:



Fig. 6. a: Situation in ACE

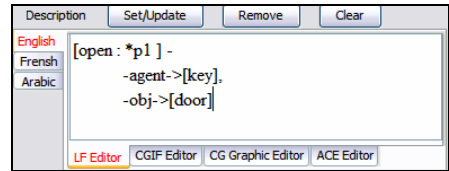


Fig. 6. a: Situation in CG LF

4.3 Text Analysis and Questions Answering System (TAQAS)

The third application of ACE2CG mapping is a simple Text Analysis and Question Answering System based on ACE.

TAQAS is a system that analyses a text given by users and generates the CG corresponding to its semantic.

The user can then ask TAQAS questions about the meaning conveyed by the text. TAQAS analyses the question and generates its CG and through cg operations (like subsume) TAQAS extracts the response CG.

TAQAS can actually trait all allowed question types in APE (Which, What, Who, How, Where and When).

TAQAS was integrated into Amine too with a user friendly interface (figure 7).

Here is an example of question answering for the same ACE text: “John is in the bank in the morning. John gives Mary a card carefully. The code is correct and the bank accepts the card and Mary is happy.”

Please note that TAQAS is just a simple prototype that illustrates what can be done using the mapping of ACE to CG; the negation, recommendation, necessity and the others modality sentences are not supported. It is also noteworthy that the system is only looking for information in the base and does not seek its truth.

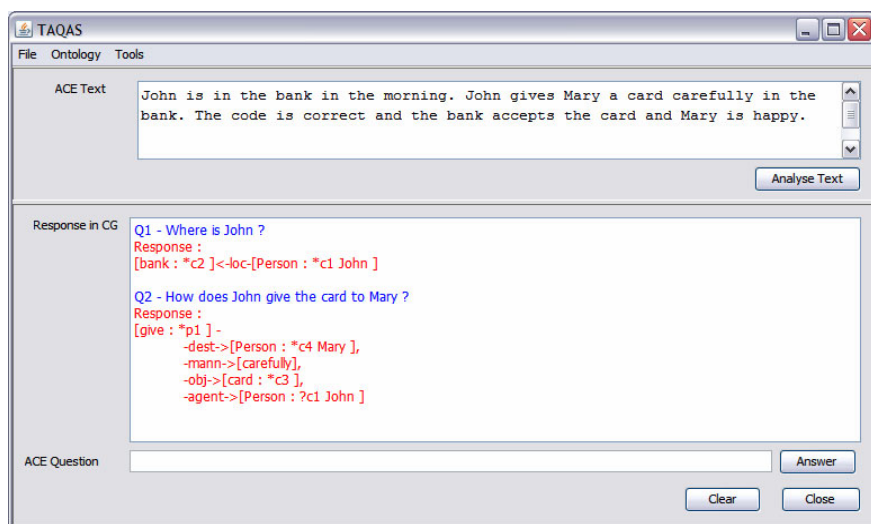


Fig. 7. TAQAS questions analysis example

4.4 ACE as an Executable Specification Language, with ACE2CG and Prolog+CG

Schwitter proposes to use ACE as an executable specification language [12, 14, 15]. As an example, he specifies a library database application as a set of rules expressed in ACE [15]. The result of the syntactico-semantic analysis of the rules correspond to DRS structures that are reformulated in Prolog.

```

ACE Text :
  If a member adds a copy of a book to the library and no entry of the book exists in the catalog
  then the member creates an entry
  and the member enters the id of the copy
  and the copy is available.

Prolog+CG rule :
[cg : *p1 [id : *p3 ] -
  <-attr-[copy : c4 ],
  <-obj-[enter : *p4 ]-agent->[member : c3 ]
] -
-and->[cg : [copy : c4 ]-property->[available]
,
] -
<-and-[cg : [create : *p2 ] -
  -agent->[member : c3 ],
  -obj->[entry]
] :-
[plus : *p1 ] -
  -agent->[member : c3 ],
  -obj->[copy : c4 ]<-poss-[book : c2 ],
  -to->[library],
[ant : [entry : c1 ]<-poss-[book : c2 ]
]-imply->[conseq : [prop : [exist : *p1 ] -
  -agent->[entry : c1 ],
  -in->[catalog]
]-property->[vericity : false ]
] .

```

Fig. 8. ACE2Prolog+CGRule example 1

With ACE2CG and Prolog+CG [3], we obtain a more simple and direct analysis of executable specification using ACE; the analysis of the rules expressed in ACE produces Prolog+CG rules, instead of Prolog rules. Below are the two Prolog+CG rules that result from the mapping of two rules from a specification in ACE:

- “If a member adds a copy of a book to the library and no entry of the book exists in the catalog then the member creates an entry and the member enters the id of the copy and the copy is available.” (Figure 8).
- “If a copy of a book is checked-out to a borrower and a member returns the copy then the copy of the book is available.” (Figure 9).

It should be noted that if the same object has been introduced and is referred in the same rule, ACE2Prolog+CG generates a variable for it, otherwise if it is no referred.

```

ACE Text :
If a copy of a book is checked-out to a borrower and a member returns the copy
then the copy of the book is available.

Prolog+CG rule :
[copy : c1 ]-property->[available] :-
  [copy : c1 ] -
    -property->[checked-out],
    -to->[borrower],
    <-poss-[book],
  [return : *p1 ] -
    -agent->[member],
    -obj->[copy : c1 ].

```

Fig. 9. ACE2Prolog+CGRule example 2

5 Conclusion

In this paper, we presented our work about natural language processing into Amine, especially the mapping from ACE to CG, which allows for processing language specifications written in controlled language ACE to generate conceptual graphs corresponding to their semantics.

This work has been integrated into Amine platform in order to allow users use ACE (which is more convenient) instead of the CG formalism. Thus, as first concrete implementation, the user can now formulate conceptual structures in ACE.

ACE2CG can be the basis of several works, particularly those based upon CGs, some of them have been outlined above as an illustration (syntactico-semantic analysis of Text, Q/A and information retrieval, executable specification), but this list is far from being complete. For instance, this work can be used also in many semantic web applications.

6 Future Works

We are concerned by at least four axes of research:

- Generation of ACE from CG. This can be used in Amine CG editor (generates an ACE formulation for a given CG), in Question/Answering, in translation systems and in many other applications.
- The development of a "controlled Arabic language", we could then develop an Arabic-English translation system based on semantic.
- The development of the use of ACE/CG as an executable specification.
- The extension of ACE in order to support more sentence forms.

References

1. Kabbaj, A.: An Overview of Amine. In: Hitzler, P., Schärfe, H. (eds.) *Conceptual Structures in Practice*, pp. 321–348. Taylor and Francis Group, Chapman and Hall/CRC (2009)
2. Kabbaj, A.: Development of Intelligent Systems and Multi-Agents Systems with Amine Platform. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) *ICCS 2006. LNCS (LNAI)*, vol. 4068, pp. 286–299. Springer, Heidelberg (2006)
3. Kabbaj, A.: PROLOG+CG version 2.0, user's manuel
4. Kamp, H., Reyle, U.: *From Discourse to logic*. Kluwer Academic Publishers, Dordrecht (1993)
5. Sowa, J.F.: Conceptual Graphs for Representing Conceptual Structures. In: Hitzler, P., Schärfe, H. (eds.) *Conceptual Structures in Practice*, pp. 101–136. Taylor and Francis Group, Chapman and Hall/CRC (2009)
6. Sowa, J.F.: Conceptual Graphs. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) *Handbook of Knowledge Representation*, pp. 213–237. Elsevier, Amsterdam (2008)
7. Sowa, J.F.: *Conceptual Structures: Information Processing in Mind and Machines*. Addison-Wesley, Reading (1984)
8. Sowa, J.F.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole Publishing Co., CA (2000)
9. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In: Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S. (eds.) *Reasoning Web. LNCS*, vol. 5224, pp. 104–124. Springer, Heidelberg (2008)
10. Fuchs, N.E., Kaljurand, K., Kuhn, T.: *Discourse Representation Structures for ACE 6.6*, Technical Report ifi-2010.0010, Department of Informatics, University of Zurich (2010)
11. Fuchs, N.E., Höfler, S., Kaljurand, K., Rinaldi, F., Schneider, G.: Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines. In: Eisinger, N., Małuszyński, J. (eds.) *Reasoning Web. LNCS*, vol. 3564, pp. 213–250. Springer, Heidelberg (2005)
12. Fuchs, N.E., Schwertel, U., Schwitter, R.: Attempto Controlled English — Not Just Another Logic Specification Language. In: Flener, P. (ed.) *LOPSTR 1998. LNCS*, vol. 1559, p. 1. Springer, Heidelberg (1999)

13. Fuchs, N.E., Schwertel, U., Torge, S.: Controlled Natural Language Can Replace First-Order Logic. In: 14th IEEE International Conference on Automated Software Engineering, pp. 295–298. Society Press (1999)
14. Schwitter, R.: Controlled English for Requirements Specifications, PhD thesis, Department of computer science, University of Zurich (1998)
15. Schwitter, R., Fuchs, N.E.: Attempto - From Specifications in Controlled Natural Language towards Executable Specifications. In: GI EMISA Workshop, Natürlichsprachlicher Entwurf von Informationssystemen, Tutzing, Germany (May 1996)

Identifying Relations between Medical Concepts by Parsing UMLS[®] Definitions

Ivelina Nikolova and Galia Angelova

Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences
25A, Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
{iva,galia}@lml.bas.bg
<http://lml.bas.bg/>

Abstract. To automatically analyse medical narratives, one needs linguistic and conceptual resources which support capturing of important information from texts and its representation in a structured way. Thus the conceptual structures encoding domain concepts and relations are crucial for the development of reliable and high-performance information extraction system. We present research work enabling automatic extraction of relations between medical concepts. The lack of conceptual resources with Bulgarian ontological vocabulary provoked us to reuse already existing resources with English labels, more especially the UMLS[®] Metathesaurus[®]. We form a terminological dictionary of the Bulgarian terms of interest, translate them to English and extract their UMLS definitions which are short English statements in free text. These definitions are processed automatically by a semantic parser; afterwards we apply additional extraction, alternation and validation rules and built a set of new relations to be inserted in our conceptual resource. The article presents the input data and available tools, the knowledge chunks extracted from UMLS and their processing, as well as a discussion of the present results.

Keywords: relation extraction, clinical terms, biomedical NLP.

1 Introduction

Secondary use of medical health records is an important research trend which has been given high priority especially in the recent years. Processing patient data by all possible means will enable the improvement of the clinical decision-support systems, health management and treatment through provision of personalised healthcare services. There are various ways to reuse the information stored in the electronic health records; an important activity is to automatically analyse the free text paragraphs which present the most important findings regarding the case history. The records contain focused information and differ from patient to patient, from author to author. In order to unify the structure of the extracted descriptions and produce a common template where each patient's feature with

its attributes and values has a separate slot, we need a kind of conceptual framework which supports the segmentation and extraction of focal information from the raw text. Such a framework could be provided by an ontological resource covering the medical concepts/terms used in the patient records and the relations between them. Thus knowledge about concepts and relations is crucial for building reliable and high-performance information extraction (IE) systems.

Analysing automatically Patient Records (PRs) in Bulgarian language, we need underlying conceptual resources labeled by Bulgarian medical terms (to be able to map the text units onto conceptual entities). Unfortunately no medical ontologies with Bulgarian vocabulary exist (except the International Classification of Diseases ICD) but there are large resources available for other languages. The UMLS Metathesaurus [1], which is widely used in biomedical Natural Language Processing (NLP), contains information about biomedical and health related concepts, their various names in several languages, and some labeled relationships among them. One finds their descriptions of various depth and quality: comprehensive, systematic representation of concepts and relations in some areas and schematic or poor coverage in other areas.

We present here research efforts and experiments done in order to enrich a domain model supporting IE from hospital PRs in Bulgarian language. In general we aim at the construction of a task-specific Bulgarian terminological bank enriched with relations that are obtained automatically by reusing existing resources in English. We have built a terminological dictionary containing Bulgarian terms of interest found in the training corpus of PRs, translated these terms to English and extracted their UMLS definitions which are short English statements in free text. These definitions were processed automatically by a semantic parser, that transforms them to dependency and semantic framing structures which easily map to concept graphs (CGs). Afterwards we have applied additional extraction, alternation and validation rules and have built a set of new relations to be inserted in our conceptual resource. The article will present the work done so far with focus on the conceptual processing: the input data, the knowledge chunks extracted from the UMLS system using the available tools, the procedures for automatic processing of the free text definitions in order to extract relations between clinical terms, and the evaluation of the present results. To the best of authors knowledge this task has been explored for Bulgarian only under the current project [2].

The article is structured as follows: section 2 summarises related work regarding automatic relation acquisition; section 3 considers the background resources and tools; section 4 presents our relation extraction approach; section 5 discusses the results and section 6 contains the conclusion.

2 Related Work

Defining relations between concepts is a puzzling knowledge representation exercise since the relations in general reflect implicit and task-dependent connections between entities. All attempts to unify the AI approach to relation elicitation

have failed; for instance we still see practical solutions where relations between concepts are labeled by verbs and the concept labels are interpreted as verb role fillers, e.g. *subject-object*; on the other hand using the verb thematic roles (e.g. *agent, object, instrument*) as conceptual relations is considered a good style of conceptual design because it enables to address systematically most domain entities of interest [3]. Given some application area, e.g. medicine and healthcare, the natural choice for acquisition of *concepts* is to juxtapose concepts to important terms (most often nouns); however, there could be a variety of approaches and application-dependent considerations regarding the definition of *relations*. This is clearly seen in the largest collection of medical terms UMLS: it comprises more than 100 nomenclatures, controlled vocabularies and terminology systems with over 5 million concept names; the UMLS Metathesaurus is organised by concepts but the most often relations are the 'classical' *IS-A* and *part-of* (even these relations are not always encoded). Many available 'properties' convey either very general relationships or relationships that are hard to interpret in the NLP context [4]. In this way automatic acquisition of relations is a hot research task, especially in large domains where manual elicitation is almost impossible, and there is a variety of application-specific solutions to explicate some of the numerous relations existing between the concepts.

Usually we assume that conceptual relations between entities can be (semi-) automatically acquired by (i) automatic identification of *linguistic* relations between the corresponding terms in some text descriptions, and (ii) filtering and refinement of the linguistic relations in the process of their interpretation as *conceptual* relations. Actually the text-based acquisition of domain entities is applied for concept elicitation as well, e.g. [5] describes a plug-in OntoLT for the widely used Protégé ontology development tool that supports the interactive extraction and/or extension of ontologies from text. The linguistic analysis is integrated with ontology engineering through the definition of mapping rules that map linguistic entities in annotated text collections to concept and attribute candidates (i.e. Protégé classes and slots). In this way a shallow ontology for the neurology domain was derived from a corresponding collection of neurological scientific abstracts. Focusing on relation extraction, the system RelExt extends an ontology by automatically identifying highly relevant triples (pairs of ontology concepts connected by a relation) from a domain-specific text collection. RelExt works by extracting relevant verbs and their grammatical arguments (i.e. terms) and computing corresponding relations through a combination of linguistic and statistical processing [6]. A system with similar name - RelEx - supports relation extraction from free text [7]. RelEx is based on NL preprocessing producing dependency parse trees and applies a small number of simple rules to these trees. RelEx was evaluated on a comprehensive set of one million Medline abstracts dealing with gene and protein relations and extracted approximately 150,000 relations with an estimated performance of both 80% precision and 80% recall [8].

Regarding the automatic processing of relations in UMLS, [9] analyses the potential of using ontological relations to produce correct semantic structures for a medical document automatically. Presenting a method called SeReMeD,

the article discusses an approach to generate representations of unstructured medical narratives. The method makes use of UMLS concept relations and UMLS Semantic Network (SN) semantic types to acquire additional semantic relations and support the structuring process. The results show that the relations can enhance and ameliorate the automatically generated semantic structures.

There are a lot of studies for relation extraction but we are focused on extraction from short medical definitions and elaboration of constraints (e.g. filtering rules) that might help to refine and interpret the discovered relations. Therefore, we have studied approaches for relation processing as well. The article [10] presents a method for relation filtering and a method to discover new relation instances that were developed in the context of cross-language information retrieval (CLIR) and exploit semantic annotation, particularly semantic relations, in the medical domain. As the baseline for automatic semantic annotation [10] uses the existing semantic relations between medical concepts in UMLS. Both methods were applied to a corpus of English and German medical abstracts and evaluated for their efficiency in CLIR. Results show that filtering reduces recall without significant increase in precision, while discovery of new relation instances indeed proved a successful method to improve retrieval. Another article suggesting helpful hints is [11]; it reports about experiments for identifying and evaluating context features and machine learning methods to identify medical semantic relations in texts (more precisely Medline abstracts). Using hierarchical clustering the authors compare and evaluate the linguistic aspects of relation context and different data representations. Through feature selection on a small data set they show that relations are characterised by typical context words, and by isolating these they can construct a more robust language model representing the target relation.

As shown above, to accomplish the task of relation extraction from biomedical texts, researchers use wide-ranging techniques that take advantage of domain, statistical, and linguistic information. Although some studies focus on only one technique, the majority integrate multiple methods to accomplish their aims. These experiments leverage the power of statistical pattern matching but also integrate linguistic characteristics and expert knowledge; this combination is essential to the holy grail of biomedical natural language understanding [12].

The environment OntoLT [5], dealing with ontology extraction from text, proposes a precondition language for defining *mapping rules*. Preconditions are implemented as XPATH expressions over the XML-based linguistic annotation. If all constraints are satisfied, the mapping rule activates one or more operators that describe in which way the ontology should be extended if a candidate is found. We find this idea useful for the elaboration of our approach to relation extraction.

3 Prerequisites

We apply a pipeline of existing open source NLP tools and new software components to process automatically various linguistic and conceptual resources. The background is presented here.

As a terminological framework to support our relation extraction task we employed the UMLS Metathesaurus. The advantage of UMLS as a terminological bank is that it contains well documented, consistently structured information, moreover all resources have the same internal representation in RRF text format and are easy to process. In addition there are tools which support the browsing and extraction of term-related information. We used the Metathesaurus for two purposes: *(i)* to filter more carefully the terms that we want to process and to select a concept label for them, and *(ii)* to extract term definitions, which are short paragraphs written in a domain-specific language (in contrast to longer Wikipedia articles where the definitions are written in a popular style).

Another tool we applied out of the box is the relation extractor RelEx [7]. RelEx is a syntactic dependency extractor and semantic framing generator; it parses English language sentences and returns the dependency relationships between different parts of the sentence, and also provides semantic framing tags based on syntax and semantic categories. The core component extracts the dependency relationships. Additional modules perform functions such as anaphora resolution and provide semantic frame output. The Link Grammar Parser [15] is the underlying engine, providing the core sentence parsing ability. Wordnet [18], [19] is used to provide basic English morphology, such as singular versions of (plural) nouns, base forms (lemmas) of adjectives, adverbs and infinitive forms of verbs. Dependency grammar, as formulated by Lucien Tesnière, was one of the influences on the development of conceptual graphs and related versions of semantic networks and similar graph representations for syntax and semantics. Dependency parsers and the related link parsers are useful for generating graphs that have a simple mapping to CGs. In the next section we show examples of parse trees produced in LinkGrammar. We stress that any parser labels the links between the sentence objects only by names of syntactico-semantic linguistic relationships which reflect the sentence structure and general linguistic knowledge about semantic connections between sentence phrases; in other words no domain-specific relations appear as tags in any parse tree.

Our vocabulary of interest is constructed by a bottom up approach: we have analysed automatically the patient status in a corpus of 1200 hospital PRs of diabetic patients and have extracted important clinical terms. The extraction was done semi-automatically starting with an initial list of often used terms and augmenting it iteratively by an expert. The final set of terms was translated to English (where possible more than one translation was given) and the English nouns were lemmatised. The terms denoting diabetes complications were justified using the diabetes ontology at the Biomedical portal [13]. Please note that extraction of UMLS definitions is only possible if the exact (up to lemmatisation) UMLS entry is specified, therefore the English terms have to be represented in their UMLS format.

Hence, the next preprocessing step includes defining the final term list in a sense of UMLS semantic concepts - the terms were turned to single words or phrases of 2-3 words. This task was supported by continuous browsing of the UMLS resources. By extracting the concept identifiers from UMLS we actually checked the availability of the term in the Metathesaurus. Once a term was found

in the Metathesaurus, an expert manually selected which of the corresponding concepts are of interest for our study.

In our opinion many important domain-specific relations are rarely declared explicitly in single, well structured sentences (to be automatically extracted from there with the algorithms of the present NLP tools). Therefore we also looked for sources providing hints and insights about human experts' perspective to possible relationships between medical concepts. One such source is the list of relations in the UMLS Semantic Network [16]. These 52 relations are not explicitly encoded between most UMLS concepts but they present an instance of an expert perspective how medical concepts might be interconnected. Having such a list at hand simplifies a bit the relation extraction task because one knows what should be extracted; e.g. we can aim at the extraction of *affect* or *cause* from the definition sentence.

4 Extracting Relations from UMLS Definitions

4.1 Definition Extraction

The text corpus in our experiment is a collection of term definitions extracted from the UMLS Metathesaurus. Each term in UMLS can name one or more concepts. Each concept is given a unique identifier for the entire UMLS database - the so-called Concept Unique Identifier (CUI). Each concept encodes a different meaning for certain term and the concept definition is given in some source vocabulary (within the hundred resources integrated under UMLS). Sometimes a concept can be encoded without any text definition. Often terms name more than 2 concepts (up to 5) and some concepts might have up to 8 text definitions in the various resources; obviously not all of them are subjects of our study. We used remote access to the UMLS servers provided by the UMLS Terminology Services (UTS) API [17] and extracted all possible definitions for the terms of interest. These are standard procedures provided by the UTS.

Table 1 presents sample definitions of concepts obtained when submitting the term "pulse" to the UTS service. The result contains the concepts' CUIs, in this case *C0391850* and *C0034107*, their corresponding labels "*Physiologic pulse*" and "*Pulse taking*" and the definitions. The first concept has only one definition and the second one has two definitions, extracted from various UMLS vocabularies. Queries to the UTS were sent for all terms of interest and after the definition extraction, an expert filtered out only the definitions of interest which are suitable for the analysed conceptual subset. For instance two definitions were selected from Table 1: *Def1* for "*Physiologic pulse*" and *Def1* for "*Pulse taking*" but *Def2* is removed because it is irrelevant for our experiment. Where definitions contain more than one sentence we analyse only the first one of them as our assumption is that it carries the most important information defining the concept.

Table 1. Extracted definitions for the term "pulse" from UTS

Input term	CUI	Output term	Definitions
pulse	C0391850	Physiologic pulse	Def 1. The rhythmic wave within the arteries occurring with each contraction of the left ventricle.
pulse	C0034107	Pulse taking	<p>Def 1. The rhythmical expansion and contraction of an ARTERY produced by waves of pressure caused by the ejection of BLOOD from the left ventricle of the HEART as it contracts.</p> <p>Def 2. Actions performed to measure rhythmical beats of the heart.</p>

4.2 Definition Parsing and Relation Explication

The system RelEx, which is employed as a basic tool at this stage, works in the following way:

- Step 1:* Executes the Link Parser and converts the Link Parser output to a feature structure representation;
- Step 2:* Executes a series of Sentence Algorithms which modify the feature structure;
- Step 3:* Extracts the final output representation by traversing the feature structure.

From the RelEx output we use both the dependency parse of the definition sentence and the semantic framing relations shown on figure 2 and combine them in a conceptual structure which is further used for inference of new relations. We apply further application-specific rules on the text, which improve the relation elicitation process. The rules are developed by studying the available corpus from task-specific perspective. The parser performance at Step 1 is much better after the following transformations are done (examples are shown on table 2):

- (i) First letter of each sentence was capitalised;
- (ii) Dot was put in the end of each definition which lacked it;
- (iii) Examples were removed from the definitions;
- (iv) Special mark-ups pointing to source vocabularies were removed;
- (v) Pronunciation transcriptions were removed;

Transformations of the syntactic structures. At step 1 RelEx came across few obstacles, mostly due to the very complex or too simple syntax of the analysed definitions. Only 34% of the definitions were completely parsed and the rest 66% failed in recognising at least one of the words. Since parsing is build

Table 2. Normalisation of the extracted UMLS definitions

Original Definition	Corrected Definition
(eh-DEE-ma) Swelling caused by excess fluid in body tissues.	Swelling caused by excess fluid in body tissues.
A blood vessel that carries blood away from the heart. (NCI)	A blood vessel that carries blood away from the heart.
swelling from excessive accumulation of serous fluid in tissue.	Swelling from excessive accumulation of serous fluid in tissue.
The controlled release of a substance by a cell. [GOC:mah]	The controlled release of a substance by a cell.

around the sentence verb, definitions which are lacking a verb are not juxtaposed a parse tree and respectively no semantic framing was performed(see example 1).

Example 1. (LIMB) A body region referring to an upper or lower extremity.
(MUSCLE) One of the contractile organs of the body.
(PHALANX OF HAND) A bone of the hand.

We transformed these phrases to sentences by adding in the beginning the term which is defined followed by the verb "to be". Thus the sentences from example 1 were changed to:

Limb is a body region referring to an upper or lower extremity.
Muscle is one of the contractile organs of the body.
Phalanx of hand is a bone of the hand.

Parsing failures are also due to improper punctuation and structure like in example 2. We also noticed that the parser sometimes fails to disambiguate verbs/nouns for words like e.g. measure, joint, etc.; sometimes nouns are parsed as verbs which is the case with the sentences in table 3.

Example 2. Touch; the faculty of touch, the sensation produced by pressure receptors in the skin.

Table 3. Wrong disambiguation of verbnoun POS tags

Sentence	Constituency Parse Tree
Touch; the faculty of touch, the sensation produced by pressure receptors in the skin.	(S (VP touch [;] (NP (NP the faculty) (PP (NP the sensation produced by pressure receptors in the skin))) , (NP (NP the sensation) (VP produced (PP by (NP pressure receptors)) (PP in (NP the skin)))) .))))
A joint connecting the lower part of the femur with the upper part of the tibia.	(S [a] (S (VP joint [connecting] (NP (NP the lower part) (PP of (NP the femur))) (PP with (NP (NP the upper part) (PP of (NP the tibia)))))) .)

The final output of RelEx includes constituency and dependency parse trees (including the set of all dependency relations and features), a link grammar parse tree and relations determined by semantic framing rules, such as for identifying the discourse entities:

1_Entity:Entity(Diabetic_Cataract, Diabetic_Cataract)

The upper part of figure 2 contains two RelEx outputs for the sentence *Diabetic Cataract is a rare, usually bilateral, opacity shaped like a snowflake, affecting the anterior and posterior cortices of young diabetics.*

4.3 Adding New Relations to the Original Relation Set

IS_A Relation

Using the dependency graph, the regular structure of the narrative sentence - Subject Verb Object (SVO), the fact that the definitions are written in a very short form, and following a commonly accepted rules about meaning composition, we are able to infer the "IS_A" relation out of some parsed definitions. After transforming to sentences the definitions-phrases like the ones in example 1, some 10% of the definitions have exactly this structure.

Rule 1. If we have the dependency relations corresponding to subject, object, and an object and/or subject modifier in the main sentence marked as:

- _obj(be, N3)
- _subj(be, N1)
- _tense(be, present)
- _nn(N1, N0)
- _nn(N3, N2)

then we can infer that N1 IS_A N3. The last two relations are optional and if present N0 would be modifier to N1 and N2 to N3 correspondingly.

Example 3. Given the definition: "Artery is a blood vessel that carries blood away from the heart." The extracted relations of interest are shown on the graph on figure 1 as solid and the inferred IS_A relation is marked dashed.

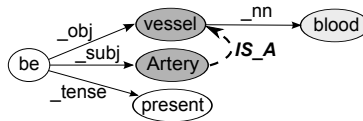


Fig. 1. Dependency graph explicating the inference of IS_A relation by Rule 1

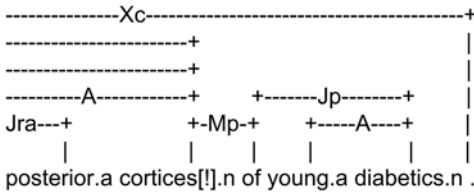
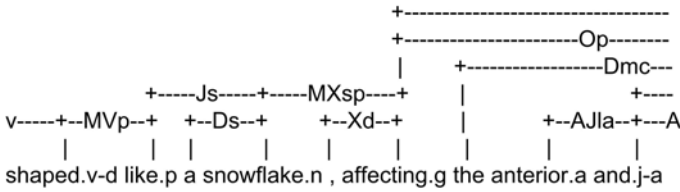
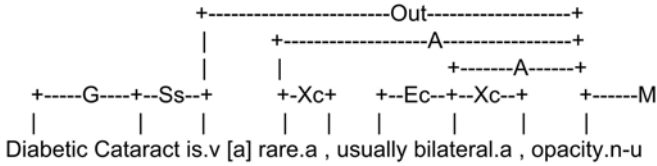
AFFECTS Relation

Another important relation is the AFFECTS relation, especially between complications/diseases/symptoms and the organs they affect. We consider examples extracted from definitions of Diabetes Mellitus complications. The RelEx syntactic and semantic analysis delivers the entities occurring in the definitions for further processing of each definition. The lexico-semantic structure, needed

Constituency parse

(S (NP Diabetic Cataract) (VP is [a] (NP (NP (ADJP rare ,) (ADJP (ADVP usually) bilateral ,) opacity) (VP shaped (PP like (NP a snowflake (VP , affecting (NP (NP the (ADJP anterior and posterior) cortices) (PP of (NP young diabetics))) .))))))))))

LinkGrammar parse



Dependency tree

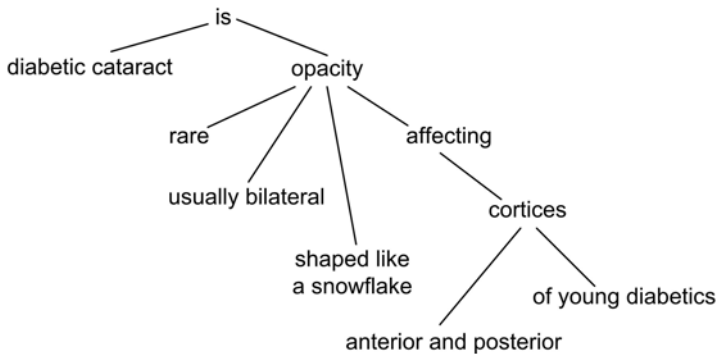


Fig. 2. RelEx outputs and illustration of the dependency skeleton supporting relation elicitation

for relation extraction, is elaborated in five steps which are illustrated here for the sentence *Diabetic Cataract is a rare, usually bilateral, opacity shaped like a snowflake, affecting the anterior and posterior cortices of young diabetics.*

Step 1: Construct a list of the entities obtained from RelEx.

Entities {Diabetic_Cataract, opacity, snowflake, cortex, affecting, Diabetic, diabetic, like}

Step 2: Augment each entity with its possible modifiers collected at the parsing stage. These could be adjectives or nouns in the role of adjectives, forming a noun phrase, or compounds connected by preposition extracted from the parser (e.g. *accumulation of amount*). Skip the modifiers which are stop words (e.g. *and, or, due etc.*).

Entities extended with modifiers {Diabetic_Cataract, opacity, bilateral opacity, rare opacity, snowflake, shape like snowflake, cortex, *and cortex*, affecting, *Diabetic*, diabetic, young diabetic, like}

At this step the following terms were subtracted: *and cortex*, because *and* is a stop word; *Diabetic*, because it is already in the list and *Diabetic_Cataract* is transformed to *Diabetic Cataract*.

Step 3: Search in UMLS for the words and compounds from *Step 1* and *Step 2* in order to prove which of them are medical terms. Extract the terms which have the least Hemming distance to the initial term.

Entities recognised by UMLS as medical terms {Diabetic Cataract (Diabetic Cataract), Retinal opacity (opacity), Snowflake retinal degeneration (snowflake), Visual Cortex (cortex), affecting (affecting), diabetic (diabetic)}

Step 4: Assign the UMLS semantic type to the terms extracted at *Step 3* given by UMLS and obtain the following figure.

Semantic type attachment (semantic types are in the brackets): {Diabetic Cataract (*Disease or Syndrome*), Retinal opacity (*Finding*), Visual Cortex (*Body Part, Organ, or Organ Component*), snowflake (*Acquired Abnormality*), affecting (*Functional concept*), diabetic (*Finding*)}

Step 5: Construct a new lexico-semantic representation corresponding to the initial definition and including the semantic types.

Lexico-semantic representation: (Disease or Syndrome) is a rare, usually bilateral, (Finding) shaped like a (Acquired Abnormality), affecting the anterior and posterior (Body Part, Organ, or Organ Component) of young (Finding).

Step 6: Infer the AFFECTS relation and its arguments given the features collected on the previous steps and the dependency information available from the RelEx output.

In contrast to the "bag of words" approach, these steps give us the advantage to deal at *Step 6* with concrete medical terms, to have their possible modifiers,

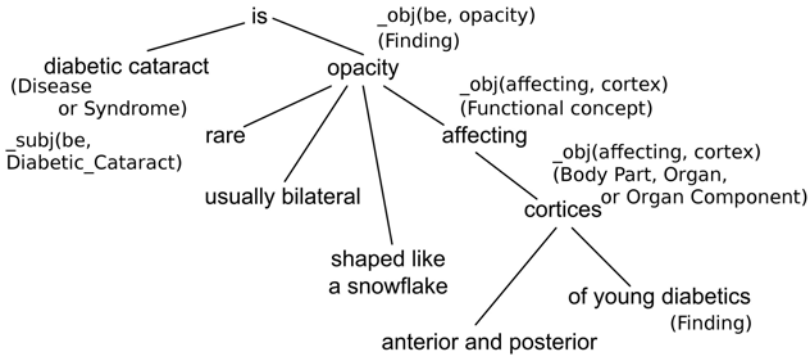


Fig. 3. Feature integration and relation selection

and a dependency representation helping to resolve the concept which is being affected. Since our approach is bottom-up and we are looking for concrete relations between the diseases and the organs they affect, and given the fact that the analysed text is a definition, the dependency relation *subject* is of crucial importance to us. That is why at this final step we organise all features selected in the previous step as shown on figure 3.

Rule 2. If we have dependency and semantic structures matching the following conditions:

_subj(be, N1) AND N1 has Semantic Type *Disease or Syndrome*

_obj(be, N2) AND N2 has Semantic Type *Disease or Syndrome* or *Finding*

A verb V singling AFFECT relation

_obj(V, N3) AND (N3 has Semantic Type (*Body Part, Organ, or Organ Component*) OR the augmented entity of N3 identifies has Semantic Type *Body Part, Organ, or Organ Component*

Then we infer by this rule that N1 AFFECTS N3

and by *Rule 1* that N1 IS-A N2.

E.g. *Diabetic Cataract* IS-A *opacity* AND *Diabetic Cataract* AFFECTS *cortices*.

5 Discussion

In this experiment we parsed 194 definitions corresponding to 129 terms. Lexical constructions referring to Body parts or Organs were found and further analysed in 57% of the definitions. The analysis of the constituency parsing proved that the focal terms (organs/body parts) are always located in a prepositional phrase attached to the verb of interest. We studied the patterns matching the AFFECTS relation and made a list of verbs expressing the availability of this relation.

The lexical expressions occur in active or passive voice. Most frequent lexical patterns are: *affecting*, *consisting of* and *characterized by*. The phrase "resulting from" signals the availability of another disease or condition, which triggers the disease of interest.

The extraction of the IS-A relation was done with 81% precision, whereas in the subset of definitions with automatically transformed syntactic structure this result is as high as 89%. There are several reasons for the lower performance over the whole collection; the first one is wrong parse trees, due to the complicated syntactic structure; another one is the partial recognition of compound terms which were mapped to subject and/or object which reverts the relation recognition. The recall for definitions with explicitly stated IS-A relations is 86%.

The extraction of the AFFECTS relation performed, as expected, worse than IS-A extraction, because the variety of affect expressions is much higher. Another complication is due to the fact that the arguments of AFFECTS may be positioned in the sentence in longer distance from each other, thus the parsing errors imply incorrect inferences by Rule 2. Some of the words signaling AFFECTS are ambiguous and lead to a different relation.

6 Conclusion

Our goal in this study was to prove the availability of NLP tools and resources as well as their readiness to serve for developing of new conceptual resources. We presented an approach enabling automatic extraction of relations between medical concepts by reusing existing resources and NLP tools and applying additional transformation rules. By analysing the output in the intermediate steps we notice that often the failure of our algorithm is due to wrong parsing trees.

The IS-A relations we extracted were often available in the UMLS Metathesaurus, but some 45% are newly created. As for AFFECTS, only 3 of the extracted relations were available in the Metathesaurus and they were not concretely specified, but available as concept relations *has relationship other than synonymous, narrower, or broader*. While analysing the results in the intermediate steps we noticed that often the failure of the algorithm was due to an error in the parsing stage. Therefore a better parsing would lead to improvement in the relation extraction algorithm as well.

In the future we plan to do a more thorough evaluation of the learned rules and go into detail of the extraction relations towards specifying their different subtypes such as *functionally_related_to manages, treats, disrupts, complicates, interacts_with, prevents*.

Acknowledgments. The research work presented in this paper is partly supported by grant DO 02-292/December 2008 "Effective search of conceptual information with applications in medical informatics", funded by the Bulgarian National Science Fund in 2009-2012 [2].

References

1. Terminological Services of Unified Medical Language System (UMLS), <https://uts.nlm.nih.gov/home.html>
2. Project Effective search of conceptual information with applications in medical informatics, <http://www.lml.bas.bg/evtima>
3. Sowa, J.: *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading (1984)
4. Nirenburg, S., McShane, M., Zabłudowski, M., Beale, S., Pfeifer, C.: *Ontological Semantic Text Processing in the Biomedical Domain*. University of Maryland Baltimore County, Institute for Language and Information Technologies, Working Paper, 3-5, http://naboo.ilit.umbc.edu/ILIT_Working_Papers/ILIT_WP_03-05_Biomed_Mesh.pdf (last visited February 2011)
5. Buitelaar, P., Olejnik, D., Sintek, M.: A Protg Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) *ESWS 2004*. LNCS, vol. 3053, pp. 31–44. Springer, Heidelberg (2004)
6. Schutz, A., Buitelaar, P.: RelExt: A Tool for Relation Extraction from Text in Ontology Extension. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A., et al. (eds.) *ISWC 2005*. LNCS, vol. 3729, pp. 593–606. Springer, Heidelberg (2005)
7. RelEx, <https://launchpad.net/relex>
8. Fundel, K., Kffner, R., Zimmer, R.: RelEx - Relation extraction using dependency parse trees. *Journal of Bioinformatics* 23(3), 365–371 (2007)
9. Denecke, K.: Enchancing Knowledge Representations by Ontological Relations. In: Andersen, K., et al. (eds.) *Proc. of MIE 2008, eHealth Beyond the Horizon - Get IT There*, Goteborg. *Studies in Health Technology and Informatics*, vol. 136, pp. 791–796. IOS Press, Amsterdam (2008)
10. Vintar, S., Buitelaar, P., Volk, M.: Semantic Relations in Concept-based Cross-language Medical Information Retrieval. In: *Adaptive Text Extraction and Mining (ATEM)*, Cavtat-Dubrovnik (2003)
11. Vintar, S., Todorovski, L., Sonntag, D., Buitelaar, P.: Evaluating context features for medical relation mining. In: *ECML/PKDD Workshop on Data Mining and Text Mining for Bioinformatics* (2003)
12. Chapman, W., Cohen, K.B.: Current issues in biomedical text mining and natural language processing. *Journal of Biomedical Informatics* 42(5), 757–759 (2009)
13. Diabetes Ontology at the BioPortal provided by the Medical Dictionary for Regulatory Activities Terminology (MedDRA), <http://bioportal.bioontology.org/visualize/42280/?conceptid=10012614> (last visited February 2011)
14. Browne, A.C., Guy, D., Aronson, A., McCray, A.: UMLS language and vocabulary tools. In: *Proceedings Annual Symposium AMIA* p. 798 (2003)
15. Link Grammar Parser, <http://www.link.cs.cmu.edu/link/>
16. Current Relations in the UMLS Semantic Network, http://www.nlm.nih.gov/research/umls/META3_current_relations.html
17. Terminological Services of Unified Medical Language System (UMLS), Developer's Guide, <https://uts.nlm.nih.gov/doc/devGuide/index.html>
18. Miller, G.A.: *WordNet: A Lexical Database for English*. *Communications of the ACM* 38(11), 39–41 (1995)
19. Fellbaum, C. (ed.): *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge (1998)

Topicality in Logic-Based Ontologies

Chiara Del Vescovo, Bijan Parsia, and Ulrike Sattler

The University of Manchester, Oxford Road, Manchester, M13 9PL, UK
{delvescc,bparsia,sattler}@cs.man.ac.uk

Abstract. In this paper we examine several forms of modularity in logics as a basis for various conceptions of the topical structure of an ontology. Intuitively, a topic is a coherent fragment of the subject matter of the ontology. Different topics may play different roles: e.g., the main topic (or topics), side topics, or subtopics. If, at the lowest level, the subject matter of an ontology is characterized by the set of concepts of the ontology, a topic is a “coherent” subset of those concepts. Different forms of modularity induce different, more or less cognitively helpful, notions of coherence and thus distinct topical structures.

1 Introduction

When formalising a set of concepts in some logic we encounter a variety of structural issues. For example, we look for *definitions* of concepts in terms of categorizing attributes, that is, for the “internal” structure of our concepts. We also seek to discover “external” relations between concepts, e.g., of subsumption, equivalence, or disjointness. In a logic based knowledge representation system, we can hope that by giving the former, the system can discover the latter.

Common such systems are those based on description logic *ontologies*. Ontologies are (decidable) logical theories describing a shared vocabulary about a domain in terms of a set of concepts plus the relationships between those concepts. In notable examples, as SNOMED CT¹ (Systematized Nomenclature of Medicine – Clinical Terms) that contains more than 300,000 axioms, while the internal structure of any concept may well be intelligible, the large scale structure is not. In particular, the subsumption hierarchy, as a whole, is not particular well suited to guide the interested reader toward a grasp of “what the ontology is about.” All concepts – relevant or irrelevant, well or cursorily described, central or peripheral – participate, at least vacuously, in the subsumption hierarchy. One could attempt to heuristically organise concepts in the subsumption hierarchy in some larger grain way so that more strongly related terms were clustered together (e.g., as in [15]). With such a coarse grain structure, one might hope to discern the “main topics” of the ontology, various side topics, as well as topics that are neglected in the formalisation. Unfortunately, such heuristic organisation, even if based on the prior, logic-derived structural aspects of concepts, is not, itself, derived from logical features of the representation.

¹ <http://www.ihtsdo.org/snomed-ct/>

That is, it is not derived from, nor is it properly sensitive to, the semantics of our representation. Ideally, there would be a notion of logical topicality which, 1) comports with our intuitions and expectations about topical organization, 2) is computationally reasonable, and 3) supports ontology engineers in practice.

A promising foundation for logical topicality is the extensive recent work on *logically sensible* modules, that is, modules which offer strong logical guarantees for intuitive properties of modules [2]. For example, one such guarantee is called *coverage* of a set of terms (*signature*), and means that the module captures all the ontology’s knowledge about this set. It is easy to see that such a notion is a reasonable candidate to underly topicality. The aim of this paper is to lay foundations for a theory of logical topicality in ontologies, by considering five different forms of logical modularity and possible sorts of topicality they induce.

2 Preliminaries

In this paper, we are concerned with logic-based ontologies, i.e., finite sets of axioms, formulated in a suitable logic. Our main focus is on Description Logics (DLs) [1] – decidable fragments of first order logics closely related to modal logics that form the logical underpinning of state-of-the-art ontology languages such as OWL 2² [4] – but our discussion is straightforwardly applicable to other logical formalisms.

A *signature* is a set of *terms*; in DLs, these are *individual* names to denote elements of the domain (constants in FOL), *concept* names to denote monadic predicates, and *role* names to denote binary predicates. A given DL, say *SHIQ*, determines the set of *axioms* we can form over terms, and any finite set of axioms is an *ontology*. For \mathcal{O} an ontology, we will use \mathcal{O} for the set of terms occurring in \mathcal{O} , i.e., for its signature. For example, the following axioms characterize animals, define parents, and assert facts about two individuals:

$$\begin{aligned} \text{Animal} &\sqsubseteq \text{Organism} \sqcap \neg \text{Plant} \\ \text{Parent} &\equiv \text{Animal} \sqcap \exists \text{hasChild}.\text{Animal} \\ \text{Animal}(\text{peter}), \quad &\text{Parent}(\text{mary}), \quad \text{hasChild}(\text{peter}, \text{mary}) \end{aligned}$$

Traditionally, in DLs, an ontology consists of two parts, namely a *TBox* – those axioms that involve only concepts and roles, like the first two axioms above – and an *ABox* – those axioms that involve individuals, like the last three axioms above. For our considerations, however, this distinction can be mostly neglected.

We assume that our logic comes with an entailment relation \models . For DLs, this relation is the standard, first order entailment, which (standardly for these logics) is not only decidable but for which various decision procedures have been implemented.

Finally, we make use of the standard notion of *conservative extension*: roughly speaking, for Σ a signature and ontologies $\mathcal{O}_1 \subseteq \mathcal{O}_2$, we say that \mathcal{O}_2 is a (deductive) Σ -conservative extension of \mathcal{O}_1 if $\mathcal{O}_1 \models \alpha$ implies $\mathcal{O}_2 \models \alpha$, for every

² <http://www.w3.org/TR/owl2-overview/>

axiom that we can build over Σ . That is, if \mathcal{O}_2 is a (deductive) Σ -conservative extension of \mathcal{O}_1 , then \mathcal{O}_2 says as much or as little about terms in Σ as \mathcal{O}_1 . Conservative extensions and practical approximations have been recently used in DLs to define notions of modules that are now implemented and used in various ontology engineering tasks [10,3].

3 Topicality

Similar to, say, a book, an ontology can be about one or more things, from one or more angles, and it can be about various more or less independent topics. For example, “Zen and the Art of Motorcycle Maintenance” (by Robert M. Pirsig) can be said to be about two topics, philosophy and motorcycles, but it ties them rather closely together so that they may be viewed as dependent. In order to determine the topic of a book – or an ontology – we thus need to first agree on a suitable notion of *coherence* and its dual, *independence*.

In this paper, we are concerned with the following concepts:

- a *description*, a syntactical object, e.g., a text, a conversation, a thought, or an ontology. We assume that this object can be broken down into smaller pieces such as chapters, sentences, axioms, etc.
- a *vocabulary* and a *grammar* used to write well-formed sentences. In logics, the former is called signature and the latter determined by the syntax of the logic. We assume that the syntactical object conforms to our grammar and sticks to the signature.
- a notion of cognitive or logical *coherence*, which determines whether we take a given description as a coherent whole, i.e., one where the terms involved in it depend on each other, or whether it disaggregates into various pieces that can be said to be *independent* of each other. Clearly, coherence and independence are dual to each other, and there can be different notions of coherence.
- the *topic* or subject matter of the description, that is, a label describing the main concepts and their relationships within the description we are concerned with. We want to emphasize that the label describing a topic does not follow the same grammar rules as the elements of the description. For example, the topic addressed in the book by R. Pirsig can be described as “philosophical discussion during motorcycle riding”. This concept description does not conform to the given grammar as it misses a predicate. For ontologies, we may think of a topic as a suitable selection of terms from the signature, organized in structured expressions such as “animals in terms of their energy sources and reproduction”.

So, assume you want to automatically determine *the* topic of an ontology: this is clearly an underspecified task. Firstly, assume we have a high-quality ontology that we all agree is a coherent representation of the relevant concepts, regardless of how strict a notion of coherence we employ. Then, how do we represent its topic? Clearly, taking the set of all terms is unhelpful: it is probably too

verbose and detailed, and also fails to distinguish between the *main* concepts, e.g., animals, and auxiliary concepts, e.g., their energy sources. Thus representing a topic also involves a measure of importance and means for structuring. Secondly, both for informal descriptions and for ontologies, we are confronted with different notions of coherence of varying granularity and with different properties. For example, it depends on our notion of coherence whether “Zen and the Art of Motorcycle Maintenance” is about motorcycle maintenance and, rather unrelated to that, about philosophy, or whether Pirsig has related both topics together so that the book is about both. Hence we are, in this paper mostly concerned with logic-based notions of coherence and leave the problem of how to represent a topic in a useful way for future work.

In what follows, we briefly introduce five logic-based notions of coherence that will later be discussed in more detail.

Signature independence. The most coarse-grained notion of coherence is more easily defined in terms of independence: two descriptions are independent if they talk about different things, i.e., if they do not share any (non-logical) terms. Thus a description is coherent if it cannot be partitioned into (independent) fragments whose signatures are disjoint. In general, we would assume each book to be coherent in this sense, and for its title to be some representation of its content.

Δ -signature independence. Clearly, the above notion is so coarse-grained that most ontologies will be considered as one coherent whole, regardless of whether we could easily point out intuitively independent fragments. For example, a more detailed version of our ontology about animals and their energy sources might talk both about parts of animals involved in the metabolic process, e.g., their stomachs, and about parts of metabolic processes. If we use the same role, say `hasPart` to describe parthood, then anatomy and metabolism are a coherent whole w.r.t. to the first notion of coherence, whereas we might want to see them as two independent “subtopics”. Hence a second notion of coherence can be obtained from the first one by loosening the signature disjointness condition and allowing independent fragments to share terms from a special, shared part of the signature, e.g., `hasPart`.

Natural independence. Another notion of coherence can be identified by observing that, in our ontology, we may be talking about different *kinds* of things, e.g., about animals and (metabolic) processes, and that we describe the former in terms of the latter. This would allow us to decompose our ontology into a fragment about processes and one about animals, where the latter depends on the former. With this in mind, we can say that a fragment is coherent if it talks about one kind of things, and independent otherwise.

(Ir)relevance. Clearly, the above notion is still rather coarse-grained in that it cannot distinguish between, e.g., “organisms in terms of their metabolism” and “organisms in terms of their reproduction”, even though these topics might be largely independent: we can talk about the former without mentioning the latter, i.e., the latter does not come up naturally when talking about the former.

In this sense, we can say that a fragment is coherent if it completely covers a set of terms, e.g., `Organism` and `livesOf`. In contrast to the former notion, this now allows us to consider the same things, e.g., organisms, as being the subject of independent topics.

Minimal (ir)relevance. Now this last notion is a rather loose one: taking it literally, it would allow us to consider anything that “covers” a set of terms as a coherent whole, regardless of how loosely connected these terms are. It is hard to think of an ontology where the fragment of organisms and tax forms could be considered coherent: we would expect it to fall apart into two fragments. Thus a further condition leads us to our final notion of coherence: we say that a fragment is coherent if it completely covers a set of terms, e.g., `Animal` and `hasChild`, and is not simply the union of two or more such covering fragments.

Each of these notions of coherence allows us to determine *how many* topics an ontology is about. Some of them even give rise to a possibly interesting structure of topics, i.e., a topic can be a subtopic of another one.

4 Logic-Based Notions of Coherence

In what follows, we relate the five notions of coherence sketched above with the corresponding logical formalisations.

4.1 Signature Independence

The logical formalisation of this first notion of coherence was introduced by Parikh in 1999 in the context of Belief Revision [14]. The question addressed originally is essentially the following: if we want to revise a theory with a new piece of knowledge that contradicts some of what is entailed, do we have to check it against the whole theory? Or do we have some kind of safety that allows us not to touch those parts that are *independent* from this new finding? Hence Parikh was mainly concerned with independence, and thus his definition of coherence was only a byproduct. He formalises a way to split a logical theory \mathcal{T} into independent parts each of which is, in a maximally fine-grained split, coherent.

Definition 1. *Let \mathcal{O} be a logical theory over the signature $\tilde{\mathcal{O}}$ and let $\{\tilde{\mathcal{O}}_1, \tilde{\mathcal{O}}_2\}$ be a partition of $\tilde{\mathcal{O}}$. We say that $\tilde{\mathcal{O}}_1, \tilde{\mathcal{O}}_2$ split the theory \mathcal{O} if there are formulae α over $\tilde{\mathcal{O}}_1$ and β over $\tilde{\mathcal{O}}_2$ such that the logical closure of α and β coincides with the logical closure of \mathcal{O} . In this case, we say that $\{\tilde{\mathcal{O}}_1, \tilde{\mathcal{O}}_2\}$ is a \mathcal{O} -splitting. In general, we say that (mutually disjoint) signatures $\tilde{\mathcal{O}}_1, \dots, \tilde{\mathcal{O}}_n$ split \mathcal{O} if there exist formulae $\alpha_i \in \tilde{\mathcal{O}}_i$ for $i = 1, \dots, n$ such that \mathcal{O} is the logical closure of $\alpha_1, \dots, \alpha_n$.*

Please note that Parikh’s splitting is a *signature* splitting: it may be the case that \mathcal{O} is a single formula, and thus we cannot identify any coherent *subsets* of \mathcal{O} , even though we can identify independent subsets of its signature: please

note how, in first order logic, we can take any finite set of axioms and express it equivalently in a single axiom.

A nice property of \mathcal{O} -splittings is that they are unique.

Lemma 2. *Given a theory \mathcal{O} over the signature $\tilde{\mathcal{O}}$, there is a unique finest \mathcal{O} -splitting of $\tilde{\mathcal{O}}$, i.e. one which refines every other \mathcal{O} -splitting.*

As mentioned above, Parikh’s notion of coherence is rather loose: any ontology whose signature cannot be decomposed in *disjoint*, independent subsets is coherent, and would thus give rise to only a single topic.

4.2 Δ -Signature Independence

To obtain a more fine-grained view of topics of an ontology, we can loosen Parikh’s notion of independence, i.e., we can say that any (part of an) ontology is coherent if its signature *apart from some special, common terms* cannot be decomposed in disjoint, independent subsets. These special, common terms can be distinctive of the whole area, but can be used in different contexts. They can be considered as descriptive of patterns and methodologies of a theory, but not specific of a topic: roughly speaking, they belong to a “meta-topic”. In our introductory example, `hasPart` was such a special term.

In [12] this approach has been formalized by introducing Δ -decompositions. The authors assert that some terms, like `hasPart` in our example, behave as logical symbols under certain points of view. So, the idea is to identify the set Δ of these terms in order to decompose the ontology into “signature-but- Δ ”-disjoint sub-ontologies.

In contrast to Parikh’s approach, we now have an approach that depends on the choice of Δ .

Definition 3. *Let \mathcal{O} a finite theory of formulae in SO , $\Delta \subseteq \tilde{\mathcal{O}}$ and \mathcal{L} a fragment of SO . A partition $\Sigma_1, \dots, \Sigma_n$ of $\tilde{\mathcal{O}} \setminus \Delta$ is called a signature Δ -decomposition of \mathcal{O} in \mathcal{L} if there are $\mathcal{O}_1, \dots, \mathcal{O}_n$ theories of formulae in \mathcal{L} such that*

- $\tilde{\mathcal{O}}_i \subseteq \Sigma_i \cup \Delta$ for $i = 1, \dots, n$
- $\mathcal{O}_1 \cup \dots \cup \mathcal{O}_n \equiv \mathcal{O}$.

$\mathcal{O}_1, \dots, \mathcal{O}_n$ is called a realization of the signature Δ -decomposition $\Sigma_1, \dots, \Sigma_n$ in \mathcal{L} .

This definition involves second order logic for technical reasons, namely to ensure that the realization of a Δ -decomposition always exists. As for Parikh’s approach, there is a unique finest Δ -decomposition.

Theorem 4. *Let \mathcal{O} a finite theory of SO formulae, $\Delta \subseteq \tilde{\mathcal{O}}$, and let $\Sigma_1, \dots, \Sigma_n$ and Π_1, \dots, Π_m be Δ -decompositions of \mathcal{O} in SO . Then, the partition $\Sigma_i \cap \Pi_j$ for all i, j with $\Sigma_i \cap \Pi_j \neq \emptyset$ of $\tilde{\mathcal{O}} \setminus \Delta$ is a Δ -decomposition of \mathcal{O} in SO . Thus, there exists a unique finest Δ -decomposition of \mathcal{O} in SO .*

Also as in Parikh’s approach, we decompose a signature and not necessarily an ontology: given an ontology \mathcal{O} in a given DL \mathcal{L} , it can be the case that there exists a suitable, non-trivial Δ -decomposition of \mathcal{O} but we are not able to automatically compute corresponding subsets of \mathcal{O} in the given DL. That is, in contrast to second order logic, other logics like DLs do not allow the so-called *unique decomposition realization* (UDR).

Another challenge is the suitable selection of a set Δ . As the authors say, they “do not expect signature decompositions to be a push-button technique, but rather envision an iterative and interactive process of understanding and improving the structure of an ontology, where the designer repeatedly chooses sets Δ and analyzes the impact on the resulting decomposition.”

This remark leaves us with the open question: can we use this technique to extract the topics of an ontology? Since the decomposition depends on the selection of terms in Δ , the result obtained does not reflect an intrinsic logical coherence of topics – at least, we still do not have conditions to ensure such a property.

4.3 Natural Independence

A totally different approach to the partitioning of an ontology \mathcal{O} is carried out by identifying fragments that respect validity of axioms that they contain. In 1989 Garson [9] proposed that a *logical module* \mathcal{M} should be:

- *logically correct*, i.e. any axiom entailed by \mathcal{M} should be entailed by \mathcal{O}
- *logically complete*, i.e. any axiom over \mathcal{M} that is entailed by \mathcal{O} should be entailed by \mathcal{M}

The intuition is that a logical module should preserve all the entailments that involve the signature the logical module “deals with”. This means that different logical modules can share terms. In [5] the authors apply \mathcal{E} -connections to *decompose* ontologies, instead than to compose them as originally defined in [13]. The (computable) notion of module they are searching for is such that no subsumption relations exist between concepts (as in DLs, i.e. unary predicates) inside the module and concepts outside the module. This intuition leads to the following notion of module.

Definition 5. A *TBox* $\mathcal{M}_A \subseteq \mathcal{O}$ is a module for a concept $A \in \tilde{\mathcal{O}}$ if:

- \mathcal{M}_A is a logical module in \mathcal{O}
- for every concept $B \in \tilde{\mathcal{O}}$, the following holds:
 - (a) $\mathcal{M}_A \models \{A \sqsubseteq B\} \iff \mathcal{O} \models \{A \sqsubseteq B\}$
 - (b) $\mathcal{M}_A \models \{B \sqsubseteq A\} \iff \mathcal{O} \models \{B \sqsubseteq A\}$
- there are no concepts $C, D \in \tilde{\mathcal{O}}$ such that $C \in \widetilde{\mathcal{M}}_A$, $D \notin \widetilde{\mathcal{M}}_A$ and either $\mathcal{O} \models C \sqsubseteq D$ or $\mathcal{O} \models D \sqsubseteq C$.

To obtain such modules from an ontology, the authors describe a 3-steps algorithm: a safety-check, a partitioning algorithm, and the identification and extraction of modules.

The safety-check enforces a (mild) limitation in the ontologies that can be modularized. If an ontology is not safe, then this algorithm cannot be applied.

The partition algorithm, instead, aims at creating groups of concepts that can be interpreted independently from the other groups (see Theorem 3 in [5]). In particular, we obtain a partition of the *domain*, whose parts can be of three types: (Red) those which import vocabulary from others, (Blue) those whose vocabulary is imported, and (Green) isolated parts. Intuitively, this property means that either the parts correspond to actual non-overlapping subject matters, or the ontology is underspecified and some of the parts correspond to “unused information”. In both cases, this seems to reflect a logical structure of the ontology. Moreover, any of these partitions can be automatically labelled with the highest common concept name in the concept hierarchy.

The step that identifies and extracts the modules is needed to determine which partitions are modules, or if an aggregation of some of these is necessary to ensure that what we have is a logical module. However, this step is irrelevant for the purposes investigated in this paper. In Fig. 1 it is shown the partitioning of the toy ontology *Koala*³ that contains 42 axioms.

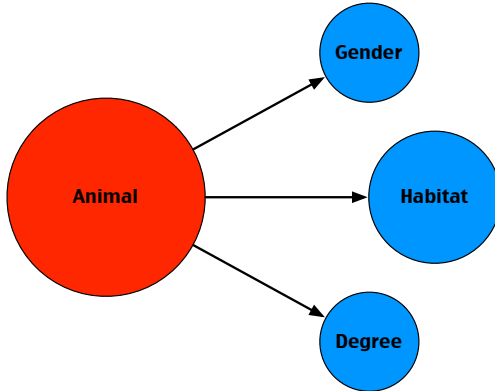


Fig. 1. \mathcal{E} -connections of the ontology *Koala*

Using the procedure described, when \mathcal{E} -connections succeeds, it generates modules that correspond to an intuitive partition of the ontology. Moreover, it reveals also a structure: parts that need to import other parts to be defined. In particular, distinct parts can share part of the vocabulary, but no instances: for example, we can have a part dealing with *Pets* and another with *PetOwners*. Unfortunately, this technique fails sometimes in partitioning an ontology, and it returns a unique block, even if the ontology seems in principle well structured; for example, this happens with the ontology *Periodic*⁴

³ <http://protege.stanford.edu/plugins/owl/owl-library/koala.owl>

⁴ www.cs.man.ac.uk/~stevensr/ontology/periodic_full_06012009.owl

4.4 (Ir)relevance

Another approach to the identification of topics within an ontology starts from a different point of view: the same individual can belong to different topics, as for example if we talk about a person, we can mean she in terms of her job, or she in terms of her family, etc. Intuitively, we think to these as different topics. In other words, a topic is identified not from a single label, but from the relationships between all the concepts it deals with.

In [2] the authors define *locality-based* modules: given an arbitrary set of terms Σ , such a module contains all axioms that “know everything” about Σ , where “know everything” depends on the notion of locality-based module used. These modules are an approximation of conservative extensions, as defined in Sect. 2: in other words, an ontology is always a conservative extension of any locality-based module. Among the strong logical properties that such modules satisfy, we have that they are logical modules as defined before. Moreover, such modules can be efficiently extracted. Hence, it would be interesting to investigate the family of all possible modules of an ontology \mathcal{O} .

By definition, locality-based modules can overlap. In principle this could lead to an exponential number (w.r.t. the size of the ontology) of modules; in [6] the trendline of the number of modules has been empirically studied for a selection of different ontologies, and the exponentiality of the family of modules seems to be confirmed by the experiments. Hence, studying the whole family of all locality-based modules is in general an infeasible task to be approached by brute force.

4.5 Minimal (Ir)relevance

In [7] we introduced a new approach to represent the whole family $\mathfrak{F}_{\mathcal{O}}$ of locality-based modules of an ontology \mathcal{O} . The key point is observing that some axioms appear in a module only if other axioms do. In this spirit, we defined a notion of “logical dependence” between axioms: the idea is that an axiom α depends on another axiom β if whenever α occurs in a module \mathcal{M} then β also belongs to \mathcal{M} .

Then, by using some notions of algebra, we have identified clumps of highly inter-related axioms, called *atoms*, defined to be maximal disjoint subsets of ontologies such that their axioms either appear always together in modules, or none of them does. In other words, atoms never split over two or more modules. Therefore, we can slightly extend the definition of “logical dependency” to atoms as in the following definition.

Definition 6. *Let \mathbf{a} and \mathbf{b} be two distinct atoms of an ontology \mathcal{O} . Then:*

- \mathbf{a} is dependent on \mathbf{b} (written $\mathbf{a} \succeq \mathbf{b}$) if, for every module $\mathcal{M} \subseteq \mathcal{O}$ containing \mathbf{a} , we have $\mathbf{b} \subseteq \mathcal{M}$.
- \mathbf{a} and \mathbf{b} are independent if there exist two disjoint modules $\mathcal{M}_1, \mathcal{M}_2$ of \mathcal{O} such that $\mathbf{a} \subseteq \mathcal{M}_1$ and $\mathbf{b} \subseteq \mathcal{M}_2$.
- \mathbf{a} and \mathbf{b} are weakly dependent if, they are neither independent, nor dependent; in such case, there exists an atom \mathbf{c} which both \mathbf{a} and \mathbf{b} are dependent on.

Without loss of generality, we can remove from the ontology *syntactic tautologies*, i.e. always-local axioms, and *global axioms*, i.e. axioms that belong to all modules. We can always remove these unwanted axioms and consider them separately. As a consequence, the empty set is a module of the ontology. More importantly, the set of atoms is a partitioning of the ontology, hence linear w.r.t. its size.

The computation of the AD is polynomial w.r.t. the size of the ontology (provided that the extraction of a module is polynomial), and the algorithm to obtain the AD given an ontology and a (suitable) notion of module is discussed in [7]. The first (and fundamental) step can be described as follows: for each axiom α of the ontology, the algorithm takes as input its signature $\tilde{\alpha}$, and returns the module $\top \perp^* \text{-mod}(\tilde{\alpha}, \mathcal{O})$. These modules are non empty, since we already removed syntactic tautologies, and consequently at least α is non-local w.r.t. $\tilde{\alpha}$. By definition, then, this module contains the atom \mathbf{a} that α belongs to. Moreover, the strong logical properties of locality-based modules imply that this module is the smallest that contains \mathbf{a} . As a consequence, over the set $\mathcal{A}(\mathfrak{F}_{\mathcal{O}})$ of atoms, called *Atomic Decomposition* (AD), is induced a structure of partially ordered set. Hence the AD of an ontology can be represented by means of a Hasse diagram.

Beside covering all atoms, the modules we obtained following this procedure do not fall apart into two modules, and are also called *genuine*. As a consequence, they form a basis of the family of modules $\mathfrak{F}_{\mathcal{O}}$.

Definition 7. *A module is called fake if there exist two uncomparable (w.r.t. set inclusion) modules $\mathcal{M}_1, \mathcal{M}_2$ with $\mathcal{M}_1 \cup \mathcal{M}_2 = \mathcal{M}$; a module is called genuine if it is not fake.*

It is clear that there is a 1-1 correspondence between atoms and genuine modules. In particular, given an atom \mathbf{a} the corresponding genuine modules can be retrieved by considering all atoms \mathbf{a} is dependent on.

Definition 8. *The principal ideal of an atom \mathbf{a} is the set $(\mathbf{a}] = \{\alpha \in \mathfrak{b} \mid \mathfrak{b} \preceq \mathbf{a}\} \subseteq \mathcal{O}$.*

We summarize in the following table the ways described so far to look at ontologies’ fragments.

Structure	\mathcal{O}	$\mathfrak{F}_{\mathcal{O}}$	$\mathcal{A}(\mathfrak{F}_{\mathcal{O}})$
Elements	axioms α	modules \mathcal{M}	atoms $\mathbf{a}, \mathfrak{b}, \dots$
Maximal size	baseline	exponential	linear
Mathem. object	set	family of sets	poset

The decomposition of an ontology in atoms seems to capture a very fine-grained notion of coherence; however, in order to determine the topic of an atom we still need a way to identify a suitable label. Obviously, such suitable label will be chosen within the signature of the principal ideal of the atom itself. One possible way consists of labelling each atom \mathbf{a} with the vocabulary used in its axioms, and then, to express the “logical dependency”, of recursively removing

the terms already used in some atom that a is dependent on. In this way, each atom is labelled only with the “new terms” introduced. Notice that in this case some atoms can have empty labels. In Fig. 2 is represented the Hasse diagram for the AD of the ontology Koala, whose labels are picked as described. Please note that the heights of the nodes vary: the heigher, the more numerous their axioms are.

The intuition suggests that the “core” atoms of an ontology can be recognized by looking at this structure, because many other atoms depend on them. However, we still need to carry out real experiments to validate how cognitively significant this way of representing an ontology and of labelling it is.

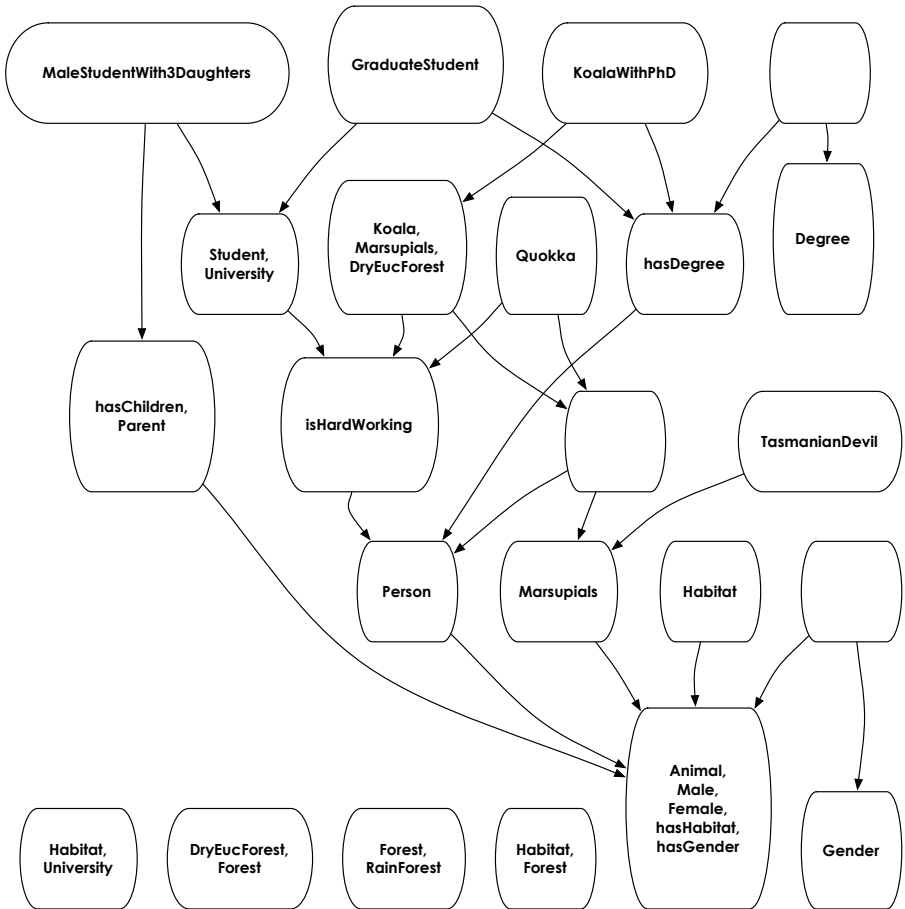


Fig. 2. Locality-based decomposition of part of the ontology Koala

5 Discussion and Outlook

The more an ontology reflects faithfully the knowledge about a domain, the more flexible and reusable it can be. However, ontologies are complex systems, since the tasks that ontology engineers have to perform are *per se* complex, and include the design, the implementation and the maintenance of ontologies. The design deals with the understanding of the domain, the implementation is a translation of the knowledge about the domain into a suitable ontology language, and the maintenance consists of updating the knowledge about the domain, as well as testing the consistency between the intended modelling of the domain and the actual modelling of the ontology.

Hence, the complexity of ontology engineering demands for the development of tools and methodologies to support engineers in manipulating ontologies. A crucial task is to support ontology engineers in the comprehension of ontologies. In particular, we have addressed the problem of topicality in ontologies, that is, “what the ontology is about”, by means of modularity. However, to support comprehension of an ontology, we need to identify not only coherent fragments of the ontology but also their meaning.

The first two approaches we described here define coherence as “sharing terminology”, and this, as we have seen, lead to a coarse decomposition of the ontology, the second being a refinement of the first. In principle, this can provide support in the understanding of the big picture of the ontology. Whilst the signature independence is straightforward to determine, the computation of Δ -independence can involve rewriting part of the ontology: complexity results for different languages are also carried out in [12]. However, both techniques are still merely theoretical, that is, there is no implementation, and we do not have any evaluation of their cognitive usefulness: these approaches can display a weakness in applicability if ontology engineers need only tools to support a fine tuning of the ontology. Moreover, since these methods do not involve any extraction of topic, the ontology engineers still have to look at all axioms of a partition. Finally, we recall that the selection of terms to be included in the set Δ is not automatic or guided: for any selection of terms, the engineer has to evaluate if the decomposition obtained matches with her understanding of the domain.

The approach based on \mathcal{E} -connections aims at identifying fragments of the ontology dealing with different kinds of individuals. The method has been implemented and it is available on the web⁵. In [5] the authors prove that the modularisation algorithm is polynomial in the size of the ontology. Moreover, when the algorithm succeeds in partitioning the ontology, the result corresponds to the intuition of users, and since it provides also labels, it helps in the understanding the structure of the ontology. Ontology engineers use this tool in real applications. However, it does not capture the finer-grained notion of topic, that occurs when we want to focus on individuals in terms of a specific aspects. And this is the intuitive reason for returning just one part – the ontology itself – for highly interrelated ontologies.

⁵ <http://www.mindswap.org/2004/SW00P>

The extraction of a locality-based module is a well-understood and starting to be deployed in standard ontology development environments, such as Protégé 4.6 and online.7 It is used, for example, in the field for ontology reuse [11]: the modules extracted are quite small, and capture the knowledge of the ontology about the signature provided for the extraction. In [6] the authors tried to extract all such modules from ontologies in order to get insight into the modular structure of the ontology, but for medium size ontologies the algorithm, although highly optimised, did not succeed in the task. However, such modules capture the notion of topicality described as relationships between terms.

The atomic decomposition takes advantage of the nice logical properties of locality-based modules and separate parts of the ontology that show a minimal irrelevance. In [8] the authors prove that computing the atomic decomposition of an ontology is polynomial in the size of the ontology, provided that the extraction of a module is polynomial. Although methods for automatically labelling the atoms are future work, from a first evaluation it seems that this kind of decomposition can be used by ontology engineers in fine tuning ontologies. Moreover, this structure suggests a different notion of relevance from just counting the number of axioms of a part: an atom is more relevant if it is needed by many atoms. However, for big ontologies the decomposition looks too fragmented. This problem can be solved by using one of the previous methods, even if other formal techniques to group together some atoms into meaningful but coarser parts are included in our future work.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
2. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. *J. of Artif. Intell. Research* 31, 273–318 (2008)
3. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Extracting modules from ontologies: A logic-based approach. In: Stuckenschmidt, H., Parent, C., Spaccapietra, S. (eds.) *Modular Ontologies*. LNCS, vol. 5445, pp. 159–186. Springer, Heidelberg (2009)
4. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: The next step for owl. *J. of Web Sem.* 6(4), 309–322 (2008)
5. Cuenca Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Modularity and web ontologies. In: *Proc. of KR-2006*, pp. 198–209. AAAI Press, Menlo Park (2006)
6. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: an empirical study. In: *Proc. of DL 2010*. CEUR-WS.org, vol. 573 (2010)
7. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: Atomic decomposition. In: *Proc. of IJCAI 2011* (accepted 2011)
8. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: atomic decomposition. Tech. rep., The University of Manchester (2011), <http://bit.ly/i4o1Y0>

⁶ <http://www.co-ode.org/downloads/protege-x>

⁷ <http://owl.cs.manchester.ac.uk/modularity>

9. Garson, J.: Modularity and relevant logic. *Notre Dame Journal of Formal Logic* 30(2), 207–223 (1989)
10. Ghilardi, S., Lutz, C., Wolter, F.: Did I damage my ontology? A case for conservative extensions in description logics. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) *Proc. of KR 2006*, pp. 187–197. AAAI Press, Menlo Park (2006)
11. Jimeno, A., Jiménez-Ruiz, E., Berlanga, R., Rebolz-Schuhmann, D.: Use of shared lexical resources for efficient ontological engineering. In: *SWAT4LS 2008*. ceir-ws.org, vol. 435 (2008)
12. Konev, B., Lutz, C., Ponomaryov, D., Wolter, F.: Decomposing description logic ontologies. In: *Proc. of KR 2010*, pp. 236–246 (2010)
13. Kutz, O., Lutz, C., Wolter, F., Zakharyashev, M.: \mathcal{E} -connections of abstract description systems. *Artificial Intelligence* 156(1), 1–73 (2004)
14. Parikh, R.: Beliefs, belief revision, and splitting languages. In: Moss, L.S., de Rijke, J.G., M. (eds.) *Logic, language and computation*, vol. 2, pp. 266–278. Center for the Study of Language and Information, USA (1999)
15. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) *ISWC 2004*. LNCS, vol. 3298, pp. 289–303. Springer, Heidelberg (2004)

A Concept Discovery Approach for Fighting Human Trafficking and Forced Prostitution

Jonas Poelmans¹, Paul Elzinga³, Guido Dedene^{1,4}, Stijn Viaene^{1,2},
and Sergei O. Kuznetsov⁵

¹ K.U. Leuven, Faculty of Business and Economics, Naamsestraat 69,
3000 Leuven, Belgium

² Vlerick Leuven Gent Management School, Vlamingenstraat 83,
3000 Leuven, Belgium

³ Amsterdam-Amstelland Police, James Wattstraat 84,
1000 CG Amsterdam, The Netherlands

⁴ Universiteit van Amsterdam Business School, Roetersstraat 11
1018 WB Amsterdam, The Netherlands

⁵ National Research University Higher School of Economics (HSE), Pokrovskiy blv. 11
101000 Moscow, Russia
{Jonas.Poelmans, Stijn.Viaene, Guido.Dedene}@econ.kuleuven.be
skuznetsov@hse.ru
Paul.Elzinga@amsterdam.politie.nl

Abstract. Since the fall of the Iron curtain starting in 1989 in Hungary, millions of Central and Eastern European girls and women have been forced to work in the European sex industry (estimated 175,000 to 200,000 yearly¹). In this paper, we present our work with the Amsterdam-Amstelland (Netherlands) police to find suspects and victims of human trafficking and forced prostitution. 266,157 suspicious activity reports were filed by police officers between 2005 and 2009 that contain their observations made during a police patrol, motor vehicle inspection, etc. We used FCA to filter out interesting persons for further investigation and used the temporal variant of FCA to create a visual profile of these persons, their evolution over time and their social environment. We exposed multiple cases of forced prostitution where sufficient indications were available to obtain the permission from the Public Prosecutor to use special investigation techniques. This resulted in a confirmation of their involvement in human trafficking and forced prostitution resulting in actual arrestments being made.

1 Introduction

Irina, aged 18, responded to an advertisement in a Kiev, Ukraine newspaper for a training course in Berlin in 1996. With a fake passport, she traveled to Berlin, Germany where she was told that the school had closed. She was sent on to Brussels, Belgium for a job. When she arrived she was told she needed to repay a debt of

¹ Eerste rapportage Nationaal Rapporteur Mensenhandel
[http://www.bnrm.nl/Images/Rapportage%201%20\(Ned\)_2002_tcm63-83113.pdf](http://www.bnrm.nl/Images/Rapportage%201%20(Ned)_2002_tcm63-83113.pdf)

US\$10000 and would have to earn the money in prostitution. Her passport was confiscated, and she was threatened, beaten and raped. When she didn't earn enough money, she was sold to a Belgian pimp who operated in Rue D'Aarschot in the Brussels red light district. When she managed to escape through the assistance of police, she was arrested because she had no legal documentation. A medical exam verified the abuse she had suffered, such as cigarette burns all over her body (Hughes et al. 2003).

The above story is a typical example of a woman of Eastern Europe who was forced into the European sex industry. Rough estimates suggest that 700,000 to 2 million women and girls are trafficked across international borders every year (O'Neill 1999, U.S. Department 2008). The majority of transnational victims are trafficked into commercial sexual exploitation. Human trafficking is the fastest growing criminal industry in the world, with the total annual revenue for trafficking in persons estimated to be between \$5 billion and \$9 billion (United Nations 2004). The council of Europe states that "people trafficking has reached epidemic proportions over the past decade, with a global annual market of about \$42.5 billion" (Equality division 2006). The most popular destinations for trafficked women are countries where prostitution is legal such as the Netherlands (Hughes 2001). According to Shelley et al. (1999) most of these women are in conditions of slavery. Girls of Dutch nationality who were forced to work in prostitution in Amsterdam typically fell prey to a loverboy. The loverboy is a relatively new phenomenon (Bovenkerk et al. 2004) in the Netherlands. A loverboy is a man, mostly with Moroccan, Antillean or Turkish roots who makes a girl fall in love with him and then uses her emotional dependency to force her to work as a prostitute.

In this paper we report on our Formal Concept Analysis (FCA)-based (Ganter et al. 1999) efforts for identifying unknown suspects and victims of human trafficking and forced prostitution in the police region Amsterdam-Amstelland in the Netherlands. Since the introduction of Intelligence Led Policing (Collier 2006, Viaene et al. 2009) in 2005, a management paradigm for police organizations which aims at gathering and using information to allow for pro-active identification of suspects, police officers are required to write down everything suspicious they noticed during motor vehicle inspections, police patrols, etc. These observational reports, 34,817 in 2005, 40,703 in 2006, 53,583 in 2007, 69,470 in 2008 and 67,584 in 2009, may contain indications that can help reveal individuals who are involved in human trafficking, forced prostitution, terrorist activities, etc. However, till date almost no analyses were performed on these documents.

We first used concept lattices to visualize the observational reports and distill interesting indicators and concepts that can be used for tracking down suspects. For each person mentioned in these reports, a document vector was constructed containing all relevant attributes or indicators that were found in the data. This concept lattice in which all available information for each person was gathered, revealed some cases where there were sufficient indications for starting an in-depth investigation. We applied FCA and its temporal variant to zoom in on some real life cases and suspects, resulting in actual arrestments being made and/or illegal prostitution locations closed down.

In section 2 we give background information on human trafficking, forced prostitution and the guidelines that were developed by the Attorney Generals of the

Netherlands to help detect trafficking and loverboy suspects. In section 3 we describe the dataset. In section 4 we describe our analysis method to detect and profile potential suspects. In section 5 we describe some real life cases where the suspects were found with FCA. Finally, section 6 concludes the paper.

2 Human Trafficking and Forced Prostitution

Victims of human trafficking rarely make an official statement to the police. The human trafficking team of the Amsterdam-Amstelland police is installed to proactively search police databases for any signals of human trafficking. Unfortunately, this turns out to be a laborious task. The investigators have to manually read and analyze the police reports, one by one, because only an estimated 15% of the information containing human trafficking indications has been labeled as such by police officers. As soon as the investigators find sufficient indications against a person, a document based on section 273f of the code of criminal law is composed for the person under scrutiny. Based on this report, a request is sent to the Public Prosecutor to start an in-depth investigation against the potential suspects. After permission is received from the Public Prosecutor, the use of special investigation techniques such as phone taps and observation teams is allowed.

The following list contains the types of indications mentioned in the guidelines developed by the Attorney Generals of the Netherlands based on which police forces can gather evidence of human trafficking and forced prostitution against potential suspects. These guidelines define in which cases pro-active intervention by police may be necessary. This information had not yet been used to actively search police databases for suspicious activity reports containing human trafficking indicators.

1. Dependency on exploiter: Typically in human trafficking the housing, clothing and transportation of the woman are arranged through the exploiter, the woman will often have debts towards the exploiter and will be forced to earn the money back.
2. Deprivation of liberty: Often the victim is not allowed to have contact with the outside world. She typically does not have her passport with her which is carried by the pimps.
3. Being forced to work under bad circumstances: The victim has to work for many hours, cannot freely dispose of the money she earns, etc.
4. Violation of bodily integrity of the victim: The victim is forced to work as a prostitute through physical violence, threatening, etc.
5. Non-incident pattern of abuse by suspect(s) can be observed.

3 Dataset

Our dataset consists of 266,157 suspicious activity police reports, 34,817 in 2005, 40,703 in 2006, 53,583 in 2007, 69,470 in 2008 and 67,584 in 2009. These police reports are stored in the police databases as unstructured text documents and have the following associated structured data fields: title of the incident, project code assigned by the responsible officer, location of the incident and optionally a formally labeled suspect, victim and/or other involved persons. The unstructured part of these

suspicious activity reports describes observations made by police officers during motor vehicle inspections, during a police patrol, when a known person was seen at a certain place, etc. These reports were extracted from the database and turned into html documents that were indexed using the open source engine Lucene.

The thesaurus constructed for this research contains the terms and phrases used to detect the presence or absence of indicators in these police reports. This thesaurus consists of two levels: the individual search terms and the term cluster level which was used to create the lattices in this paper. We used a semi-automated approach as described in (Poelmans et al. 2010a). Search terms and term clusters were defined in collaboration with experts of the anti-human trafficking team and gradually improved by validating their effectiveness on subsets of the available police reports. Each of these search terms were thoroughly analyzed for being sufficiently specific. The quality of the term clusters was determined based on their completeness. The validation of the quality of the thesaurus and the improvements were done by us and in conjunction with members of the anti-human trafficking team. Concept structures were created on multiple randomly selected subsets of the data. It was manually verified if all relevant indicators were found in these reports and no indicators were falsely attributed to these reports. For example, the term cluster “prostitute” in the end contained more than 20 different terms such as “prostitutee”, “dames van lichte zeden”, “prosti”, “geisha”, etc. used by officers to describe a prostitute in their textual reports. To create the formal contexts in this paper, the term clusters in the thesaurus were used as attributes and the police reports as objects. A prototype of the FCA-based toolset CORDIET (which is currently being developed under a collaboration between KULeuven and Moscow Higher School of Economics) was used during the analysis process (Poelmans et al. 2010d).

4 Method

Our investigation procedure consists of multiple iterations through the square of Fig. 1. For background information on FCA and its applications in KDD we refer the reader to Poelmans et al. (2010c). The guidelines of section 2 contain a non-limitative list of indications and the indications can be subdivided into 5 main categories. If at least one of the thesaurus elements corresponding to these indications is present for a person or a group of persons, we might be dealing with a case of human trafficking or forced prostitution. From the 266,157 reports in our dataset, the relevant reports which contain at least one indicator are selected. Then, the persons mentioned in these reports are extracted and FCA lattices are created, showing all the indications observed for each person. From these lattices containing persons, potential suspects or victims can be distilled and they can be further analyzed in detail with FCA and temporal concept lattices. If sufficient indications are available, a document based on article 273f of the code of criminal law can be created and sent to the Public Prosecutor with the request for using advanced intelligence gathering instruments such as observation teams, phone taps, etc. If the suspects are indeed involved in human trafficking and forced prostitution they can be taken into custody.

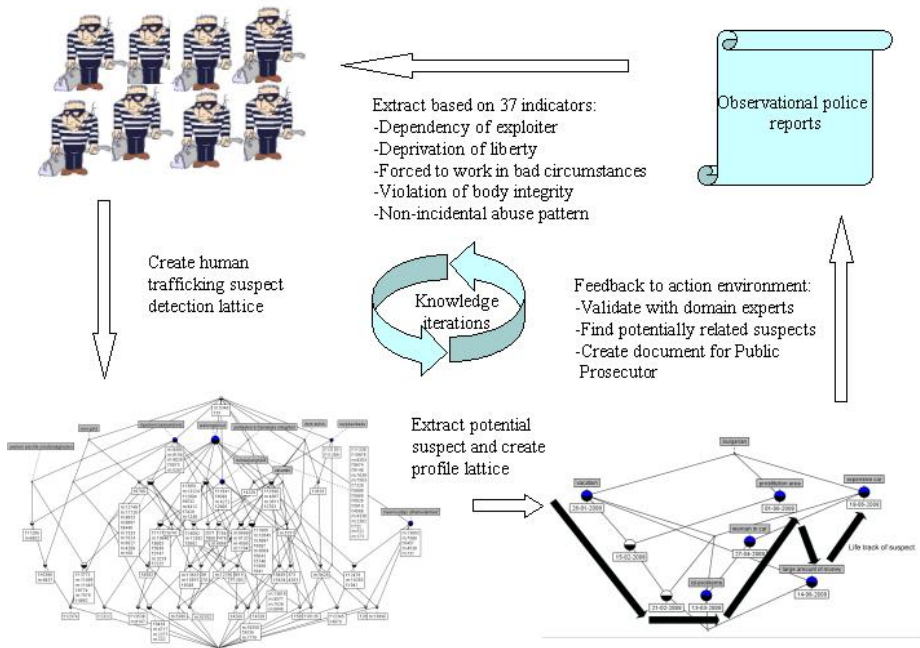


Fig. 1. Criminal intelligence process

Our method based on FCA consists of 4 main types of analysis that are performed:

- **Concept exploration of the forced prostitution problem of Amsterdam:** In (Poelmans et al. 2010a, Poelmans et al. 2010b) our FCA-based approach for automatically detecting domestic violence in unstructured text police reports is described in detail. We not only improved the domestic violence definition but also found multiple niche cases, confusing situations, faulty case labelings, etc. that were used to amongst others improve police training. Part of the research reported on in this paper such as the construction of the thesaurus, consisted of repeating the procedures described in our domestic violence case study papers.
- **Identifying potential suspects:** Concept lattices allow for the detection of potentially interesting links between independent observations made by different police officers. When grouping suspicious activity reports on a per person basis, the available information about the individuals is displayed in one intuitive and understandable picture that facilitates efficient decision making on where to look. In particular persons lower in the lattice can be of interest since they combine multiple early warning indicators.
- **Visual suspect profiling:** Some FCA-based methods such as Temporal Concept Analysis (Wolff 2005) were developed to visually represent and analyze data with a temporal dimension. Temporal Concept lattices were used in (Elzinga et al. 2010) to create visual profiles of potentially interesting terrorism subjects. Scharfe et al. (2009) used a model of branching time in

which there are alternative plans for the future corresponding to any possible choice of a person and used it as the basis of an ICT toolset for supporting autism diagnosed teenagers. For creating the temporal profile of individual suspects, we use traditional FCA lattices and the timestamps of the police reports on which these lattices are based are used as object names. The nodes of the concept lattice can then be ordered chronologically.

- **Social structure exploration:** Concept lattices may help expose interesting persons related to each other, criminal networks, the role of certain suspects in these networks, etc. With police officers we discussed and compared various FCA-based visualization methods of criminal networks. Individual police reports mentioning network activity were used by us as objects and the timestamps of these police reports together with each suspect name mentioned in these reports as object names.

5 Analysis and Results

Traditional data mining techniques often focus on automating the knowledge discovery process as much as possible. Since the detection of actual suspects in large amounts of unstructured text police reports is still a process in which the human expert should play a central role, we did not want to replace him, but rather empower him in his knowledge discovery task. We were looking for a semi-automated approach and in this section we try to illustrate the main reasons why FCA was ideal for this type of police work. With FCA at the core, we were able to offer police officers an approach which they could use to interactively explore and gain insight into the data to find cases of interest to them on which they could zoom in or out. Section 5.1 shows a lattice diagram which was of significant interest to investigators of the anti-human trafficking team. For the first time, the overload of observational reports was transformed into a visual artifact that showed them a set of 1255 persons potentially of interest to the police and the indicators observed for each of them. The lattice diagram visually summarizes the data and makes it more easily accessible for officers who want to efficiently explore it and extract unknown suspects. We chose to first highlight the case of the Turkish human trafficking network in section 5.2. From the lattice diagram in section 5.1, two potential suspects were distilled since they were regularly spotted performing illegal activities. We found the name of a bar was mentioned a couple of times and used this information to build the concept lattice of section 5.2. This lattice diagram was of particular interest to police officers since FCA quickly gave them a concise overview of the persons that were observed to be involved around a suspicious location and the lattice structure helped them to identify the most important suspects in this network. In particular the visualization of persons in a lattice was helpful during their exploration. FCA's partial ordering gave them clues on where to look first. The lower a person appears in the lattice, the more indicators he has. Section 5.3 showcases how the FCA visualization was used to combine temporal and social structure information in one easy to interpret picture. Such profile lattices were of significant interest to police officers since they allow for quick decision making on whether or not a person might be involved in illegal activities. Moreover, the lattices may help infer the roles of the persons mentioned in

the network. Finally section 5.4 shows how an FCA lattice can give insight into the evolution of a person over time, in this case of a loverboy. The remaining part of this section describes cases of human trafficking and forced prostitution and two of them were identified in the lattice in Fig. 2 and further investigated with FCA. Note that real names were replaced by false names because of privacy reasons.

5.1 Detection of Suspects of Human Trafficking and Forced Prostitution

Multiple concept lattices were created for detecting human trafficking suspects in the set of persons. Each of these concept lattices contained over 200 concepts and were based on different combinations of attributes. Since the format of this paper does not allow to visualize the entire lattices in a readable way, we chose to simplify one of these lattices and zoomed in on its most important aspects. Fig.2. contains the lattice diagram with 1255 Bulgarian, Hungarian and Romanian persons. The concept containing some of the suspects of section 5.2 was found on the right and bottom part of the lattice and has 10 persons in its extent. The concept containing the main suspect

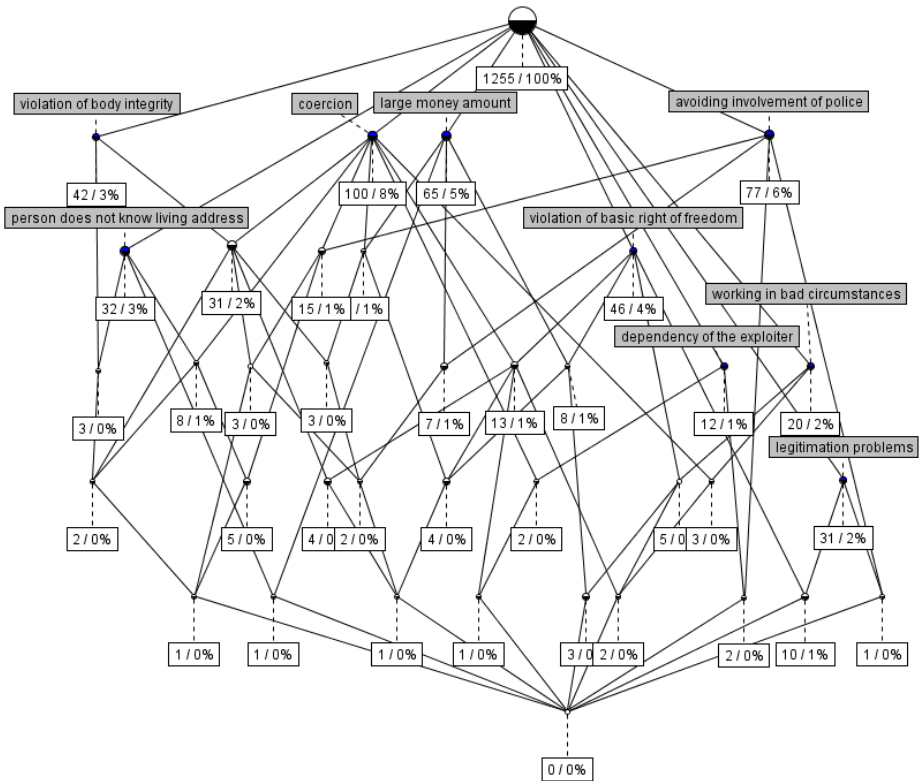


Fig. 2. Human trafficking suspect detection lattice diagram

of section 5.3 was found on the left and bottom part of the lattice and has 1 object in its extent. The following 2 sections will be used to describe and profile each of these suspects in detail.

5.2 Case 1: Turkish Human Trafficking Network

By analyzing the concept lattice based on observational reports, we were able to expose a criminal network operating in Amsterdam, involved in illegal and forced prostitution. The concept lattice diagram in fig. 3 contains the 61 persons and indicators found in the police reports mentioning activity around a bar in Amsterdam that played a central role in the network's activities and was closed down in 2009. Multiple suspects operating in this network were found and some of the observations will be described in this section. The most important suspects are the persons with indication legitimation problems, since they were carrying the id papers of the girls. The police reports contained many indications of illegal and forced prostitution taking place, activities that were run by the owners or acquaintances of the owners of the bar. We found out the bar was used as a central hub, where mostly Turkish men met up with Bulgarian girls who had been forced into prostitution and took them to another location. We found at least two pimps who have multiple girls working for them.

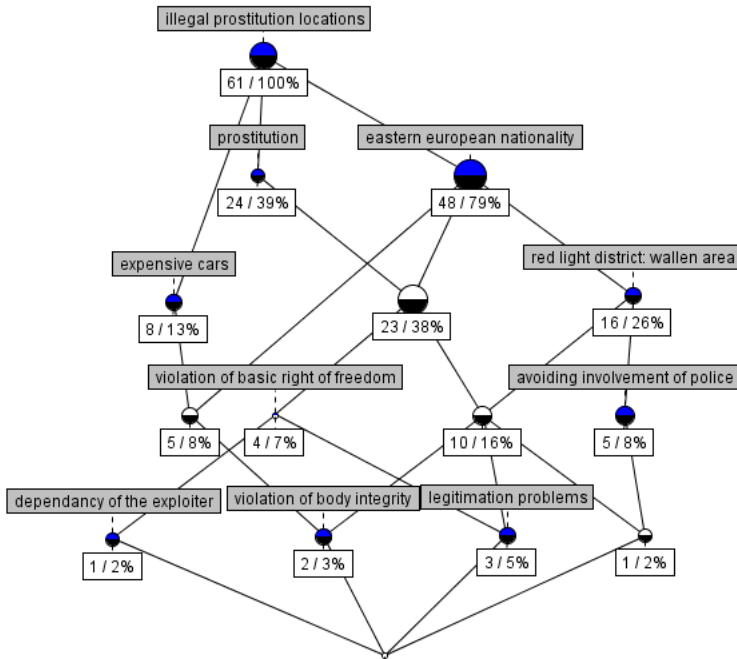


Fig. 3. Concept lattice diagram of human trafficking network

Starting in 2007, the first observations were made that hinted at illegal and forced prostitution being organized from within this bar. On 2 June 2008, victim H declared to the police that she was forced to work as a prostitute in the bar and did not get any money for that. She was never allowed to leave the house alone and the door of her apartment was locked from the outside such that she couldn't leave. On 12 December 2008, suspect A came out of the bar with a girl, their statements to the police did not match and moreover the girl was dressed in sexy clothing. Most likely the girl works as a prostitute and the driver is her pimp. On 25 January 2009, police officers stopped a car and behind the wheel was suspect B and next to him the victim E. We found woman E is often sitting at the bar and also the car is regularly parked in front of the bar. Suspect B gave the passport of victim E to the police and afterwards he placed it back in his pocket. Moreover, suspect B was carrying a large amount of cash money, 1000 euros in his pocket. On 26 January 2009, police did a check-up on the guests in the bar. One girl was new and told she only just arrived by train, she had no train tickets with her and she did not know her living address. Suspect B was also there and told the police he is a car trader so he travels a lot between Bulgaria and Netherlands. An excuse typically used by criminals responsible for the logistics of a trafficking network. Also victim E and two other girls, victims F and G were there. On 20 February 2009, police officers saw suspect A talking to the driver of a car with Bulgarian license plate. Afterwards he forced a girl to follow him and when the police asked about their relationship they told they had been friends for 3 months. The girl did not have her id-papers with her and the police went to her living address. In the house there were many mattresses and another girl. Both of them told they have no job. Most likely the house serves as an illegal prostitution location for the criminal gang.

Sufficient indications were found and on 17 June 2009, an observation team observed the bar during the evening. Eastern European women were sitting at the bar and mostly Turkish, Moroccan and Eastern European men at the tables. During the evening, the team saw multiple girls that were taken out of the bar by a customer to a hotel, house, etc. and brought back to the bar afterwards. On 15 July 2009 sufficient evidence was gathered that illegal prostitution was organized from within this bar and authorities closed down the bar.

5.3 Case 2: Bulgarian Male Suspect

In this section we describe a profile of a Bulgarian suspect who was also operating in Amsterdam. The lattice diagram in Fig. 4 shows that on 3 October 2007, suspect A was observed for the first time during a police patrol. An officer told the driver of a BMW car with Bulgarian license plate to turn right instead of left, the driver however ignored the instructions he received and quickly drove to the left with squeaking tires. The officer went after and in the end stopped the car. There were 3 men and one woman in the car. Suspect B was the driver and suspect A was sitting next to him. On the backseat of the car were woman F and man K. They told the officer they only arrived 3 days ago in the Netherlands and are a couple. Suspect A and suspect B were taken to the police office; man K and woman F walked away and were followed by a second officer. He saw that K was strongly holding the hand of F and forced her into

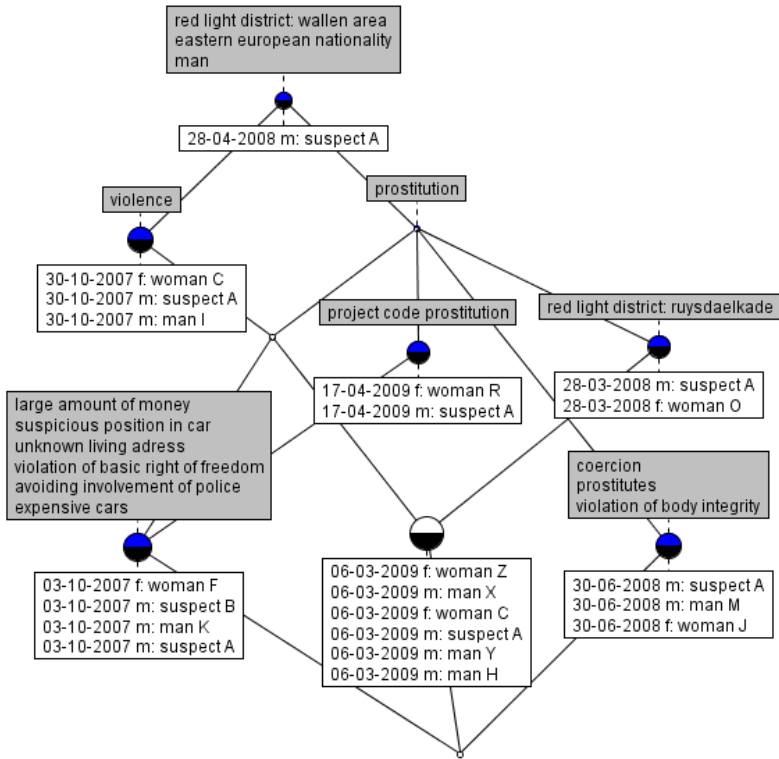


Fig. 4. Profile lattice diagram of individual suspect and his network

a home at the corner of a street in central Amsterdam. In the police office, suspect B was not able to tell the address of the apartment he was going to rent. Suspect A was carrying a large amount of cash money in his pocket.

On 30 June 2009, woman J went to the police to ask if they could supervise the undersigning of a tenancy agreement of an apartment by man M who promised her accommodation. She told suspect A was intimidating and trying to scare away man M because suspect A wanted to rent the apartment for prostitution purposes. She was very afraid of suspect A and the officer noted that she might have been forced in prostitution by him. On 30 October 2007, the police did a routine inspection of 2 individuals who were waiting with two motorcycles in a street that had been plagued by street robberies. This was the second observation of suspect A by the police and his motorcycle was registered by the name of woman C who had been involved in human trafficking activities as a victim. On 6 March 2009 the police received a tip that a fugitive Colombian criminal might be living at a certain address owned by professional criminal H. When they entered the apartment they found 2 men and 2 women of Bulgarian nationality. Man X and woman C declared to be on holiday and would go back to Bulgaria although we found suspect A was driving around with a scooter registered at C's name in 2007. Man Y declared he exports expensive cars to Bulgaria and regularly drives back and forth between Netherlands, an excuse typically

used by suspects taking care of logistics of a human trafficking gang. Woman Z declared to work in prostitution in Groningen. When the officers left the apartment they found a motorcycle registered on the name of suspect A. The last observation dates back to 17 April 2009 when the police saw suspect A call somebody while standing in the entrance hall of prostitute R. He tells the police he has nothing to do with prostitution and owns a restaurant in Bulgaria. After his phone call he gives the cell phone to the prostitute.

To conclude, suspect A and B are most likely involved in human trafficking and there were sufficient signals found to request the use of special investigation techniques. Permission was granted, our suspicions were confirmed and both A and B were arrested by the police in 2010. Moreover these lattices showed some other people who are involved in the same gang and could be monitored.

5.4 Case 3: Loverboy Suspect

In this section we describe a loverboy case which we exposed by gathering evidence from multiple observational reports. This person was not found by analyzing the lattice diagram in Fig. 2 but by investigating a lattice based on Antillean, Moroccan and Turkish persons. Victim V is a girl of Dutch nationality who officially lived in the Netherlands but fell prey to a loverboy of originally Antillean nationality. We found multiple indications in filed suspicious activity reports that referred to elements of the model in section 2. The lattice diagram of suspect A and victim V is displayed in Fig. 5.

On 27-04-2006, Suspect A and victim V were noticed for the first time on the streets during a police patrol. They had a serious argument with each other and suspect A took the cell phone with force out of V's hand. When the police intervened they claimed nothing happened. In the police station she declared that she works voluntarily in prostitution although her words were not convincing to the officer. On 15-08-2006 an Amsterdam citizen sent an email to the police about young Antillean men who constantly surveillance some women in the red light district. Amongst other suspect A brings food and drinks to the women who are not allowed to leave their rooms. On 31-10-2006 during a police patrol, victim V was noticed while she got out of a car and quickly ran inside. The driver of the car was suspect A. She told the police later on that she was brought to and picked up every day at this apartment by her boyfriend suspect A. The police noticed her dismayed and timid attitude and asked again if she was forced to work in prostitution. In a non-convincing way she responded that she did her job voluntarily. On 15-09-2006, suspect A had to stay in jail for 6 hours because of illegal weapon possession. When the police asked about his income he told he earned good money thanks to his girlfriend who works in prostitution. On 2-11-2006, officers noticed the car of victim V was parked on the road and two Negroid men were inside. The driver, suspect A got out of the car and yelled to the girl he was picking up at her apartment, that she had to hurry up. The whole scene looked very intimidating to the police and it turned out the girl was victim V. Suspicious was that the car was registered on the name of V while V had no driver license. On 28-03-2007, victim B came to the police office to ask if she was allowed to work with a badly damaged id-document or if she had to wait for a new one. She mentioned that suspect A was her ex-boyfriend and that she and victim V were the victim of extortion but she did not dare to make an official statement to the

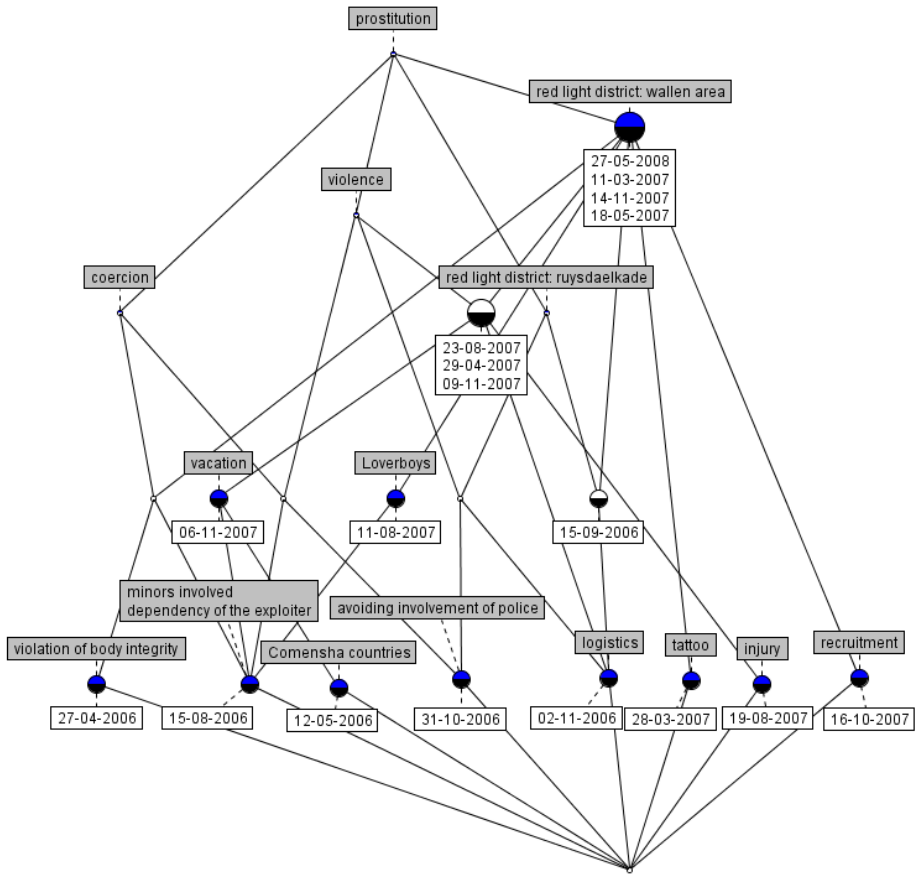


Fig. 5. Profile diagram of loverboy suspect

police. Afterwards, the police checked a home where they found 2 women: victim V and B. Victim V had a big tattoo on her right shoulder and a smaller tattoo on her upper arm. On 19-08-2007, suspect A was involved in a knifing incident in the red light district between 3 men and one of these men got seriously injured. This man wanted sex with victim V but suspect A did not allow this because of the man's ethnicity, which caused the fight. On the camera surveillance videos, victim V was observed to accompany suspect A all the time. On 16-10-2007, officers observed that suspect A who walked over the streets said hi to all women who passed by.

6 Conclusions

Textual documents contain a lot of useful information that is rarely turned into actionable knowledge by the organizations that own these data repositories. The police of Amsterdam-Amstelland disposes of a large amount of such textual reports

that may contain early warning indicators that can help to proactively identify persons involved in illegal activities. Since the observations of one suspect are typically made by different officers who are not aware of each others work, spread over multiple databases, etc. automated analysis techniques such as FCA can be of significant importance for police forces who are interested in the proactive identification of perpetrators. FCA is one of the few techniques that can be used to interactively expose, investigate and refine the underlying concepts and relationships between them in a large amount of data. In this paper we described our successful application of FCA to find suspects of human trafficking and forced prostitution in the Amsterdam-Amstelland police district. From 266,157 observational reports we distilled multiple suspicious cases of which 3 have been described in this paper. For each of these persons and networks we composed a document containing all the indicators and evidence available and sent this to the Public Prosecutor. Permission to use special investigation techniques was obtained by the anti-human trafficking team based on these documents. For each case we exposed, phone-taps, observation teams, etc. indeed confirmed the suspect's involvement in human trafficking and forced prostitution. We believe that in making the shift from reactive police work, where action is only undertaken when a victim comes to talk directly to the police, to the pro-active identification of suspect's, FCA can play an important role.

Acknowledgements. The authors would like to thank the police of Amsterdam-Amstelland for granting them the liberty to conduct and publish this research. In particular, we are most grateful to Deputy Police Chief Reinder Doleman and Police Chief Hans Schönfeld for their continued support. Jonas Poelmans is aspirant of the Fonds Voor Wetenschappelijk Onderzoek – Vlaanderen or Research Foundation – Flanders.

References

1. Bovenkerk, F., Van San, M., Boone, M., Van Solinge, T.B., Korf, D.J.: Loverboys of modern pooierschap in Amsterdam. Willem Pompe Instituut voor Strafwetenschappen, Utrecht (December 2004)
2. Collier, P.M.: Policing and the intelligent application of knowledge. *Public money & management* 26(2), 109–116 (2006)
3. Elzinga, P., Poelmans, J., Viaene, S., Dedene, G., Morsing, S.: Terrorist threat assessment with Formal Concept Analysis. In: *Proc. IEEE International Conf. on Intelligence and Security Informatics*, Vancouver, Canada, May 23-26, pp. 77–82 (2010)
4. Equality Division, Directorate General of Human Rights of the Council of Europe, Action against trafficking in human beings: prevention, protection and prosecution. In: *Proceedings of the Regional Seminar, Bucharest, Romania (April 4-5, 2006)*
5. Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer, Heidelberg (1999)
6. Hughes, D.M.: The “Natasha” Trade: The transnational shadow market of trafficking in women. *Journal of international affairs*, 53(2) (Spring 2000)
7. Hughes, D.M., Denisova, T.: *Trafficking in women from Ukraine U.S. Department of Justice research report (2003)*

8. O'Neill, R.A.: International trafficking to the United States: a contemporary manifestation of slavery and organized crime- and intelligence monograph. Exceptional Intelligence Analyst Program, Washington, DC (1999)
9. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Curbing domestic violence: Instantiating C-K theory with Formal Concept Analysis and Emergent Self Organizing Maps. *Intelligent Systems in Accounting, Finance and Management* 17, 167–191 (2010), doi:10.1002/isaf.319
10. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formally Analyzing the Concepts of Domestic Violence. *Expert Systems with Applications* 38, 3116–3130 (2010), doi:10.1016/j.eswa
11. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formal Concept Analysis in Knowledge Discovery: a Survey. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 139–153. Springer, Heidelberg (2010)
12. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Concept Discovery Innovations in Law Enforcement: a Perspective. In: IEEE Computational Intelligence in Networks and Systems Workshop (INCos 2010), Thessaloniki, Greece (2010)
13. Shelley, L.: Human trafficking: defining the problem. *Organized crime watch-Russia* 1(2) (February 1999)
14. Scharfe, H., Oehrstrom, P., Gyori, M.: A Conceptual Analysis of Difficult Situations – developing systems for teenagers with ASD. In: Suppl. Proc. Of the 17th Int. Conf. On Conceptual Structures (ICCS), Moscow, Russia (2009)
15. U.S. Department of State (2008) Trafficking in persons report <http://www.state.gov/g/tip/rls/tiprpt/2008> (retrieved on 26-12-2010)
16. United Nations, Economic and social council, Economic causes of trafficking in women in the Unece region. Regional Preparatory Meeting, 10-year review of implementation of the Beijing Platform for Action (December 14-15, 2004)
17. Viaene, S., De Hertogh, S., Lutin, L., Maandag, A., den Hengst, S., Doleman, R.: Intelligence-led policing at the Amsterdam-Amstelland police department: operationalized business intelligence with an enterprise ambition. *Intelligent systems in accounting, finance and management* 16(4), 279–292 (2009)
18. Wolff, K.E.: States, transitions and life tracks in Temporal Concept Analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 127–148. Springer, Heidelberg (2005)

A Modeling Method and Declarative Language for Temporal Reasoning Based on Fluid Qualities

Matei Popovici, Mihnea Muraru, Alexandru Agache, Cristian Giumale,
Lorina Negreanu, and Ciprian Dobre

POLITEHNICA University of Bucharest,
Splaiul Independentei nr. 313, sector 6, Bucuresti,
Postal Code: 060042
{pdmatei,mmihnea,alexandruag,crislian.giumale,
lorina.negreanu,cipsmm}@gmail.com

Abstract. Current knowledge representation mechanisms focus more on providing a static description of a modeled universe and less on capturing evolution. Ontology modeling languages, such as OWL, have no inherent means for describing time or time-dependent properties. In such settings, time is usually represented along with other application-dependent concepts, yielding complex models that are difficult to maintain, extend, and reason about. On the other hand, in imperative languages that allow the definition of time-dependent behavior and interactions such as WS-BPEL, the emphasis is on specifying the control flow in a service-oriented environment. In contrast, we argue that a declarative approach is more suitable. We propose a modeling method and a declarative language, designed for representing and reasoning about time-dependent properties. The method is applicable in areas such as ubiquitous computing, allowing the specification of intelligent device behaviour.

Keywords: temporal representation, temporal reasoning, hypergraph, declarative language.

1 Introduction

Traditionally, ontologies provide static descriptions of a given domain of interest. That is the case of medical approaches such as [8] or semantic lexicons such as WordNet [10]. Existing ontology modeling methods have no inherent means for representing evolution. Also, languages such as the Ontology Web Language (OWL) [5] lack the same feature: they have no dedicated constructs that accommodate time or temporal properties. The reasoning process is focused on taking a snapshot of the modeled universe that includes concepts, relations and individuals, and derive additional properties such as concept subsumption and satisfiability [5].

In many cases temporal concepts can be defined on top of existing modeling primitives, such as in OWL-Time [12], and thus provide a high-level modeling

layer. Time-related concepts reside on the same representational level with other application-specific concepts. Creating such models proves difficult, lacks scalability, and makes the reasoning process computationally hard. When modeling real-life behavior, ontologies are challenged to accommodate temporal evolution. For example, concepts such as *married two times* or *widower* are inherently dependent on properties that held in the past, but are no longer valid presently. Dedicated mechanisms for representing and reasoning about time-dependent knowledge are required in this case.

We propose an ontology-based modeling method for time-dependent applications that accomplishes these goals. Our approach takes a static hierarchy of concepts and relations between concepts (defined in a manner similar to the one used in OWL), and extends it with a structure able to represent time and change. Temporal elements are no longer represented on the application layer; they become modeling primitives. Consequently, the method provides *native* means for representing temporality. Unlike conventional ontologies, where the instantiation relationship between an individual and a concept is unique, our approach allows for multiple instantiations with respect to the same concept-individual pair, at different moments of time. This allows individuals to be enrolled in concepts such as *single, but married two times in the past*, by simply exploring past *marriage* instantiations with respect to the same individual. All time-dependent instances are stored in a dedicated structure designed to preserve temporal order. We call such a structure a *hypergraph*. Temporal reasoning is accomplished by exploring the hypergraph.

The paper is structured as follows: Section 2 describes our modeling approach based on individuals, qualities, actions, as well as the structure responsible for storing the ordering of concept instances: the hypergraph. Section 3 introduces a language that allows both static and time-dependent modeling. Some similarities and differences with respect to other declarative languages, such as CLIPS [11] and Prolog [7], are discussed. In Section 4, a case study for intelligent device behavior in an ubiquitous environment [3] is described. Section 5 compares our method with other approaches. Section 6 presents conclusions, as well as future work.

2 A Modeling Method Based on Fluid Qualities

2.1 Modeling Primitives

Individuals. Concepts, together with concept instances, are useful for defining a static universe of discourse. As an alternative to such traditional representations, our framework relies on a modeling approach based on individuals, fluid qualities and actions, introduced in [4]. Individuals are atomic entities, identifiable by themselves. They are *perennial*: during the evolution of a model, individuals do not disappear or suffer structural changes. Individuals can however acquire or lose qualities.

Qualities. A (fluid) quality $Q(i_1, \dots, i_n)$ or $Q(\vec{i})$ represents a time-dependent n -ary relationship between individuals $\vec{i} = (i_1, \dots, i_n)$. A unary quality $Q(i)$

stands for a time-dependent property associated with individual i . Qualities hold on specific time-intervals Δt . Δt denotes the time-slice associated with $Q(\bar{i})$. Qualities are created by instantiating a quality prototype $Q(\bar{x})$. Here, \bar{x} represents variables. We say that a quality was destroyed if its time-slice ended at a certain moment in time. Once introduced in a model, a quality is never completely erased. Qualities have similarities to property instances from OWL [5]. One major difference is that qualities are time-dependent and thus need not be unique. As a result, if an individual i was enrolled in Q in the past, then $Q(i)$ has a time-slice Δt associated with an interval from the past. If i is enrolled in Q once more, then a new quality $Q(i)$, with a different time-slice, will be introduced. The time-slices of the two quality instances cannot overlap. For example, *Married(John)* can be present several times in a model that describes *John's* marriages. In a similar way, qualities such as *On(AirConditioner)* can be present at different moments, in a model for home devices. Quality prototypes are equivalent to concepts and properties from conventional ontologies. They allow the instantiation of qualities, with respect to certain individuals. Quality prototypes can form hierarchies. We use the meta-relationship *is-a* to build the hierarchy of quality prototypes. If $Q_1(\bar{x})$ *is-a* $Q_2(\bar{y})$, then all individuals enrolled in a quality Q_1 over a specific time slice Δt must also be enrolled in Q_2 , as well as in all other qualities enforced by Q_2 , through the chain of its existing *is-a* meta-relations. For example, assume the following quality prototype definition: *AirConditioner(x) is-a Device(x)*. In this case, if an individual possesses the quality *AirConditioner*, then it is automatically enrolled in the quality *Device*.

Actions. An action $a(\bar{i})$ represents an external stimulus that changes the state of the modeled universe. Actions can enrol one or several individuals, and can require the presence or absence of particular qualities. When actions occur, they produce side-effects: the creation and/or destruction of qualities. From this point of view, actions can be considered constructors and destructors for qualities. With respect to time, actions are instantaneous: they occur at a particular moment and have no duration. For instance, in a model describing marital evolution, where the qualities $q_j = \textit{Single}(\textit{John})$ and $q_a = \textit{Single}(\textit{Anne})$ hold, executing the action *marries(John, Anne)* will terminate the qualities q_j and q_a and create a new quality *married(John, Anne)*.

2.2 Representing Time

The hypergraph. The evolution of a model is encoded in a structure $H = (A, T, E_q, E_a)$, where H is an oriented, acyclic graph. A is the set of action instances, T is the set of temporal nodes, E_q is the set of qualities and E_a is the set of preconditions.

Time is represented using actions and qualities. A moment in time is defined as a set of actions $t_a = \{a_1, \dots, a_k\}$ that occur simultaneously. $t_a \in T$ is a temporal node in a hypergraph. Action nodes are depicted in white in Fig. [1]. Temporal nodes are shown in grey, and contain action nodes. There is a distinguished temporal hypernode, *Init*, that refers to the starting moment of

the modeled application. It contains an implicit action denoted by a_{init} , which is the constructor for all initial qualities in the model. Similarly, the *Current* hyper-node denotes the current moment in the unfolding of the model evolution. It also has an implicit $a_{current}$ action, that is considered to be a pseudo-destroyer of all existent qualities.

Time intervals are defined by pairs of actions (not necessarily consecutive). For example assume an action $a_1(\bar{x})$ is executed. As a result, a certain quality $Q(\bar{x})$ is introduced. Assume also that another action $a_2(\bar{x}')$ destroys $Q(\bar{x})$. Then, if t_1 is a temporal node such that $a_1 \in t_1$ and similarly $a_2 \in t_2$, then the time-slice of $Q(\bar{x})$ is $\Delta t = [t_1, t_2]$. This duration is represented as an edge between action nodes a_1 and a_2 . If a_2 happens to be $a_{current}$, it means that the quality holds at the present moment. Notice that different qualities might have identical durations, as a result of being created (and destroyed) by actions that occur simultaneously. Also, it might be that multiple qualities are created/destroyed by the same action.

Temporal nodes need not have specific values. In some applications, time is relative and the focus is on event ordering only. In these situations, temporal nodes are symbolic. In cases where measurements of time are important, temporal nodes can be assigned actual timestamps. Depending on the desired precision, timestamps can encode minutes, seconds, milliseconds etc.

The E_q set contains edges that stand for n -ary qualities $Q(\bar{x})$. These edges span action nodes $a_{start}, a_{end} \in A$ that construct and destroy $Q(\bar{x})$, respectively. Edges from E_a are shown as solid arrows, in Fig. 1. The E_a set contains directed edges that designate preconditions of action nodes from A . An edge $e = (q, a)$ from quality q to action a designates q as a satisfied precondition of a . e actually connects a quality edge to an action node. Precondition edges are shown as dotted arrows in Fig. 1.

Initially, H contains only *Init* and *Current*, and all predefined qualities span action nodes a_{init} and $a_{current}$. H changes, as new stimuli are recorded by the model. If an action (or a set of actions) is executed, and the required preconditions, according to the action's prototype, hold, then: (1) a new temporal node is created, and inserted in H , just before *Current*; (2) the temporal node is

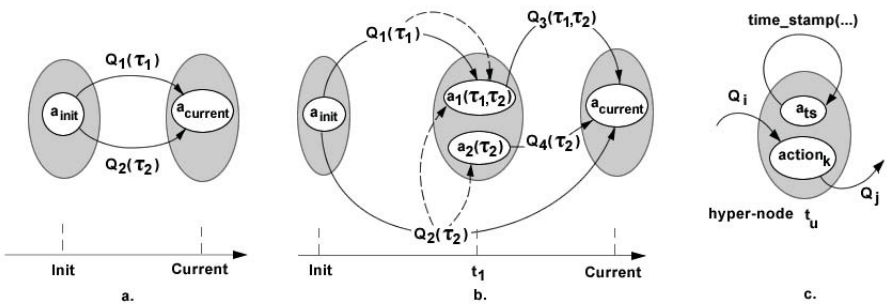


Fig. 1. The hypergraph

populated with all valid actions that are executed; (3) for each satisfied precondition, a dedicated edge between a required quality and the current action is added to E_a ; (4a) if an action a destroys a quality, then its ending action node is modified from $a_{current}$ to a ; (4b) if an action creates a new quality q , then a new edge $(a, a_{current})$ corresponding to q , is added to E_q .

As an example, consider the hypergraph from Fig. II(a). It contains two qualities $q_1 = Q_1(\tau_1)$ and $q_2 = Q_2(\tau_2)$ that hold at the current moment of time. In this scenario, actions $a_1(\tau_1, \tau_2)$ and $a_2(\tau_2)$ are signalled. $a_1(\tau_1, \tau_2)$ requires the presence of q_1 and q_2 , and $a_2(\tau_2)$ requires q_2 . These dependencies are shown with dotted edges. Since all preconditions are satisfied, the action is executed. The effects can be seen in Fig. II(b). a_1 will terminate q_1 , and create a new quality $q_3 = Q_3(\tau_1, \tau_2)$. a_2 does not terminate any quality, but creates $q_4(\tau_2)$.

Temporal primitives. In the above examples, actions are solely conditioned by the existence of qualities at the current moment of time. In this context, the hypergraph performs as a structured log for recording events. In the following, we introduce temporal primitives as a mechanism for creating complex, temporal-based constraints on action execution. A temporal primitive enrolls two qualities, and enforces a certain temporal relation between them. In our model we use the Region Connection Calculus (RCC) relations to express temporal primitives. They are fully described in [9]. We shall not review the entire set of temporal primitives and their associated RCC relations. Instead, we will focus on some relations such as the ones shown in Table II. These relations, as well as their inverses will be used in the following sections. For example, a possible precondition associated to an action $a(\bar{x})$ is the existence, somewhere in the past, of qualities $q_1 = Q_1(\bar{x}')$ and $q_2 = Q_2(\bar{x}'')$ such that the constraint: q_1 *just_after* q_2 is satisfied. A hypergraph exploration will attempt to find instances q_1 and q_2 that satisfy the *just_after* constraint. If such instances are found, action a is executed, along with the entire process of edge and node insertions described previously.

Table 1. Examples of temporal primitives

Temporal Primitive	Associated RCC relation
q_1 <i>after</i> q_2	X DC Y (X and Y are disconnected)
q_1 <i>just_after</i> q_2	X PO Y (X and Y are partially overlapping)

3 A Declarative Language Based on Fluid Qualities

3.1 Motivation

Assume an application that models devices in a intelligent house scenario. We consider a simple model with two devices A and B . The device A can be turned on if both devices A and B are off. The device B can be turned on if: (1) it is off and (2) the device A has been off for a time period of at least T seconds.

In conventional programming or ontology-based languages, the program or ontology encoding the above scenario must contain specific data structures and handling mechanisms, for keeping track of time intervals associated to events. Even if these data-structures and functions are implemented as libraries (in the case of programming languages), or as time-related concepts (in the case of ontology languages) the resulting program or ontology will be larger, more cumbersome to write, modify and understand. In most existing approaches, a model for the above scenario should encode a potentially large finite-state automaton that contains states for all possible ways of turning on the two devices. Alternatively, a declarative language enhanced with temporal primitives would explicitly convey the semantic content of the model. In the following, we use the modeling method described in Section 2 in order to introduce a declarative language for time-dependent applications.

3.2 Defining Quality Prototypes

Assume the definitions from Program 1. Variables are designated using the symbol “?”, in a manner similar to that in CLIPS [11].

```
individual ac, p
quality Device(?d), HasPower(?d,?val), AirConditioner(?d) is-a Device(?d)
```

Program 1. Individuals and qualities

The simple model from Program 1 introduces two simple 1-ary quality prototypes: `Device`, and `AirConditioner`, and a binary quality prototype (or relation): `HasPower`.

3.3 Rules

Rules are the basis for defining actions, their preconditions as well as effects. A simple rule, such as the one from Program 2, can be read in the following way: *In order to execute action turnOn, the individual ?x must be a device, and also off. If these preconditions are satisfied and the action is signalled from the external environment, then the quality Off will be destroyed, and ?x will acquire the quality On.* Program 2(a) has similarities with rule definitions from declarative languages such as CLIPS. A possible translation of Program 2(a) in CLIPS is shown in Program 2(b).

Notice that CLIPS facts have been used to model qualities. Facts are also a means for representing actions. This appears to correspond to our modeling perspective: actions are external stimuli that, in temporal contexts where their preconditions hold, produce certain effects. Nevertheless, this assumption may cause problems, due to the instantaneous nature of actions. The following question is raised: *When is an action fact retracted in CLIPS, during the rule firing cycle?*

<pre>rule start_device preconditions: Device(?x), Off(?x) as ?off action: turnOn(?x) effects: destroy ?off, On(?x)</pre> <p style="text-align: center;">(a)</p>	<pre>(defrule start_device (device ?x) ?off <- (off ?x) (turnOn ?x) => (retract ?off) (assert (on ?x)))</pre> <p style="text-align: center;">(b)</p>
---	--

Program 2. A simple turnOn rule

Take for instance the CLIPS rule `start_device` from Program 3, where the more general fact `(turnOnAll)` replaces the particular action fact `(turnOn ?x)`. We assume that `(turnOnAll)` is externally introduced, in order to trigger the turning on of all devices. Notice that the retraction of `(turnOnAll)` is essential. In its absence, a rule such as `invalid` would be incorrectly executed, although its preconditions did not hold at the particular moment when `(turnOnAll)` was signalled. This happens because (1) the effects of the rule `multiple_start_device`, more precisely the assertion of `(on ?x)`, validate the preconditions of `invalid` and (2) the action fact `(turnOnAll)` is not removed, as it should.

<pre>(defrule multiple_start_device (device ?x) ?off <- (off ?x) ?all <- (turnOnAll) => (retract ?off) (retract ?all) (assert (on ?x)))</pre>	<pre>(defrule invalid (on ?x) (turnOnAll) => ...)</pre>
--	--

Program 3. Rule activation

However, retracting `(turnOnAll)` causes other problems. In CLIPS, an activation record for the rule `multiple_start_device` is created for a particular device in state off and selected in a nondeterministic fashion, from the set of stopped devices. When the rule is fired for that activation record, the fact `(turnOnAll)` is removed, thus inhibiting the turning on of other devices in state off. Since activation records are created in a sequential manner, and since, for each record, preconditions are checked, the solution is to mark the devices for which the rule is applicable, but defer carrying on the effects until after `(turnOnAll)` has been retracted. This way, newly created qualities will not be able to erroneously trigger the execution of other rules, such as `invalid`, having unsatisfied preconditions at the expected moment. The instantaneous nature of actions can only be modeled by translating single action facts that affect multiple entities, to multiple action facts associated with single entities.

In Program 4, (`turnOnAll`) is translated to multiple, particular action facts (`turnOn ?x`). The salience declaration gives this rule a higher priority. When all particular action facts have been generated, (`turnOnAll`) is removed by the rule `remove_turnOnAll`. By appending to Program 4 the rule `start_device` from Program 2(b), the effects of turning on all devices are added to the working memory and the desired behavior is finally obtained.

```
(defrule translate                                (defrule remove_turnOnAll
  (declare (salience 10))                        (declare (salience 1))
  (device ?x)                                    ?all <- (turnOnAll)
  ?off <- (off ?x)                                =>
  ?all <- (turnOnAll)                             (retract ?all))
  =>
  (assert (turnOn ?x)))
```

Program 4. Modeling simultaneous actions

It is easy to see that, in CLIPS, actions are difficult to define. In contrast, our approach makes a clear distinction between facts (or qualities) and actions. The latter are equivalent to signals that remain active throughout the rule execution cycle. As a result, the execution of one rule instance can affect other instances, by means of quality changes only. If translated in our language, Program 3 would execute for each device, as (`turnOnAll`) is an action and its removal would be handled by the interpreter, at the end of the rule execution cycle.

Returning to Program 2(b) notice that, in the absence of (`turnOn ?x`), the CLIPS rule would be executed for each entity that is a device, and that is in state *Off*. As a result, all devices in state *Off* would be turned on. Our approach is essentially different: qualities (and actions) are generated by actions only. It implicitly means that there must be at least an initial action, i.e. an entry-point in the program, that starts the model interpretation. As a consequence, the rule `start_device` from Program 2(a), would not be executed each time preconditions hold, but only when a *turnOn* action is signalled.

3.4 Representing and Computing Values

Numeric values. Numeric values are represented using predefined individuals. Predefined qualities can be assigned to such individuals, in order to disambiguate their numeric types. `Double(1)`, `Integer(1)`, `Float(1)` are such qualities. Since the modeling method does not explicitly define a mechanism for evaluating expressions, a special behavior is defined for numeric individuals. More precisely, their symbolic/textual representation encodes their actual value. For example, in `Double(1)`, 1 refers to a numeric individual that possesses the double value 1.0. Operators are defined using predefined actions. For instance, the arithmetic addition is an action that can be applied on, and compute, numeric individuals. The expression `?x = 2 + 3.5` is a shorthand for the execution

of the special action `+(2,3.5)` that produces, as a side-effect, the binding of variable `?x` to individual 5.5. The action `+` is responsible for type casts. Since both `Float(2)` and `Integer(2)` hold for 2, `+` will correctly evaluate the above expression. The hierarchy of types is internally defined using the `is-a` relationship as in `Integer(?x) is-a Number(?x)`.

Time and durations. The modeling approach described in Section 2.2 assumes that moments of time, modeled as hypernodes, are symbolic. In many applications, moments of time require actual values, thus allowing the application to compute durations, and make decisions based on them. As a result, the predefined qualities `Day`, `Month`, `Year`, `Time`, `Date` are introduced. The individuals enrolled in these predefined qualities have specific identifiers. In some cases, these individuals can be *polymorphic*. For instance, `Year(2010)` and `Number(2010)` enroll the same individual, 2010, that acts as both a year and a number. In these cases, qualities act as type casts that disambiguate the usage of an individual.

We introduce another predefined quality, `Timestamp`. It is associated automatically, by the model interpreter, with every hypernode from the hypergraph. To preserve uniformity, we consider that `Timestamp` is an edge having the same action as constructor and destructor. Its time span is zero, as seen in Fig. 1(c). As a result, timestamps can be used in preconditions, like any other quality. For instance, in `Temperature(?t) just_after Timestamp(08:00/1.1.2011)`, `?t` would hold the temperature value recorded in the hypergraph just after 08:00, on the 1st of January. Also, primitives for selecting the timestamp of nodes are defined. These can be applied on end-points of quality edges. `current_moment()` returns the timestamp associated with the *Current* node in the hypergraph. Also, `quality_start_moment(q)` and `quality_end_moment(q)` returns the timestamps associated with a quality edge end-point.

In order to compute timestamps, an alternative inspired from Haskell and from the unification process in Prolog [7] is used. Assume that, in $Q(\bar{x})$, \bar{x} refers to an enumeration of individuals and variables. An example is $Q(i_1, ?x, ?y, i_2, i_3)$. Also, if S is a substitution, i.e. a set of bindings of variables to individuals, we denote by $Q(\bar{x})/S$ the replacement of variables from $Q(\bar{x})$ with their respective individuals, according to S . The expression $Q(\bar{x})$ is $Q(\bar{y})$ produces the *unification* of the two qualities and, as a side-effect, the necessary variable bindings. For example, the unification from Program 5(a) solves the problem *what timestamp ?t/?d.?m.?y incremented by 2:00/2.0.0 yields 1:00/1.1.2011?*. Therefore, $S = \{?t = 23 : 00, ?d = 29, ?m = 12, ?y = 2010\}$ and `Timestamp(?t + 2:00, ?d + 2.?m.?y)` is `Timestamp(23:00/29.12.2010)`. There are cases when unifications can fail, as in Program 5(b). This happens because no such values can be assigned to `?t`, `?d`, `?m`. Also, there are cases such as the one shown in Program 5(c), where the unification leaves unbound variables.

4 Case Study: A Model for Intelligent Buildings

In the following, we illustrate a model for intelligent device behavior, described using the declarative language introduced in Section 3. In order to be

- (a) `Timestamp(1:00/1.1.2011)` is `Timestamp(?t + 2:00, ?d + 2.?m.?y)`
- (b) `Timestamp(1:00/1.1.2011)` is `Timestamp(?t + 2:00, ?d + 2.?m.2011)`
- (c) `Timestamp(?x/?y.?z.2011)` is `Timestamp(?t + 2:00, ?d + 2.?m.2011)`

Program 5. Timestamp unification

operational, the model relies on the following assumption: devices can be controlled individually using an uniform interface: web services. In this setting, actions are mapped on service invocations. For instance, in order to turn on device A, the invocation `AService.turnOn()` must be performed. In Program 6 we assume the existence of an individual `ac` and three qualities: `AirConditioner(ac)`, `CurrentTemp(20deg)` and `DesiredTemp(10deg)`.

```

rule get_current_temp
  preconditions: CurrentTemp(?x) as ?crt
  actions: newTemp(?y)
  effects: destroy ?crt, CurrentTemp(?y)

rule modify_temp
  preconditions:
    DesiredTemp(?x), ?x != ?y,
    AirConditioner(ac), Off(ac)
  actions: newTemp(?y)
  effects: turnOn(ac), setTemp(?x,ac)

rule stop_cooler
  preconditions:
    DesiredTemp(?x), ?x == ?y,
    AirConditioner(ac), On(ac)
  actions: newTemp(?y)
  effects: turnOff(ac)

```

Program 6. A model for air conditioners

We have omitted from Program 6 basic quality and action definitions such as `On`, `Off` and `turnOn`, `turnOff`, respectively. They were introduced previously in Section 3.3. The model from Program 6 connects a temperature sensor with an air conditioner. Whenever the sensor detects a temperature change, it generates an action `newTemp(?y)`. If the environment's temperature is different from the desired one, the air conditioner starts, and cools or heats with the specified temperature. Notice the presence of `Off(ac)` as a precondition in the rule `modify_temp`. This prevents starting the air conditioner, if it is already on. Rule `stop_cooler` stops the device, once the desired temperature was reached.

As seen in Program 6, `newTemp` is defined in several, possibly overlapping contexts. While `stop_cooler` and `modify_temp` have mutually exclusive contexts, the rule `get_current_temp` and either of `stop_cooler` or `modify_temp` do not. As a result, the rule `get_current_temp` can be fired at the same time with `modify_temp`. In this particular case, the overlapping can be avoided by replacing rules `modify_temp` and `get_current_temp` with a more complex one. This would also eliminate the necessity of a `CurrentTemp` quality, but would make the model harder to read and to extend.

Program 6 is able to describe intelligent device behavior, but contains no temporal constraints. In order to further restrict the behavior of devices, the constraints from Program 7 may be used. Here, we assume that an external action introduces the qualities `CanicularDay(e)`, `Rains(e)`, where `e` refers to an individual representing the current environment. Program 7(a) turns on the air conditioners exactly an hour after the `CanicularDay` quality was created. Program 7(b) adds an additional constraint: the air conditioners will start in a canicular day only if current temperature reached or exceeded 35 degrees. Finally, Program 7(c) can be used to stop the air conditioners if, in a canicular day, it starts raining.

- (a) `current_moment() is Timestamp(?t+1:00/?d.?m.?y),`
`CanicularDay(e) just_after Timestamp(?t/?d.?m.?y)`
- (b) `current_moment() is Timestamp(?t+1:00/?d.?m.?y),`
`CanicularDay(e) just_after Timestamp(?t/?d.?m.?y),`
`Timestamp(?t/?d.?m.?y) during CurrentTemp(?temp), ?temp > 35deg`
- (c) `current_moment() is ?t+1:00/?d.?m.?y,`
`CanicularDay(e) just_after Timestamp(?t/?d.?m.?y),`
`CanicularDay(e) just_before Rains(e)`

Program 7. Temporal constraints

More complex models, conceptually similar to Programs 6, 7 have been simulated and tested using COOL [6], an object oriented extension for CLIPS. A translation mechanism, from facts to web service invocations and vice-versa was developed, in order to test real device behavior. COOL was especially useful for encoding the hypergraph. Actions were simulated using the method presented in Section 3.3.

5 Related Work

Most modeling methods aimed at temporal representation and reasoning focus more on formal specification, and less on actual implementations and computational effort. They are rather suitable for reasoning about a static discourse containing temporal information. Modeling time-dependent evolution is not straightforward in these approaches. It is the case of Description Logics (DL) and temporal extensions of DL. For instance, OWL [5], an ontology modeling language based on DL, focuses on representing a given state-of-the-world, using primitives such as: individuals, concepts, and properties. Here, time is not directly addressed. However, attempts to incorporate temporality exist, and we distinguish between two directions: (1) building time-related concepts on top of existing primitives, thus creating meta-ontologies able to provide some temporal reasoning, such as OWL-Time [12] and (2) extending the modeling approach with new primitives related to time. The former approach suffers from the following pitfalls: complex temporal ontologies are difficult to develop, reasoning is

often intractable, and most important, evolution cannot be represented explicitly. Individuals are inherently (and permanently) bound to concepts or properties, and their enrollment cannot be changed. These ontologies are useful for the disambiguation of time-dependent information, but are unable to model the evolution of real-world processes. Temporal extensions of Description Logic's [1] are an example of the latter approach. They increase the dimensionality of the representation, by adding a new temporal component. As a result, in these settings, instances of a concept are seen as pairs consisting of individuals, and the intervals on which they are enrolled in a particular concept. The extension of a concept becomes a Cartesian product between sets of individuals and sets of intervals. This approach makes model creation cumbersome and reasoning computationally difficult.

Another well known approach is the Temporal Logic of Intervals (TLI), proposed by Allen [2]. It introduces an interval ontology used for the representation of events, properties and temporal change. Allen defines seven basic relations (*before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, *equal*) which, together with their inverses allow a complete characterisation of intervals. An implementation of TLI uses a graph-based algorithm for temporal reasoning [2]. In such a graph, nodes stand for intervals, and edges represent temporal relations between intervals. After the graph is built, temporal reasoning is done by exploring it and inferring new interval relations. Less importance is given to the dynamic nature of the discourse. In contrast, our approach doesn't solely focus on a mechanism for analysing temporal information, but provides means for modeling the desired evolution of a particular application. The evolution is captured in the hypergraph, by continuously adding new instances, according to ontology definitions.

Event Calculus (EC) is a modeling method dealing with events, and the effects they produce [2]. There are some similarities with respect to our approach: events resemble actions, effects correspond to the creation or termination of qualities, time-points are similar to hypernodes. EC also introduces time intervals, that may correspond to quality edges. Temporal representation in EC is based on clauses such as *happens(event, time-point)* – marking the occurrence of an event, *initiates(event, property, time-point)* – marking the initiation of *property*, started by *event*, at the moment *time-point* and *terminates(event, property, time-point)* – which similarly terminates a property. In our approach, actions inherently belong to nodes from the hypergraph, whereas nodes delimit the moments in time when a quality holds. The hypergraph doesn't contain time intervals explicitly, but using timestamps, temporal durations can be computed. Reasoning in EC is based on establishing the truth value of first-order predicates. Compared to a hypergraph, where the life-span of qualities can be easily traced, the FOL-based representation from EC makes reasoning more difficult.

In [13], a declarative language for specifying web service composition (or processes) is introduced. The solution is based on Linear Temporal Logic (LTL). Using LTL formulas, temporal constraints between service invocations are specified. They replace conventional control-flows specific to imperative languages. During the execution of a process, some constraints may not be satisfied, as

not all services have been invoked. Nonetheless, when the process ends, all constraints must be satisfied. The approach from [13] uses only a subset of LTL and thus has limited expressive capabilities.

6 Conclusions and Future Work

There are many cases when devices internally implement intelligent behavior, but often this behavior is rigid, cannot be adjusted to more particular needs, and cannot be extended. Most importantly, intelligent behavior comes with a considerable increase of device cost. It is therefore more convenient to use devices with simple hardware, and to transfer intelligence to software components that are easier to design and update. The proposed modeling method and language are suitable for this endeavor. Moreover, with the introduction of a hypergraph, the approach has the advantage of reducing the temporal reasoning process to simple graph traversals. The chosen declarative approach has several benefits: declarative models are small in size, easy to write, and favor model checking and verification techniques.

When using COOL to represent a hypergraph, the simulation of instantaneous actions proves difficult, and complicates model specification. In addition, the unification mechanism described in Section 3.4 is not supported natively in CLIPS. For these reasons, a specific language as well as an interpreter, are being developed. The interpreter will interface with web services and will be able to: (1) translate service state changes to actions and deliver them to the model, and (2) translate actions generated by the model to web service invocations.

A current possible limitation of the proposed language with respect to CLIPS, is the inability to have qualities introduce other qualities (as described in Section 3.3). While this behavior might be simulated using a special action (**non-stop**) that is constantly calling itself, this approach is computationally inefficient. An intelligent alternative for having qualities introduce other qualities is planned as future work.

It is important to emphasize that the modeling method has potential advantages to numerous other applications in areas not necessarily restricted to device control or Service Oriented Architectures. Such areas remain to be further investigated.

Acknowledgment. The research presented in this paper is supported by national project: “TRANSYS Models and Techniques for Traffic Optimizing in Urban Environments”, Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RUPD ID: 238. The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

References

1. Artale, A., Franconi, E.: A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence* 30, 171–210 (2001), <http://portal.acm.org/citation.cfm?id=590341.590357>
2. Augusto, J.C.: The logical approach to temporal reasoning. *Artif. Intell. Rev.* 16, 301–333 (2001), <http://portal.acm.org/citation.cfm?id=565277.565279>
3. Carmichael, D.J., Kay, J., Kummerfeld, B.: Consistent modelling of users, devices and sensors in a ubiquitous computing environment. *User Modeling and User-Adapted Interaction* 15, 197–234 (2005), <http://portal.acm.org/citation.cfm?id=1101018.1101052>
4. Giumale, C., Negreanu, L.: Reasoning with fluid qualities. In: 17th International Conference on Control Systems and Computer Science, CSCS-17, vol. 2, pp. 197–203 (December 2009)
5. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: Owl 2: The next step for owl. *Web Semant.* 6, 309–322 (2008), <http://portal.acm.org/citation.cfm?id=1464505.1464604>
6. Giarratano, J.: *Clips reference manual* (1994)
7. Wielemaker, J., Schrijvers, T., Triska, M., Lager, T.: Swi-prolog. *CoRR abs/1011.5332* (2010)
8. Juarez, J.M., Campos, M., Palma, J., Marin, R.: Computing context-dependent temporal diagnosis in complex domains. *Expert Syst. Appl.* 35, 991–1010 (2008), <http://portal.acm.org/citation.cfm?id=1383655.1383743>
9. Li, S., Ying, M.: Region connection calculus: its models and composition table. *Artif. Intell.* 145, 121–146 (2003)
10. de Melo, G., Weikum, G.: Towards a universal wordnet by learning from combined evidence. In: *Proceeding of the 18th ACM conference on Information and knowledge management. CIKM 2009*, pp. 513–522. ACM, New York (2009), <http://doi.acm.org/10.1145/1645953.1646020>
11. NASA: Clips website (December 2010), <http://clipsrules.sourceforge.net/WhatIsCLIPS.html>
12. Pan, F.: *An Ontology of Time: Representing Complex Temporal Phenomena for the Semantic Web and Natural Language*. VDM Verlag, Saarbrücken (2009)
13. Pesic, M.: *Decserflow: Towards a truly declarative service flow language*, pp. 1–23. Springer, Heidelberg (2006)

Expressing Conceptual Graph Queries from Patterns: How to Take into Account the Relations

Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez

IRIT, Université de Toulouse le Mirail, Département de
Mathématiques-Informatique, 5 allées Antonio Machado, F-31058 Toulouse Cedex
{camille.pradel,ollivier.haemmerle,nathalie.hernandez}@univ-tlse2.fr

Abstract. Our goal is to hide the complexity of formulating a query expressed in a graph query language such as conceptual graphs. We propose a mechanism allowing one to express queries in a very simple pivot language, mainly composed of keywords and relations between keywords. Our system associates the keywords with the corresponding elements of the support (concept types, relation types, individual markers). Then it selects pre-written query patterns, and instanciates them with regard to the keywords of the initial query. Several possible queries are shown to the user. These queries are presented by means of natural language sentences. The user then selects the query he/she is interested in. The query conceptual graph is then built.

1 Introduction

The goal of our work is to facilitate the expression of queries expressed in graph languages (SPARQL, conceptual graphs). Two kinds of approaches have been proposed to solve that issue. The first one consists in helping the user formulate his/her query in an interrogation language adapted to the formalisation used for representing the annotations. This approach is not always adapted to end-users: to write a query, a user needs to know the syntax of the language and the representation of the data managed by the system. Some GUI systems propose an alternative to this issue. We can cite [1] for RQL queries, [2] for SPARQL queries or CoGui [3] for CGs in general which can be used to design a CG query used in systems such as [4]. Even if these kinds of GUI are useful for end-users, the user still needs time to get used to the GUI and to formulate his/her query by means of a graph language. The work introduced in [5] aims at extending the SPARQL language and the query projection mechanism in order to add keywords and wildcards when the user does not know the schema of the data to query. This approach needs the user to know the SPARQL language.

Other works, such as ours, aim at automatically or semi-automatically generating formal queries from keywords. The user can then express his/her information need in an intuitive way without knowing the interrogation language or the KR formalism used by the system. Approaches have been proposed for generating formal queries expressed in different languages such as SeREQ [6], SPARQL

[7,8]. In these systems, the generation of the query requires the following steps: (i) mapping the keywords to semantic entities defined in the knowledge base, (ii) building query graphs linking the entities previously detected by exploring the knowledge base, (iii) ranking the built queries, (iv) making the user select the right one (the selection is often facilitated by the presentation of a sentence expressing the meaning of the graph). The existing approaches focus on three main issues : optimizing the first step by using external resources (such as WordNet or Wikipedia) [6,9], optimizing the knowledge exploration mechanism for building the query graphs [7,8], and enhancing the query ranking score [9].

In [10], we presented a way of building a CG query from a user query composed of a set of keywords. The proposed mechanism was designed in order to hide the complexity of building queries in terms of graphs, which is quite non-natural for end-users. Our work is based on two observations. The first one is that end-users need simple languages that are mainly limited to keywords as it is the way they are all used to expressing their queries on the current Web. The second one is that, in real applications, queries expressed by the users tend rather to be variations around a few typical query families. These observations led us to propose a mechanism which transforms an end-user query expressed in terms of keywords into a CG query by modifying pre-defined typical query patterns according to these keywords.

Our approach differs from existing ones in the way that we propose to enhance the effectiveness and the efficiency of the query building step by using pre-defined query patterns. The use of patterns avoids exploring the ontology to link the semantic entities identified from the keywords since potential relations are already expressed in the patterns. The process thus benefits from the pre-established families of frequently expressed queries for which we know that real information needs exist. The main issue is being able to select the most suitable pattern and to adapt it to the initial query.

A limitation of our approach was that we did not take into account the relations: the keywords belonging to the user query were exclusively associated with concept types. Another limitation of our work was linked to the use of single keywords which does not allow the user to qualify a keyword by its type. For example, when the user asked for “Eastwood”, it was impossible to express if he was considered as an actor or as a director. This article extends the work of [10] in two ways. The first one is that we take into account the relation vertices in the final query graph: an element of the initial query of the user can still be associated with a concept vertex in the final query, but it can also be associated with a relation vertex. The second extension is that we replace the language of the user queries – a collection of keywords – by a simple language which allows the user to qualify a keyword with another keyword, or to express the relationship existing between a keyword and another keyword. This query language can be used directly by the user, but a perspective of our work is to propose ways of translating natural language queries into this simple language. This is why we call this language the “pivot language”. Note that this article does not address the issue of translating a natural language query into a pivot language query.

Section 2 presents an overview of our system as well as the support and the query patterns we use. The following section describes the process, from the syntax of the queries, the matching of the keywords to the entities of our knowledge base, the selection of the query patterns, the ranking of the potential queries to the generation of the conceptual graph queries. Finally, section 4 presents our implementation and the first experimental results.

2 Overview of Our System

2.1 Description of the Process

The process of our system is the following. Based on a query expressed in the pivot language, and on the terminological knowledge contained in the CG support, the query is matched to elements of the support (concept types, relation types, individual markers). Then we map query patterns to these elements. The different mappings are presented to the user by means of natural language sentences. The selected sentence allows us to build the final CG query.

2.2 The Terminological Knowledge: The Support

The terminological knowledge we use is based on the classic definition of a support [11,12] extended to take into account the possibility of associating synonym terms with individual markers, concept types and relation types. We use an extension of the definition presented in [10] which adds the possibility of associating synonyms with relation types.

A support is a 6-uple $S = (T_C, Syn_{T_C}, T_R, Syn_{T_R}, M, Syn_M)$, T_C being the partially ordered set of concept types, Syn_{T_C} a set of synonyms for the concept types of the topic, T_R the partially ordered set of relation types, Syn_{T_R} a set of synonyms for the relation types of the topic, M the set of individual markers, which are instances of concepts, Syn_M a set of synonyms for the individual markers. Fig. 1 shows a part of our support.

In order to enable our process to generate a query graph from a set of keywords, we store for each concept type t_c a set of synonyms denoted $Syn_{T_C}(t_c)$. For example we have $Syn_{T_C}(Film) = \{“film”, “movie”\}$.

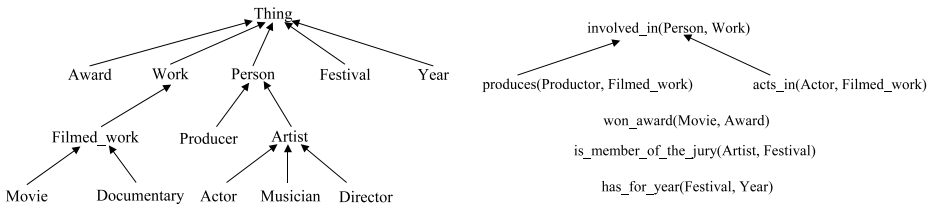


Fig. 1. A part of the support used in our examples. The left part of the schema represents the concept type set, the right part represents the relation type set.

The relation types belonging to T_R represent the nature of the links between concepts in the conceptual graphs: *located_in*, *involving*, *chartered_by*, ... are relation types we use in our application. In order to enable our process to generate a query graph from a set of keywords, we store for each relation type t_r a set of synonyms denoted $Syn_{TR}(t_r)$. For example we have $Syn_{TR}(has_as_jury_president) = \{“has as jury president”, “is presided by”\}$.

The set of individual markers M contains the instances of concept types. We store for each marker m , a set of synonyms denoted $Syn_M(m)$. For example we have $Syn_M(EmirKusturica) = \{“Emir Kusturica”, “Kusturica”, “Kusta”\}$.

2.3 The Query Patterns

A pattern is composed of a conceptual graph which is the prototype of a relevant family of queries. Such a pattern is characterized by a subset of its vertices – either concept or relation vertices –, called the *qualifying vertices*, which can be modified during the construction of the final query graph. It is also described by a sentence in natural language in which a distinct substring must be associated with each qualifying vertex. For now, the patterns are designed by experts who know the application domain and the general shape of the annotations of the documents. The CGs are built manually, the qualifying vertices are selected by the designer of the pattern who also gives the sentence describing its meaning.

Definition 1. *A pattern p is a 4-uple $\{\mathcal{G}_p, \mathcal{C}_p, \mathcal{R}_p, \mathcal{S}_p\}$ such that:*

- \mathcal{G}_p is a conceptual graph describing the pattern. Such a conceptual graph only contains binary relation vertices¹.
- $\mathcal{C}_p = \{c_1, c_2, \dots, c_n\}$ is a set of n distinct generic concept vertices belonging to \mathcal{G}_p , called the *qualifying concepts* of the pattern.
- $\mathcal{R}_p = \{r_1, r_2, \dots, r_m\}$ is a set of m distinct relation vertices belonging to \mathcal{G}_p , called the *qualifying relations* of the pattern.
- $\mathcal{S}_p = \{s, (wc_1, wc_2, \dots, wc_n, wr_1, wr_2, \dots, wr_m)\}$ is a description of the meaning of the pattern in plain text (sentence s) and a collection of $n + m$ distinct substrings corresponding to the qualifying vertices. The wc are the expressions corresponding to the qualifying concept vertices of the pattern (wc_i corresponds to the concept c_i), the wr are the expressions corresponding to the qualifying relation vertices of the pattern (wr_i corresponds to the relation r_i).

Example 1. *In this article, we use in our examples two patterns, p_1 and p_2 . Fig. 2 and 3 respectively present the conceptual graph associated with p_1 and p_2 .*

$\mathcal{C}_{p_1} = \{c_{11}, c_{12}, c_{13}, c_{14}\}$. $\mathcal{R}_{p_1} = \{r_{11}, r_{12}\}$. \mathcal{S}_{p_1} is the sentence “A movie _{w_{c11}} won award in _{w_{r11}} a festival _{w_{c12}} in a year _{w_{c13}} when an artist _{w_{c14}} is a member of the jury _{w_{r12}} ”.

$\mathcal{C}_{p_2} = \{c_{21}, c_{22}, c_{23}, c_{24}\}$. $\mathcal{R}_{p_2} = \{r_{21}\}$. \mathcal{S}_{p_2} is the sentence “An artist _{w_{21}} won award _{w_{r21}} for a film _{w_{c22}} during a festival _{w_{c23}} in a year _{w_{c24}} ”.

¹ This limitation of our implementation could easily be relaxed

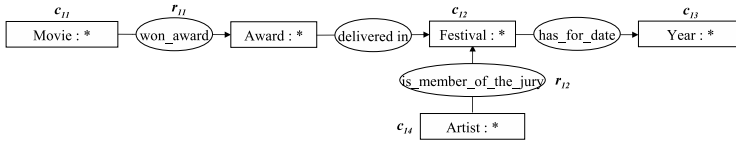


Fig. 2. The conceptual graph \mathcal{G}_{p_1} composing pattern p_1

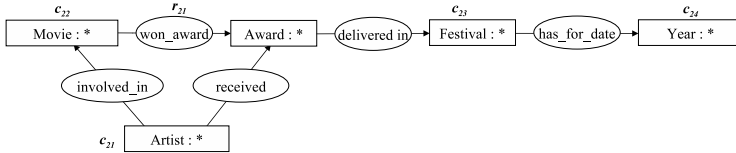


Fig. 3. The conceptual graph \mathcal{G}_{p_2} composing pattern p_2

3 Description of the Process

3.1 Step 1: Expressing the User Queries

The pivot language we propose is an extension of the language composed of keywords. The user can express a query by means of simple keywords, without knowing if they can be associated with a concept type, a relation type or an individual marker. He/she can “qualify” a keyword with another keyword, as well as qualifying, by means of a keyword, the relationship he/she wants to express between two other keywords. All these possibilities can be combined in order to propose a more sophisticated query.

A user query expressed in our pivot language is composed of a set of subqueries which can belong to one of the following types: (i) a single word w , which is considered as a simple keyword, in the usual meaning of the classic search engine. For example, the query “man” means that we search for the men; (ii) a pair of words separated by a colon $w_1 : w_2$, which corresponds to a keyword w_1 qualified by the keyword w_2 . For example, the query “man:married” means that we search for the married men; (iii) a triple of words $w_1 : w_2 = w_3$, which corresponds to a keyword w_1 which is linked to the value expressed in w_3 by the relationship w_2 . For example, the query “man:spouse=Carlita” means that we search for the men whose spouse is Carlita.

Definition 2. A *keyword* is a word expressing the meaning of a concept or a relation the user searches for.

Definition 3. A *user query element* can be either:

1. an expression of the form $[?][\$id]keyword$. The optional $?$ symbol means that the user wants to obtain specific results corresponding to that keyword. The optional $\$id$ expression (id is an integer) allows the same keyword to be used to refer to different entities. In such a case, the id values must be different.

2. a literal, which is to be understood in its usual meaning. It is characterized by its type called a **datatype** in the following.

Definition 4. A **user subquery** belongs to one of the sets Q_1 , Q_2 or Q_3 :

- when subquery $q_i = \{e_{i1}\} \in Q_1$, it consists in a simple element e_{i1} , identifying the subject of the subquery, an entity involved in the expected result,
- when subquery $q_i = \{e_{i1}, e_{i2}\} \in Q_2$, it consists in two elements e_{i1} and e_{i2} , separated by a colon; e_{i1} identifies the subject of the subquery, and e_{i2} identifies something that is in some way related to the subject,
- when subquery $q_i = \{e_{i1}, e_{i2}, e_{i3}\} \in Q_3$, it consists in three elements e_{i1} , e_{i2} (separated from e_{i1} by a colon), and e_{i3} (separated from e_{i2} by the character '='); e_{i1} identifies the subject of the subquery, e_{i2} identifies a property of the subject or a relation involving the subject, and e_{i3} identifies the value of the property or the other entity involved in the relation.

Definition 5. A **user query** $q = \{q_1, q_2, \dots, q_n\}$ is a set of one or more user subqueries $q_i \in Q = Q_1 \cup Q_2 \cup Q_3$, separated by a semicolon.

Example 2. Let us express some queries in our pivot language. It is important to note that these queries are not formal ones which can be interpreted by an algorithm; our pivot language is a way of expressing intuitively the information needed by the end-user. Then our system will search for a formal query expressed in terms of CGs corresponding to his/her actual need.

1. List of actors:
?actor
2. List of married actors:
?actor: married
3. List of 40 year old actors:
?actor: age=40
4. We can combine the previous two queries in order to ask for the list of 40 year old married actors:
?actor: age=40; ?actor: married
which can be written in a condensed way: ?actor: age=40, married
5. List of actors whose wife is 40:
?actor: spouse=woman; woman: age=40
6. List of 40 year old actors married in 2010 in Hollywood:
?actor: age=40, marriage; marriage: date=2010, place=Hollywood
That query could also been written:
?actor: age=40, marriage date=2010, marriage place=Hollywood

7. This example will be used in this article to illustrate the query process: list of award-winning movies in festivals in which Emir Kustica was president:
 festival: president= Emir Kusturica, ?award-winning movie

3.2 Step 2: Matching of the Query Elements

For each query element, we first determine all the elements in the support with which this query element can be matched. A matched element can be any element of the support (concept type, relation type or individual marker) or any type of a literal called datatype. A query element can match several entities. Then we assign to each match a trust mark, according to the presumed quality of the match, and to the consistency between the place of the element in its original subquery and the type of the matched element.

Thus, the matching process can be seen as the definition of the function *match* that associates a trust mark $t_{match} \in [0; 1]$ with each pair (e_q, e_s) , e_q being a query element and e_s a support element or a datatype. When $t_{match} = match(e_q, e_s) = 0$, it means that the query element e_q does not match e_s ; otherwise, e_q matches e_s with the trust index t_{match} .

This step of our process consists in finding which matchings are possible for each query element, and computing a trust mark for all the possible matchings. This trust mark is computed in two phases.

The first phase allows us to attribute a trust value with regard to the type of the query element and the quality of the matching. As we have seen, a query element can be matched to a support element or to a datatype:

- A match between a keyword and a support element is considered as possible when similarities are noticed between the keyword and one or more labels of the support element. Since the user is not supposed to know the ontology, and in order to ensure a better robustness to our system, this is done regardless of the element's role in the query; for instance, we also consider matches between a query element appearing as the subject of a subquery from Q_3 and a relation type, and matches between a query element appearing as the relation of a subquery from Q_3 and a concept type or an individual marker from the knowledge base. Just one type of match will not be considered: a queried variable cannot match an individual marker; it would be nonsense. This step is performed using a string comparison algorithm between the keyword of the query element and the labels of the support elements; this algorithm determines the needed trust mark according to the level of similarity.
- To determine whether a match is possible between a query element and some datatypes, the query element must be the second element of a Q_2 subquery or the third element of a Q_3 subquery. Then we try to interpret the keyword as a value of each datatype, and when the interpretation is successfully achieved, then the matching is considered as possible. For instance, the keywords '14-01-2011' and 'the 14th of January, 2011' should match the *date* type. The trust mark depends on the interpretation method.

The second phase is only suitable for the matching with support elements. It aims at balancing the previously calculated trust marks, in order to take into account potential incompatibilities between the query element's role and the type of the matched support resource. Indeed, in some cases, the roles a query element plays in subqueries where it appears allow us to infer the expected type of the matched support element. To this end, we go by the following rules:

- for a query element appearing at least once as the first element of a subquery from Q_2 or Q_3 , or as the third element of a subquery from Q_3 , the matched support elements are expected to be concept types or individual markers,
- for a query element appearing at least once as the second element of a subquery from Q_3 , the matched support elements are expected to be relation types,
- in other cases, the expected type is not constrained.

For each match generated in the previous step that does not respect those rules, we decrease its associated trust mark by multiplying it by a determined factor $f_{incompatible} \in]0; 1[$. We nevertheless continue to consider that match as possible, in case the initial user's understanding of the facts was not the same as that of the ontology. This is why we consider that matching as less relevant but still possible.

Example 3. *In the previously introduced example, “festival” matches the concept type “festival” with trust mark 1.0, and the relation types “is delivered in the festival” and “is a member of the festival jury” both with trust mark 0.325; “president” matches the relation types “has as jury president” and “is president of the jury” both with trust mark 0.667; “Emir Kusturica” matches the individual marker “Emir Kusturica” with trust mark 1.0; and “award-winning movie” matches the concept type “movie” with trust mark 0.667, the concept type “award” with trust mark 0.533, the concept type “tv movie” and the relation type “won award in” both with trust mark 0.5.*

3.3 Step 3: Mapping the Patterns to the User Query

The next step consists in mapping the patterns to the user query. To this end, the first purpose is to figure out for each pattern element all conceivable mappings to query elements – called element mappings – and their respective trust marks.

Thus, the element mapping process can be seen as the definition of the function map that associates a trust mark $t_{map} \in [0; 1]$ with each pair (e_p, e_q) , e_p being a pattern element and e_q being a query element. When $t_{map} = map(e_p, e_q) = 0$, it means that pattern element e_p is not mappable to e_q ; otherwise, e_p can be mapped to e_q with trust index t_{map} .

When performing this process, several cases can occur:

1. the pattern element refers to a concept type c_1 . There will be a possible element mapping m_e with trust mark t_{map} from this pattern element to every query element e_q verifying the following condition: e_q matches with trust mark t_{match} a concept type c_2 or an instance of this concept type, and:

- c_2 is the same concept type as c_1 ; in this case, the element mapping trust mark is equal to the involved matching trust mark: $t_{map} = t_{match}$;
 - or c_2 is an ancestor of level l of c_1 ; in this case, the element mapping trust mark is equal to the involved matching trust mark decreased by multiplying it l times by a determined factor f_{anc} : $t_{map} = t_{match} * (f_{anc})^l$;
 - or c_2 is a descendant of level l of c_1 ; in this case, the element mapping trust mark is equal to the involved matching trust mark decreased by multiplying it l times by a determined factor f_{desc} : $t_{map} = t_{match} * (f_{desc})^l$;
2. the pattern element refers to a relation type r_1 . We can determine possible element mappings with the same method, considering each query element that matched a relation type r_2 , r_2 being the same relation type as r_1 , one of its ancestors or one of its descendants.
 3. the pattern element refers to a datatype. There will be a possible element mapping m_e with trust mark t_{map} from this pattern element to every query element that matches with trust mark t_{match} this same datatype, and we will have $t_{map} = t_{match}$

In the following, when a query element e_q matches a support element e_s (concept type, relation type, individual marker or datatype), and because of that match, we can infer that a pattern element e_p is mappable to e_q , then we say that e_p maps e_q through e_s .

We can then generate for each pattern all conceivable mappings to the user query – called query mappings. A query mapping m_q consists in a set of element mappings (one or several pattern elements mapped to one or several query elements). In a considered query mapping, a pattern element can be mapped just once, i.e. it cannot be involved in more than one element mapping, whereas query elements can be mapped several times. For each pattern, we build the set M of all possible query mappings the following way: at the beginning, M contains only one query mapping, which is an empty set (it contains no element mapping); then, for each of the pattern elements e_p of the considered pattern,

- if there is no element mapping involving e_p , we do nothing,
- if there are n element mappings involving e_p , we duplicate the original mapping set into $n + 1$, each set considering a different element mapping (the element mapping is added to each query mapping of the copied set), and the last one considering that there is no element mapping involving the considered pattern element. Thus, all mapping cases concerning this pattern element are considered (including the case where no mapping is used) and no query mapping containing two or more element mappings involving the same pattern element is generated.

Example 4. *In our example, this step would lead to the discovery of several possible element mappings; for instance, “festival” from the first pattern p_1 is mappable to the query element “festival” through the concept type “festival” with the trust mark 1.0, “filmed work” is mappable to “award-winning movie” through the concept type “movie” (specialisation of a concept type) with the trust mark*

0.486, “artist” is mappable to “Emir Kusturica” through the individual marker “Emir Kusturica” (instanciation of a concept type) with the trust mark 0.9, “is a member of the jury” is mappable to “president” through the relation type “is president of the jury” (specialisation of a relation type) with the trust mark 0.6. Some pattern elements, like “has for date”, or “year”, are not mappable. Then the generation of all the possible query mappings leads to a set of several mappings. One of them is the “good one”, i.e. it represents the query as thought by the user: it uses pattern p_1 (the closest to the query) and maps c_{12} to “festival”, r_{12} to “president”, c_{14} to “Emir Kusturica”, and c_{12} to “award-winning movie”. But, for the moment this mapping is lost in the bunch of all generated mappings and there are too many of them for the user to exploit them as they are.

3.4 Step 4: Ranking the Mappings

At the end of the previous step, we have a set of query mappings, each one corresponding to a specific query. Only one of them corresponds to the query actually thought by the user. This is why we now want to rank all these mappings, in order to present first to the user the queries which seem to us to be the most relevant. To this end, this step will associate to each mapping a *relevance mark* R , made of several partial marks, each one taking into account some parameters that seem important to us.

Element mapping relevance mark R_{map} represents how much we trust the different element mappings involved in the considered query mapping. The element mappings trust marks t_{map} used to calculate this relevance mark were themselves calculated taking into account the trust marks t_{match} of concerned matchings between query elements and support elements, and the number l of levels in the taxonomic hierarchy between the matched support elements and the pattern elements. The element mapping relevance mark of a query mapping m_q is calculated as follows:

$$R_{map}(m_q) = \frac{\sum_{m_e \in m_q} t_{map}(m_e)}{|m_q|}$$

where $e_q(m_e)$ is the query element involved in element mapping m_e .

Query coverage relevance mark R_{Qcov} takes into account the proportion of the initial user query that was used to build the mapping. The more query elements ignored in the mapping, the lower the query coverage relevance mark will be:

$$R_{Qcov}(m_q) = \frac{|e_q \in q / \exists m_e \in m_q / e_q(m_e) = e_q|}{|q|}$$

Pattern coverage relevance mark R_{Pcov} takes into account the proportion of the pattern qualifying vertices that was used to build the mapping. The pattern coverage relevance mark is not as meaningful as the query coverage relevance mark (it is conceivable that a relevant mapping ignores some pattern elements, it is even one of the advantages of our method we put forward in [10]). However, this mark is more favorable to query mappings from small patterns containing as many element mappings as other query mappings from bigger patterns, which have many unmapped pattern elements and are probably not the most relevant.

$$R_{Pcov}(m_q) = \frac{|e_p \in p / \exists m_e \in m_q / e_p(m_e) = e_p|}{|p|}$$

where $e_p(m_e)$ is the pattern element involved in element mapping m_e .

We can calculate the final *relevance mark* R from previous partial marks:

$$R(m_q) = f_{map}R_{map}(m_q) + f_{Qcov}R_{Qcov}(m_q) + f_{Pcov}R_{Pcov}(m_q)$$

with $f_{map} + f_{Qcov} + f_{Pcov} = 1$

Example 5. *For the sake of brevity, we do not describe here the whole rating process for our previous mapping example. Its relevance mark, calculated in our system implementation presented in section 4, is equal to 0.85 and is the highest in the set of all generated patterns, which leads us to believe that our relevance mark is... relevant.*

3.5 Step 5: Generating Explicative Sentences and the Query

The last step of our process consists in presenting the results to the user, and allowing him to query the knowledge base. To this end, we generate for each mapping a sentence in natural language explaining the query represented by the mapping, and present them to the user in decreasing relevance order. Thus, reading the explicative sentences, the user can easily understand the meaning of each query and choose as soon as possible the one matching his/her need. The system then formulates from the chosen mapping the final query expressed in the required graph query language. Both operations, generating explicative sentences and formulating the query graph, are trivial, thanks to the explicative sentence attached to each pattern and to the graph architecture of each pattern.

For each mapping, the generation of an explicative sentence is carried out by taking the generic sentence attached to the mapped pattern and by personalizing it, replacing for each element mapping the substring associated with the mapped pattern element by a string obtained applying following rules:

- if the mapped pattern element is referring to a concept type (it is mapped to a query element either through a concept type, or through an individual marker),
 - if it is mapped through a concept type (the same concept type, a descendant or a parent concept type), the replacement string is a label of that matched concept type (preferably the label that led to the matching between query element and concept type), preceded by the indefinite article “a”: we are referring to any instance of that concept type,
 - if it is mapped through an individual marker (of that concept type, a descendant or a parent concept type), the replacement string is a label of that matched individual marker (preferably the label that led to the matching between the query element and the individual marker),
- if the mapped pattern element is referring to a relation type (it is mapped to a query element through that same relation type, a descendant or a parent relation type), the replacement string is a label of that matched relation type (preferably the label that led to the matching between the query element and the relation type),

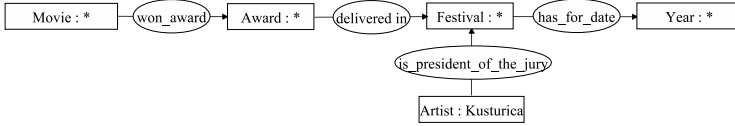


Fig. 4. The final query graph

- if the mapped pattern element is referring to a datatype (it is mapped to a query element whose interpretation as a literal of that type was successful), the string representation of the interpreted value is used for replacement,
- if the mapped query element is a queried variable and regardless of the mapped pattern element’s type, the replacement string is syntactically emphasised, in order to show to the user what is the object of the query.

As regards pattern elements which are not involved in any element mapping, we keep the default associated substrings in the sentence.

The query graph of the selected query mapping is the pattern graph, apart from the odd detail; it is generated using the procedure presented in [10], except that relation vertices can be modified by specialisation or generalisation, in a same way that it was made in [10] with concept vertices.

Example 6. *Executing the explicative sentence operation to the mapping described in our previous example will result in the following sentence (expressions in italics have been replaced, even when they are equal to default substrings; expression in bold refers to the object of the query): “**A movie** won an award in a festival in a year when Emir Kusturica is president of the jury”.*

Because the corresponding mapping obtained the best relevance mark, this sentence will be the first displayed to the user and, if chosen, the generated query graph will be the one presented in Fig. 4.

4 Implementation and Experimentation

A prototype of our approach has been implemented in order to evaluate its effectiveness; it has been adapted to integrate the semantic web framework and generates SPARQL queries in order to let the user easily query RDF triples. Thus, it has been implemented in Java using the Jena API. The first observation we can express concerns the required execution time: even with a growing knowledge base, the time needed to perform a query interpretation is quite reasonable. Our knowledge base containing 68 classes, 56 relations and 194 instances, the time necessary to process each step presented above and sort the obtained mappings, running on a 2.4GHz dual core CPU, never exceeds one second for the most complex queries, and is on average equal to 300ms.

For the evaluation, we asked 24 distinct persons to state queries about the cinema from an artistic point of view. We collected 160 queries, each one composed of a set of keywords and a sentence in natural language explaining the information need. To obtain the following results, we randomly selected 40 queries

among the set of collected queries. Then three distinct users independently formulated each query using the pivot language presented in section 3.1. We thus obtained for each query three formulations which were most of the time slightly different, because of the subjective aspect of our pivot language. We then evaluated the MRR, which consists in identifying for each query interpretation the position r of the right pattern in the sorted list and then calculating the average of $1/r$. Performing this test on our 120 examples (3 formulations for each of the 40 selected queries), we obtained 0.81. This score is noticeably better than those obtained in [78]. Moreover, in 89% of cases, the right pattern appears in the first three elements of the sorted list, which means that the user will have no difficulty in spotting and selecting it.

The development presented here considerably improves the interpretation of queries implying relations, like for instance “novel: author= Fred Vargas, adaptation= ?movie; ?movie: director= Josee Dayan” which allows to ask for the list of movies directed by Josee Dayan which are adaptations of novels by Fred Vargas. As explained earlier, these results were obtained from queries written by distinct users, which shows that our system is flexible enough to adapt itself to the subjectivity of queries and to the distinct view of each user.

5 Conclusion

In this paper, we proposed a development of the system introduced in [10], in order to allow the user to express relations in the queries he/she formulates. To this end, we proposed a new pivot query language that we wanted to be simple, intuitive and flexible, and we adapted the query interpretation process to make it take relations into account. The first evaluation results are very encouraging. We can now handle some queries that previously gave bad results, and we have other improvements in prospect. We noticed indeed that, for most of the cases where the right mapping is badly ranked, this happens because the right mapping was overtaken by others which actually do not respect the structure of the original user query. This is why we have to work towards improving the relevance evaluation by taking into account the query structure.

We also plan to work towards improving the system’s ergonomics. The pivot language could be extended with anonymous variables in order to allow the user to refer to and query entities about which he/she has no information. The mapping generation could become more dynamic by adding the possibility of using regular expressions on parts of patterns, drawing ideas from [13]; these parts could then be omitted or repeated in the final matching. This would improve the readability of the explicative sentences of each mapping and to make patterns more generic (therefore less numerous). Finally, in spite of current performances which are quite reasonable, evolutions from the algorithmic point of view will probably be necessary to scale the Web; we hope to reduce the complexity of the method by using heuristics which would drive the mapping generation and prevent the generation of mappings considered as not relevant enough.

References

1. Athanasis, N., Christophides, V., Kotzinos, D.: Generating on the fly queries for the semantic web: The ics-forth graphical rql interface (grql). In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 486–501. Springer, Heidelberg (2004)
2. Russell, A., Smart, P.R.: Nitelight: A graphical editor for sparql queries. In: Bizer, C., Joshi, A., (eds.) International Semantic Web Conference (Posters & Demos). CEUR Workshop Proceedings, CEUR-WS.org, vol. 401 (2008)
3. CoGui. A conceptual graph editor. Web site (2009), <http://www.lirmm.fr/cogui/>
4. Genest, D., Chein, M.: A content-search information retrieval process based on conceptual graphs. *Knowl. Inf. Syst.* 8(3), 292–309 (2005)
5. Elbassuoni, S., Ramanath, M., Schenkel, R., Weikum, G.: Searching rdf graphs with sparql and keywords. *IEEE Data Eng. Bull.* 33(1), 16–24 (2010)
6. Lei, Y., Uren, V.S., Motta, E.: Semsearch: A search engine for the semantic web. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 238–245. Springer, Heidelberg (2006)
7. Zhou, Q., Wang, C., Xiong, M., Wang, H., Yu, Y.: Spark: Adapting keyword query to semantic search. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 694–707. Springer, Heidelberg (2007)
8. Tran, T., Wang, H., Rudolph, S., Cimiano, P.: Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In: ICDE, pp. 405–416. IEEE, Los Alamitos (2009)
9. Wang, H., Zhang, K., Liu, Q., Tran, T., Yu, Y.: Q2semantic: A lightweight keyword interface to semantic search. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 584–598. Springer, Heidelberg (2008)
10. Comparot, C., Haemmerlé, O., Hernandez, N.: An easy way of expressing conceptual graph queries from keywords and query patterns. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 84–96. Springer, Heidelberg (2010)
11. Sowa, J.F.: *Conceptual structures - Information processing in Mind and Machine*. Addison-Welsey, London (1984)
12. Mugnier, M.-L., Chein, M.: Représenter des connaissances et raisonner avec des graphes. *Revue d'Intelligence Artificielle* 10(1), 7–56 (1996)
13. Alkhateeb, F., Baget, J.-F., Euzenat, J.: Extending sparql with regular expression patterns (for querying rdf). *J. Web Sem.* 7(2), 57–73 (2009)

Unix Systems Monitoring with FCA

Uta Priss

Edinburgh Napier University, School of Computing
www.upriss.org.uk

Abstract. There are many existing software tools for identifying specific and severe IT security threats (virus checkers, firewalls). But it is more difficult to detect less severe and more general problems, such as disclosure of sensitive or private data. In theory, security problems could be detected with existing tools, but the amount of information provided is often too overwhelming. FCA is a promising technology in this application area because it helps to reduce and explore data without prescribing what it is that is searched for from the start. This paper demonstrates the use of FCA for analysing Unix system data with respect to IT security monitoring.

1 Introduction

The background for this paper is IT security. Recently, two major international cyber attacks made for news headlines¹. Most PC users will be aware of the need for virus checkers. But apart from the major risks from illegitimate software, legitimate software can also have unwanted side-effects either because it might disclose information to third parties without the user's knowledge or because it might contain faulty code. For example, a fair amount of personal detail might be stored by a web browser for the purpose of auto-completion of regularly used web forms. Presumably, users can protect themselves by switching on a "private mode" in their browser, but as discussed by Aggarwal et al. (2010), such private modes can never be completely effective because it is not easy to decide conclusively which data must be kept private and which not. Also, since web technologies are constantly evolving, there will probably always be some risks that can be exploited. Currently, many browsers leak information about the browsing history which can be accessed by other websites via "history stealing". Wondracek et al. (2010) claim that it may even be possible to use this to identify individual users of social networking websites.

Special purpose tools (such as browser plugins, virus checkers, and firewalls) may not be sufficient to detect all such problems because they tend to focus on specific, known problems. Another approach is to use monitoring tools which are more general purpose and aimed at detecting when something "odd" happens or, in other words, using a data mining approach for IT security. As usual in data mining applications, the data supplied (in this case by system commands and logfiles) can be overwhelming and difficult to analyse without the help of tools. In addition to the amount of information,

¹ http://en.wikipedia.org/wiki/Operation_Aurora and
<http://en.wikipedia.org/wiki/Stuxnet>

system data often contains cryptic codes and abbreviations, which are more aimed at developers than users. In this paper, it is demonstrated how Formal Concept Analysis (FCA)² can be employed to assist users in making sense of system data. Interestingly, just the conceptualisation of the data already helps with the problem of understanding cryptic codes and abbreviations because codes become more meaningful if a user sees the usage context of a code. In the future, the use of FCA in this area could be supplemented with additional data mining tools, but even FCA alone already provides interesting results. So far there appear to be surprisingly few papers in the literature about FCA applications for IT security as summarised in Section 2.

There are many existing approaches to IT security which is a broad topic. In this paper, only Unix systems (such as Linux and Apple Macintosh OS X) are considered and the assumption is made that problem detection is conducted by using standard Unix commands for reading directory and process structures and logfiles. Most likely the FCA-based high-level technologies discussed in this paper are also applicable to Microsoft Windows. A Windows filesystem is accessible to a Linux partition on the same machine. Furthermore, even though the actual commands are different in Windows, the structures (using users, files and processes) are similar. But that is not further discussed in this paper.

There are numerous existing tools for monitoring operating systems and exploring logfiles³. But most of the existing tools only allow to monitor for known events or are very specialised and apply only to a specific type of logfile. Tools such as PandoraFMS⁴ facilitate the simultaneous monitoring of many servers and many types of problems using sophisticated graphical reports. But users need to specify exactly what the software is supposed to monitor, in what manner and at what times. With respect to logfiles, there are, for example, many tools which allow to monitor and analyse web server logfiles⁵ which record how many users have looked at the webpages, which countries they are from, and the dates and kinds of browsers used. Monitoring for web server errors with such tools is already slightly more difficult. The tools might list the most commonly produced errors and warnings of server-side scripts, but it may not be so easy to detect a hacking attempt against the server with such logfile analytic software, in particular if the hackers are using a novel method. There is, of course, also software for detecting hacking attempts, but such tools often focus on particular techniques (often very low-level⁶) and do not usually help users to explore the data in a more general manner.

It appears that so far a suitable, freely available, multi-purpose, high-level Unix systems monitoring tool does not yet exist. There are, however, existing tools that can be used as building blocks for such a tool (for example, logfile analytic tools, data mining tools and FCA tools). A modular design for such a tool is as follows:

² This paper does not provide an introduction to FCA. Information about FCA can be found online (<http://www.fcahome.org.uk>) and in the main FCA textbook by Ganter & Wille (1999).

³ Such as <http://swatch.sourceforge.net/>

⁴ <http://pandorafms.org>

⁵ Such as <http://sourceforge.net/projects/awstats/> and <http://www.webalizer.org/>

⁶ Such as <http://www.snort.org/>

1. data extraction (use existing tools for collecting data from the system and storing it in comma-separated files)
2. context building (use pre-defined conceptual scales and heuristics for extracting formal contexts)
3. lattice representation (use existing FCA software for visualisation)
4. post-processing (summarise information from lattices in textual format that can be read by users who do not know FCA)

Since the tools for steps 1 and 3 already exist, the focus of this paper is on step 2. Step 4 is left for future research. It would be nice to have ways of summarising information from lattices to make such a tool available for general users. But for now, the focus is on expert Unix users. Presumably, learning to read FCA lattices is far easier than becoming an expert Unix user, therefore this should not be a big hurdle.

The main aim of this paper is the practical application of FCA to IT security. But there is also a theoretical contribution in the area of “Data Weeding” (Priss & Old, 2011), which is the art of selecting appropriate sets of objects and attributes from a complex, many-valued set of data. In general it is difficult to know in advance what lattice to construct for which data set. A similar problem was previously explored by Priss & Old (2010 and 2011) with respect to lexical databases, where neighbourhood lattices appeared to be the most useful structure. With respect to Unix data, the interaction between lattice structures and hierarchies (of files) and sequential (temporal) data is most interesting.

Section 2 of this paper provides an overview of existing IT security FCA research. Section 3 provides some background on temporal data and file hierarchies as used in Unix operating systems. Section 4 suggests five main modelling tasks in this area. Section 5 provides examples of what can be achieved. This is followed by a conclusion.

2 Published FCA Research in the Area of IT Security

Since FCA is commonly used in the data mining area and since IT security is a commercially, legally and politically important application topic, it is surprising that there does not appear to be a greater amount of FCA research in this area. According to an article in the New York Times (Farley, 2006), FCA is used by the US National Security Agency for studying patterns in telephone networks, presumably to detect terrorist cells. Due to the secrecy of such agency no further details about this FCA application are known.

Maybe the first paper on IT-security and FCA was published by Becker et al. (2000). It describes a tool which models dependencies among security guidelines using the Toscana software. The underlying FCA methods used are the usual ones for many-valued contexts.

There are several papers which model the dependencies between software packages or code modules with FCA in order to detect security-related trends. These are based on a method first established in software engineering by Lindig & Snelting (1997). With respect to IT security, Neuhaus and Zimmermann (2009) use this method to trace software vulnerabilities in Linux packages. They model the partially ordered set of the dependencies as a concept lattice which is weighted by the known security risks

associated with some packages. Using the lattice one can then determine the risks for any package and observe how certain packages are the main cause of the vulnerabilities. Another IT security method based on Lindig & Snelting's approach is described by Ganapathy et al. (2007) for identifying security-sensitive operations in legacy code.

There are several papers on using FCA for web log analysis (Zhou⁷ (2004) and Pohle & Spiliopoulou (2002)). As explained above, web log analysis is not necessarily aimed at IT security, but since security-related information is often derived from logfiles, any methods developed in this area are potentially relevant for security analysis as well.

3 Some Unix Background on Temporal Data and File Hierarchies

A Unix operating system provides many sources for detecting problems, often involving logfiles. Of particular interest is temporal data and file hierarchies. Detailed runtime information about processes is available. For each file and directory, the operating system stores when it was last accessed, modified or had its status changed. Temporal information on Unix, however, can be tampered with by users (for example, using the "touch" command). Thus, hackers can change file modification times in order to cover their tracks. But it would be difficult, even for hackers, to remove all traces of their actions. For Unix processes, temporal coincidence can be (but does not have to be) an indication of causal relationship. Files that have identical modification times are often created by a single process or related processes. For example, certain files are updated regularly at the same time when the computer boots up. Installing or updating a software package will lead to several files with the same modification times, although when files are downloaded as part of a software package they sometimes keep their modification time from when they were created on the original computer. In that case, the status change time will be more accurate in showing when the file was installed. Thus, interpreting the temporal information of files requires some knowledge of how times are affected by the operation system.

The file hierarchy in Unix tends to be mostly a tree hierarchy although there can be a few "symbolic links". In traditional Unix systems, the file hierarchy tended to be fairly simple and, ideally, files were placed in standard locations so that they could be found by users. In modern Unix systems, different strategies for file locations might be mixed. For example on Apple's OS X, some files are placed using traditional Unix directories (using lowercase letters), some files are placed in Apple's special hierarchy (starting with uppercase letters) and, if package management software is used, this might be placed in yet another location. Furthermore, the directory structures created by integrated development environments (IDEs) tend to be complex and not really human-readable. Therefore it is usually not possible to gather useful security-related information by manually looking at files or directories. Even if a search is performed for files that relate to a certain pattern, the information that is retrieved can be so complex that it cannot be processed manually.

⁷ There is a paper published in 2009 in an India-Pakistan-based journal which has very similar content. Because of the dates I am assuming Zhou's work is original and the other one is plagiarised.

In this paper, the main Unix commands considered are “find”, “stat” and “lsdf”. These commands allow to select parts of the file hierarchy based on search criteria, to print file attributes and to see which files are opened by a running process. Although some of the options for these commands differ between different flavours of Unix the basic functionality should be the same for all modern versions. We have written a couple of very basic Perl scripts which convert output from these Unix commands into formal contexts. It is our intention to package the scripts as a toolkit with FcaStone and FcaFlint⁸ once we have decided which context forming strategies appear to be most useful.

Apart from monitoring existing, standard information sources, another option is to deliberately collect information in order to detect possible problems. A reason for this is because files and directories change on a regular basis and even backups might not store relevant temporal and other attributes accurately. One possibility is to take static snapshots, for example of the original configuration of a newly installed operating system (running processes, top-level files, standard logfile entries), and to take further snapshots at regular timespans in order to detect changes. With respect to a newly installed piece of software, snapshots might be taken right before and right after the installation and while the software is running, although it may also be sufficient to conduct a search for files that were changed within the last couple of minutes right after the software has been installed. Another possibility is to use tracing software that records dynamic, runtime data. This is because security sensitive events might happen between snapshots and might not be recorded.

4 Modelling with FCA

Although, systems data can be collected in many different ways, the data tends to be of similar types. In particular, tree hierarchies and posets (for directory and process structures), temporal attributes and many-valued formal contexts (for users, processes, devices, files, etc) are of interest. This section discusses *five commonly occurring types* of structures in Unix data and how these can be modelled with FCA. A useful FCA notion for this discussion which may not be widely known is “contingent”. An attribute contingent is the set of attributes in the intent of a concept which belong to the concept but do not belong to any superconcept. An object contingent is defined in the dual manner.

1) Temporal data: an interesting conceptual question is temporal chunking and granularity. Often files whose times differ by just a few seconds will have been created by the same process. In some cases, it might even be suitable to consider files that were created on the same day (during the same session) as being related. There are different possibilities: one can use some heuristic for determining chunks depending on the distribution of the temporal events; one can use FCA for determining the units by calculating a lattice of the raw temporal data and then using the contingents for determining temporal chunks; or one can use a simple strategy of ignoring units smaller than minutes, hours or days. Our experiments (in the next section) seem to indicate that the last

⁸ <http://fcastone.sourceforge.net/>

choice (which is the easiest) is sufficient, but a user needs to decide which cut-off point (hours, minutes, seconds) is appropriate for which set of data. On a more abstract FCA level, the question is how a sequential structure (of temporal units) on the objects or attributes interacts with the conceptual structure of the lattice.

2) Many-valued formal contexts for logfiles and Unix command output: logfiles can have different syntactic structures, such as comma-delimited versus tab- or space-delimited, single-lined versus multi-lined entries, but these details can be dealt with through data-preprocessing. Otherwise, logfiles tend to be similarly structured: usually starting with a timestamp, standard attributes and a short, standardised description of an event. The output for the Unix commands “find”, “ls” and “stat” also is of similar nature. It can be assumed that at the conceptual modelling stage, the data is represented as a many-valued formal context. At least one attribute of such a context tends to be temporal. The others tend to be from a standard set of attributes (users, processes, paths, etc). In some cases one further attribute contains a brief message describing the event. Standard FCA techniques for modelling many-valued contexts exist and need not be further elaborated in this paper.

3) Observing the same data at different times: if snapshots of system data are taken as described in the previous section, formal contexts might be formed which contain the same set as objects and attributes but each belonging to a different snapshot. If data from different snapshots is combined in one lattice, deviations from symmetry could indicate changes. Alternatively a separate lattice could be constructed for each snapshot in which case algorithms would need to be employed that compare lattices. This is interesting, but not further elaborated in this paper.

4) A poset of the file hierarchy: apart from temporal information, which is sequential and can be chunked, the other important re-occurring data structure is the poset formed by the file and directory hierarchy. As discussed in the previous section, Unix file hierarchies tend to be too complex to be manually examined in detail. FCA can help reducing and structuring file hierarchy information. An interesting theoretical question arises as to how the poset on the objects or attributes interacts with the lattice structure. A concept lattice might be used to simplify the poset of the file hierarchy as is demonstrated in the next section.

5) Conceptual scales for recording security information: one problem for IT security is the amount of misinformation that is posted on the web because there is no quality control for web content (e.g. web forum discussions, deliberate misinformation by some companies), nor do search engines consider content quality for their rankings. For example, one can type any process name from the Windows Task Manager into a search engine and will instantly retrieve webpages which claim that this process might be a virus even though in most cases it is completely harmless. In a similar manner to how virus scanners download virus definition files, it might be possible to create conceptual scales which are manually verified by experts and which contain information about normal structures in operation systems which can be used as a comparison to actually occurring structures so that users can check whether something they noticed on their computer is normal or suspicious. This would be a complex task and is beyond the scope of this paper.

5 Four Examples of Exploring Unix System Data with FCA

This section explores several examples of concept lattices in the area of Unix systems data monitoring. The examples below are a first attempt at determining heuristics for context construction in this domain. The examples are all generated by using short scripts that process the output of Unix commands or logfiles on an Apple OS X computer. FcaStone has then been used to produce the pictures, which have not been manually edited. The diagrams use “minimal labelling” which means that objects (in the bottom half of the concept boxes) also belong to their superconcepts; attributes also belong to their subconcepts. Or, in other words, only the contingents are written for each concept. Once we have explored a sufficient number of examples, we intend to package the scripts into a toolkit. The data has not otherwise been modified with the exception that some of the terms have been abbreviated in order to make the figures more readable in a printed medium.

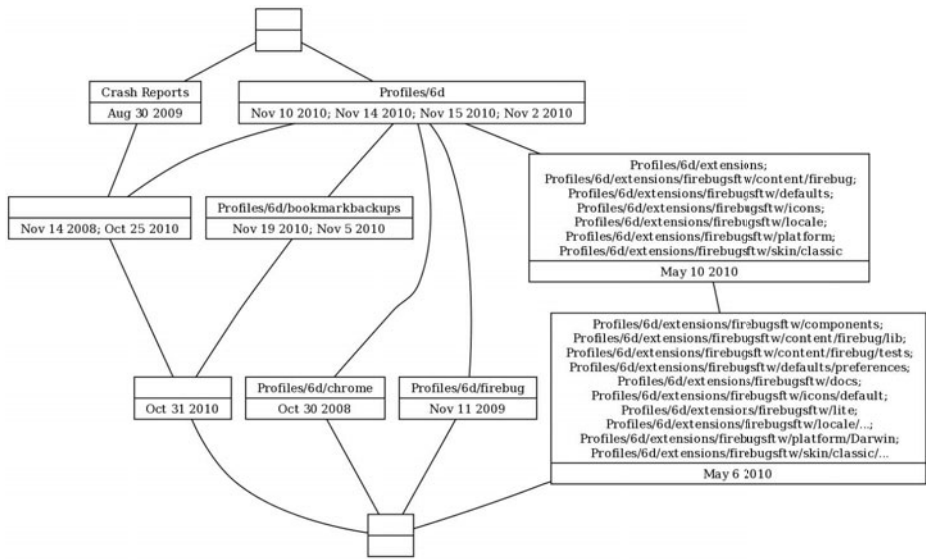


Fig. 1. The files in the Firefox library directory

The first example combines temporal data (file modification times) and file hierarchy data. The data was collected from the Firefox library directory by listing all files in the subdirectories together with their full path names (as formal attributes) and their file modification times (as formal objects). The relationship between objects and attributes was defined as “at that date this file or a file below this directory was modified”. This means that for each full path name all the superdirectories were determined and added to the set of attributes as well. A lattice created directly from this data would be too big to visualise. A bit of experimentation revealed that the lattice could be reduced to a more convenient size (Figure 1) by considering only days and ignoring hours, minutes and

seconds for the objects and by ignoring the bottom-level file/directory for each attribute. (Day-level chunking is not a limitation. Even on a larger server, some parts of the file hierarchy will not get modified after installation or update. This makes a reduction to days feasible. Obviously, there will be other parts of the file hierarchy where day-level chunking is not possible.)

In order to make the lattice more readable a few further reductions were applied to the labels. The formal objects are “restricted” (Priss & Old, 2011) because only dates which belong to at least two files are included. This does, in this case, not change the lattice structure but reduces the number of object labels. Furthermore, the labelling of the attributes was reduced by using the posets of the file hierarchy as follows: for each contingent, the poset was reduced by listing only the top and bottom elements. If all files or subdirectories under a directory belong to the same contingent, this was abbreviated by writing “/...” at the end of the directory name and omitting the files or subdirectories.

In this case the attribute order in the lattice is a tree itself and does not contradict the file hierarchy. For example, none of the concepts jointly under the distinct directories “Crash Reports” and “Profiles/6d” have any attributes in their contingent. One could reduce the path names further by omitting the directories of higher level concepts, but we do not think this will be applicable for many examples and decided against using it.

Considering that the complete dataset of this example contains 62 directories and 477 files, Figure 1 provides a human-readable overview of the data which might not easily be obtained without FCA. Figure 2 demonstrates the effect the contingent-based reduction has on the file hierarchy. On the left is the complete hierarchy (as produced by the Unix “tree” command). On the right is the hierarchy of the attribute set from Figure 1. The reduced hierarchy is focused on the actual events of when files were created or changed.

An expert Unix user can now interpret the data of Figure 1. Crash reports can occur at any times. The dates in the contingent of Profiles and bookmarkbackups tend to be recent. Backups are automatically created and deleted on a regular schedule. The browser is configured to delete cookies and similar data when the browser closes. Therefore certain backup and profile files will be modified on a regular basis. Three dates relate to the firebug extension, presumably representing the times when that extension was installed or updated and last used.

The most interesting date is November 14, 2008 because this date is both under Crash Reports and Profiles in an area where all other dates are recent. In order to understand the significance of this date, an event-driven lattice needs to be constructed. This lattice consists of all files (on the whole computer, not just in this directory) that were modified on November 14, 2008 and their exact modification times. To save space, the lattice is not shown in this paper, because this method is demonstrated for another example below. The event-driven lattice for November 14, 2008 explains that Firefox was installed for the first time on that day. It highlights which other directories belong to a Firefox installation. It might seem contradictory that the date of the chrome⁹ directory is earlier (October 2008) but that is because when files are unzipped they keep their modification

⁹ Incidentally this is a good example of how less experienced users could be confused by internal file names. The word “chrome” here does not mean that Google’s Chrome browser is installed but is a name that is used by Firefox for certain settings.

times from when they were created in the original location. A check revealed that the status change time of the chrome directory is also November 14.

The event-driven lattice almost allows to write a personal diary of the user of that computer for that day showing that the day was started with preparing teaching materials, followed by software installation. One could probably pinpoint the time a lunch break was taken on that day. The amount of historical detail revealed is quite surprising. But because the topic of this paper is security, Figure 3 shows an event-driven lattice for another example. In this case an experiment was conducted. The Firefox browser was opened and a video on a news website was watched; then the browser was closed. Using Unix “find”, all files in the user’s Library directory were retrieved which had been accessed during the last 10 minutes. A lattice was constructed of the files and their exact

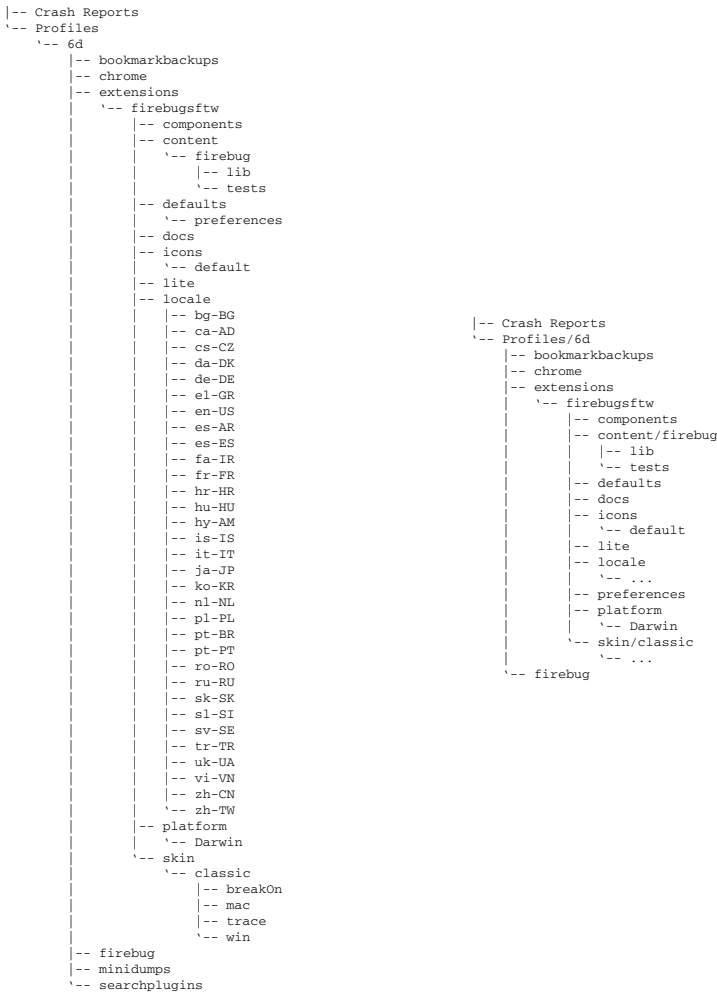


Fig. 2. The FCA based reduction of a file hierarchy

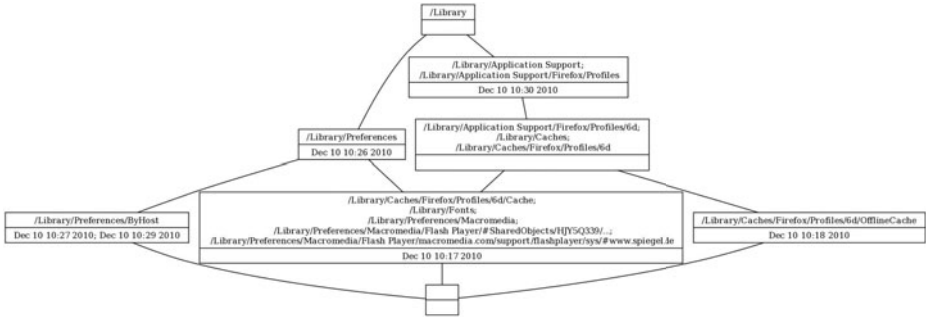


Fig. 3. An event-driven lattice

times (ignoring seconds). The attribute contingents were reduced in the same manner as in Figure 1. This lattice shows two directories which are used for storing Adobe Flash Cookies and which were accessed at the same time as Firefox’s cache. Since all other files and directories in this lattice either relate directly to Firefox or are very general (such as Fonts), this lattice would alert a user to the existence of Flash Cookies, which are different from ordinary web cookies and can be used to create cookies which are undeletable for ordinary users¹⁰.

The first example shows that it is possible to retrieve interesting information about software activity (and user activity) by analysing temporal attributes of files and directories. The second example demonstrates how monitoring of system data might alert users to security and privacy issues which are not visible through normal application software (current browsers do not let users explore Flash Cookies).

So far the examples started either with a directory (Figure 1) or a time unit (Figure 3) and used historical data. In order to establish a more general overview of currently active processes, the “list of open files” (lsOf) command is useful. Unfortunately, the complete output of that command is too complex to be visualised as a lattice. Figure 4 shows an example of the complete data set of user-owned processes, but restricted to showing only the top two directories of each full pathname. The formal objects are names of processes; the attributes are the files (libraries) that are opened by the processes. The user had two browsers open (Firefox and Safari), was looking at some image or pdf file (Preview), executed the lsOf command on the command-line (Terminal, tcsh) and was editing a Latex file (TeXShop). The overview shows that distinct types of processes are running: there are traditional Unix commands (tcsh, lsOf) which do not use Apple’s special libraries. Apple-specific commands can be divided into Applications that were started by a user (under /Applications) and programs that run all the time (under /System, but not under /Applications). The only oddity in the lattice is that TeXShop is not connected to /Users/upriss (in contrast to Safari and Firefox) even though TeXShop is used to edit files under that directory. An explanation might be that TeXShop does not keep these files permanently open. This is a general problem with using lsOf: events that happen very quickly, such as writing to a file or downloading content to a web browser,

¹⁰ As demonstrated by Samy Kamkar in his “Evercookie” <http://samy.pl/evercookie/>

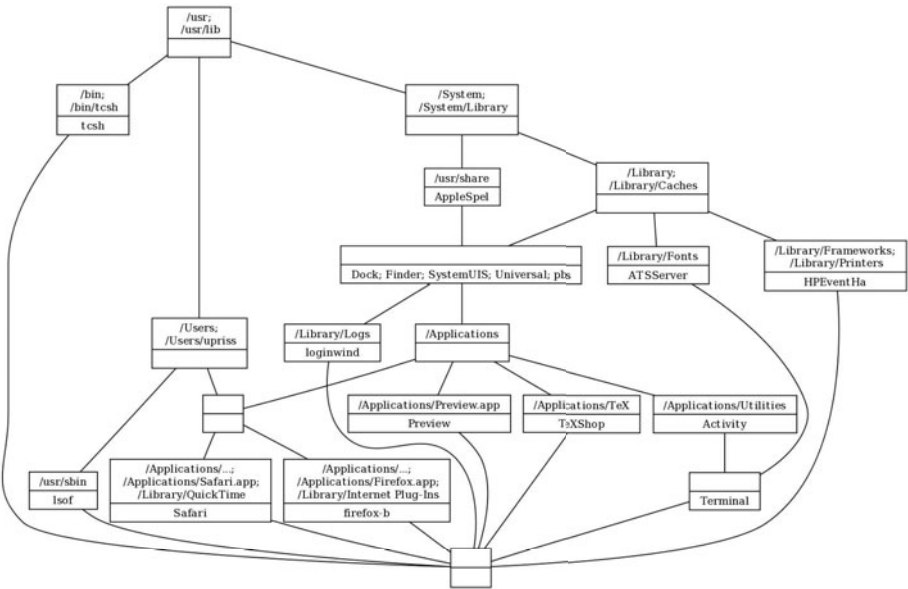


Fig. 4. List of open files

will only be listed by lsof if they happen at that very instant in time. If one needs to be certain to catch all such events, one needs tracing software instead of lsof.

The lattice does not permit to detect any security problems, but then presumably there are no viruses or other rogue processes running on this computer. Apart from detecting rogue processes, another possible security application of lsof data is similar to Neuhaus & Zimmermann’s (2009) lattice of software vulnerability dependencies. Neuhaus & Zimmermann determine the risk associated with each program based on software dependencies. A lattice of lsof data shows the same dependencies but with respect to actually running processes. Of course, in order to investigate the relevant libraries the full data would need to be used instead of using just the top two directories. A lattice of the complete data set is too complex, but if one constructs sublattices for the tree under each top-level directory at a time (/Library, /Applications, /Users) or for subsets of the objects, it becomes manageable.

The final example shows a lattice of the system logfile. In this case the preprocessing consisted of omitting seconds and rounding minutes down to multiples of 10. Furthermore, there are several messages in the file which re-occur frequently but with changing process id numbers. These process id numbers were removed. The resulting lattice is shown in Figure 5. In the system logfile, activities relating to a single event can be spread across several lines. But because they have the same timestamp, they are automatically identified as a single event. In theory it could happen that an event was started a second before the full hour and finishes a second after the full hour. In that case the event would be split but unless this happens often, this would be obvious from the lattice. Figure 5 shows that regular events are automatically grouped. The largest contingent in the lattice refers to the events that follow a wake-up from sleep. In that

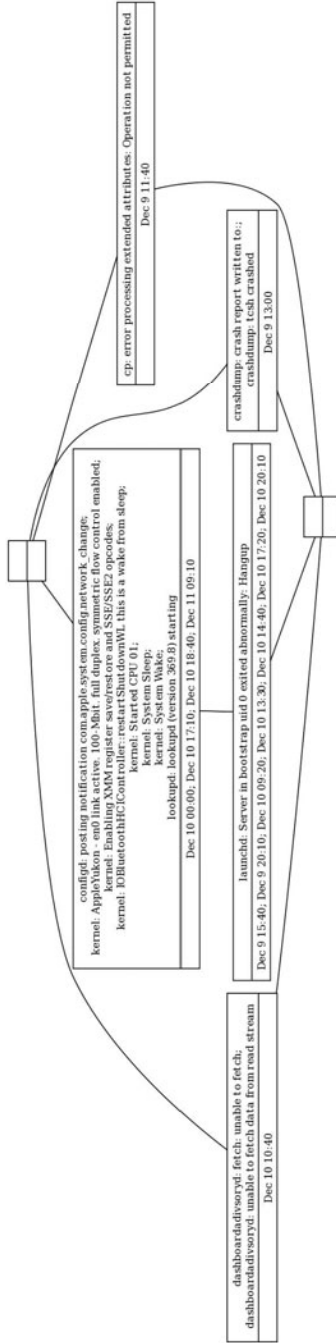


Fig. 5. System log

case the launch daemon needs to be restarted. The subconcept shows that the launch daemon also restarts at other times. The other three concepts refer to one-time events: a crash, a dashboard error and a copy error. There were no security problems detected, but since one-time events are highlighted by the lattice, any new or unusual activity would be visible. A further strategy would be to make a regular copy of these lattices and to compare them occasionally to see whether anything has changed.

6 Conclusion

This paper investigates the use of FCA lattices for analysing Unix system data. The experimental results so far are promising. On a theoretical level, the paper shows that the relationship between lattices and other structures (sequential data and other posets) is of interest. The lattice structure can be used to modify the other structures (for example reducing the complexity of file hierarchies based on FCA contingents). With respect to temporal data, this paper suggests that the easiest approach of not using the lattice structure but simply chunking the time units according to some threshold is sufficient for these applications. It might be of interest, however, to consider methods of Temporal Concept Analysis (Wolff, 2002) in the future.

The paper shows that it is possible to produce lattices that summarise the information and allow data exploration. The challenge lies in finding an appropriate “Data Weeding” (Priss & Old, 2011) technique, i.e., determining which heuristics of object and attribute selection are appropriate. A user still needs to experiment with each new data set. One strategy would be to start with a lattice of the full data. If that is too large, the temporal units could be made larger (ignoring seconds, etc) and the file hierarchy can be reduced (omitting top or bottom levels or parts of the tree). In that manner a user could alternate between making choices and viewing the lattice until the complexity has been sufficiently reduced. One potential application for this might be police investigations of computer harddrives. Currently, it takes the police in Scotland up to three years to investigate a harddrive that has been seized because of suspected criminal files on the computer. An FCA-based tool might help a police officer to investigate and process such harddrives faster. The computer could be booted from a USB Linux drive. The file hierarchy data could then be downloaded and stored in a database and processed as described in Figures 1 and 3.

Apart from a GUI interface that would be required, more pre-processing would be useful. For example, as discussed for the system log, if a logfile contains a re-occurring message, each time with a different identifier, the identifier might need to be omitted. Thus, some heuristic parsing software would be useful. We intend to continue with the experimentation with more types of data (for example, firewall and networking data) and on a variety of different flavours of Unix in order to develop a stable set of heuristics which will then be implemented as a toolkit. We also intend to investigate existing software for logfile processing and data mining in more detail in order to determine whether there are existing tools that can be combined with our methods.

References

1. Aggarwal, G., Bursztein, E., Jackson, C., Boneh, D.: An Analysis of Private Browsing Modes in Modern Browsers. In: Proceedings of Usenix Security (2010)
2. Baoyao, Z.: Intelligent Web Usage Mining (2004)
3. Becker, K., Stumme, G., Wille, R., Wille, U., Zickwolff, M.: Conceptual Information Systems Discussed Through an IT-Security Tool. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 352–365. Springer, Heidelberg (2000)
4. Farley, J.D.: The N.S.A.'s Math Problem. The New York Times (May 16, 2006)
5. Ganapathy, V., King, D., Jaeger, T., Jha, S.: Mining Security-Sensitive Operations in Legacy Code using Concept Analysis. In: Proceedings of the 29th International Conference on Software Engineering (2007)
6. Lindig, C., Snelting, G.: Assessing Modular Structure of Legacy Code Based on Mathematical Concept Analysis. In: International Conference on Software Engineering, pp. 349–359 (1997)
7. Neuhaus, S., Zimmermann, T.: The Beauty and the Beast: Vulnerabilities in Red Hat's Packages. In: Proceedings of the 2009 USENIX Annual Technical Conference (2009)
8. Pohle, C., Spiliopoulou, M.: Building and Exploiting Ad Hoc Concept Hierarchies for Web Log Analysis. In: Kambayashi, Y., Winiwarter, W., Arikawa, M. (eds.) DaWaK 2002. LNCS, vol. 2454, pp. 83–93. Springer, Heidelberg (2002)
9. Priss, U., Old, L.J.: Concept Neighbourhoods in Lexical Databases. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 283–295. Springer, Heidelberg (2010)
10. Priss, U., Old, L.J.: Data Weeding Techniques Applied to Roget's Thesaurus. In: Wolff, K.E. (ed.) KONT/KPP 2007. LNCS (LNAI), vol. 6581, pp. 150–163. Springer, Heidelberg (2011)
11. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer, Heidelberg (1999)
12. Wondracek, G., Holz, T., Kirda, E., Kruegel, C.: A Practical Attack to De-Anonymize Social Network Users. In: 2010 IEEE Symposium on Security and Privacy, pp. 223–238 (2010)
13. Wolff, K.E.: Interpretation of automata in temporal concept analysis. In: Priss, U., Corbett, D., Angelova, G. (eds.) ICCS 2002. LNCS (LNAI), vol. 2393, pp. 341–353. Springer, Heidelberg (2002)

Supporting Ontology Design through Large-Scale FCA-Based Ontology Restructuring

Mohamed Rouane-Hacene, Petko Valtchev, and Roger Nkambou

Department of Computer Science, UQAM
CP 8888, succ. Centre-Ville, Montreal, H3C 3P8, Canada
{rouanehm,nkambou}@gmail.com, valtchev.petko@uqam.ca

Abstract. Ontologies are designed to evolve and this is typically done through sequences of a local modifications in the ontological structure, a.k.a. refactorings. Yet the more complex the structure the less obvious the full impact of such a refactoring. Thus, after a protracted period of maintenance, the overall quality of an ontology may substantially deteriorate. As a remedy, an ontology restructuring task would be performed that cleans its structure and enhances the ontology with new and previously missing entities. We investigate an approach for ontology restructuring based on relational concept analysis (RCA) that allows for a thorough reshuffling of the ontology. Here we present a platform for ontology maintenance, INUKHUK, and illustrate its main workflow dedicated to restructuring. We also report on a preliminary validating study involving several small-to-medium size ontologies.

1 Introduction

Ontologies, especially domain ontologies, are designed to communicate shared conceptualizations among a community and as such their construction is typically a decentralized process [8]. Moreover, a particular emphasis in the ontology languages (e.g., OWL) and editing tools (e.g., *Protégé* [1], the *NeON toolkit* [2], *Ontolingua* [3]) is put on evolution: Ontologies are not carved in stone, they tend to evolve to reflect the extending scope of a conceptualization, the progress in their designers' understanding for the domain or the natural evolution of that domain. Consequently, the typical ontology life-cycle would comprise more or less extensive maintenance and evolution stages which involve changes in various aspects of the ontology infrastructure (concept/property sets, specialization hierarchies on top of these, concept-to-property incidences, etc.) As individual modifications are usually small-scale ones, e.g., moving a property from one concept to another, the underlying design decisions are not always met with the whole picture in mind. Indeed, with rich ontologies whose infrastructure is a complex network of inter-related entities (concepts, properties, individuals, values, etc.) anticipating the impact of a particular modification may prove beyond the designer's skills. Meanwhile, a complete theory about the way such changes, a.k.a. refactorings [5], should be carried

¹ <http://protege.stanford.edu/>

² <http://www.neon-toolkit.org/>

³ <http://ksl.stanford.edu/software/ontolingua/>

out to avoid side-effects is yet to be formulated, let alone validated (first steps are reported in [9]). As a result, changes may deteriorate the overall quality of an ontology by undermining its usability, intelligibility, ease of evolution and even consistency. While inconsistencies would be detected by a formal validation process, more subtle design defects not immediately spoiling the correctness are harder to spot and correct. For instance, the absence of important domain concepts, while not erroneous, could hamper the capacity of an ontology to be easily reused, e.g., through extension and/or adaptation. Often a missing abstraction will be mirrored by specification redundancies in the related (sub-)abstractions, e.g., replicating a common property restriction in the would-be immediate sub-concepts of a missing domain concept thus increasing the necessary effort for their understanding and evolution.

We are interested in the automated support for ontology design and evolution while the focus here is specifically on ontology restructuring. In [18], ontology restructuring has been defined as "the activity of correcting and reorganizing the knowledge contained in an initial conceptual model, and detecting missing knowledge". Following this line of thought, we propose a holistic approach that involves a thorough analysis and re-organization of the ontology structure by means of formal concept analysis (FCA [6]). The corresponding re-shuffling of the ontology amounts to a set of individual refactorings at various spots of the ontology (e.g., merge of existing concepts or factoring out their commonalities into new concept) that are performed in a consistent and synchronous manner. Due to the properties of the underlying mathematical framework, the refactoring set preserves the overall correctness of the ontology while potentially enhancing it with some missing abstractions.

FCA has been successfully used in the past as framework for analysis/restructuring software models [16,4,7] and source code [12]. Its main feature is the detection of potentially useful abstractions on top of a collection of observations - either individuals or abstractions themselves (as in the case of UML class models [11]) - that share a set of descriptors. In modeling terms this corresponds to factoring out common specifications of model entities to produce a new, more abstract entity. A key advantage of FCA over competing approaches is it constructs only abstractions maximizing the amount of shared specifications (a.k.a. formal concepts) and hierarchically orders them according to generality. Moreover, possible *is-a* links between observations are easily reflected in the resulting hierarchy (a.k.a. the concept lattice) which can therefore be straightforwardly translated into a specialization DAG in the original language of observations (e.g., UML). An additional strength, but also a serious issue, is rooted in the capacity of FCA to detect all such abstractions: While there is no need for further explorations outside the lattice, in many practical cases, a large proportion of its concepts may prove of little use for the task at hand (e.g., the overly general ones). As a remedy, filtering mechanisms for concepts based on interestingness measures are applied.

Our work on ontology engineering exploits a variety of FCA, called relational [15,11], as its input format is compatible with the relational data model, i.e., admits both own properties of observations and links between them. When dealing with ontologies and conceptual models in general, such capacity is a key asset: It enables a thorough analysis of the ontology network structure that would otherwise have to be shredded into pieces. In this paper, we present a platform for ontology maintenance, INUKHUK, that

is built on top of a relational concept analysis (RCA) engine (section 4). We present and illustrate a complete workflow for large-scale refactoring, or restructuring, of ontologies within INUKHUK. The results of a preliminary validating study involving several small-to-medium size ontologies are reported as well (section 5). The technical part is preceded by summaries on ontology restructuring (section 2) and on FCA (section 3).

2 The Restructuring Problem

In the following, we relate quality of an ontology to the general quality of models through an example showing some quality criteria of the design. RCA-based refactoring of ontologies in order to restore their quality is addressed in later sections.

2.1 Design Quality Criteria

In the design of models, redundancy is the presence of several definitions of the same element while level of abstraction represents the depth of the generalizations within the various hierarchies of the model elements. As conceptualization, an ontology must satisfy a number of quality criteria such as absence of redundancies in element specifications and appropriate level of abstraction. Both criteria ensure understandability, consistency, ease of evolution and proper use of the ontology.

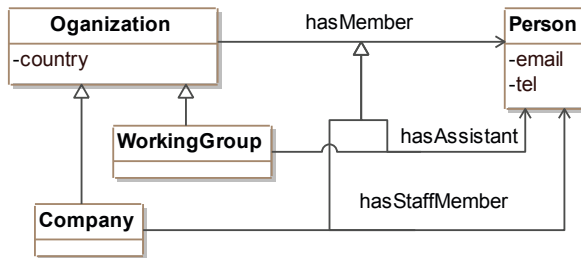


Fig. 1. Reference ontology

For instance, Figure 1 illustrates a fragment of an ontology that shows good design regarding redundancy and level of abstraction. However, the maintenance, which is necessary to keep the ontology up to date by performing various operations including adding/deleting of axioms, adapting, splitting and populating may affect these two aspects of quality. Conversely, restoring quality by restructuring the ontology, i.e., applying a set of refactorings as defined for software models in [5] and adapted for ontologies in [2], is knowingly a challenging task due to the complexity of the ontological structure. Hence the need to design automated tools to support it.

2.2 Refactoring Example

Refactoring aims at reorganizing the ontology, such as lifting a property from a class to a super-class, in order to improve some of the structural aspects while keeping its external functional behavior. As an illustration, consider the ontology depicted in Figure 2

that is distinguished from the ontology given in Figure 1 by the presence of redundancy in specifications and some imperfections in the abstraction level. Indeed, redundancy of specifications is characterized by the `country` property which appears in several classes, whereas incompleteness of abstraction can be noticed by the lack of the abstract class `Organization` that should factorizes the properties that are shared between the classes `Company` and `WorkGroup`. The ontology in Figure 2, called '*initial ontology*', highlights the kind of design defects found in ontologies that are constructed from top or assembled from parts. Moreover, beside the aforementioned defects of redundancy and incompleteness of abstractions, a deeper analysis reveals that relations `hasAssistant` and `hasStaffMember` can also yield an abstraction to link the newly identified class `Organization` to the class `Person` as illustrated by Figure 1.

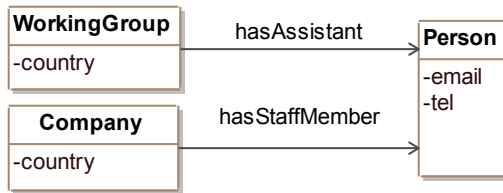


Fig. 2. Initial ontology

Refactoring tasks with the above reorganization prospects are challenging for ontology designers as the creation of new elements and moving of existing ones could have repercussions across the entire ontology. In the re-engineering of object models, several approaches are available that perform large-scale refactoring, including the one based on formal concept analysis (FCA) [6].

3 Theoretical Background on Concept Analysis

Formal Concept analysis (FCA) [6] is an approach for abstracting conceptual hierarchies from sets of objects (e.g., ontology classes) described by properties (e.g., properties.) rooted in lattice-theory. The basic data format in FCA is a cross-table $\mathcal{K} = (O, A, I)$ called formal context, where O is a set of individuals (called formal objects), A a set of properties (called formal attributes) and I the relation has on $O \times A$. For example, the table on top of Figure 3 illustrates a formal context derived from the ontology depicted in Figure 2, where individuals are classes while attributes are class properties.

A pair (X, Y) where X is a maximal set of individuals (called extent) and Y is a maximal set of shared properties (called intent), is called a formal concept. For instance, $(\{\text{WorkingGroup}\}, \{\text{workingGroup}, \text{source:c2}\})$ is a concept (see c_4 in Figure 7). Furthermore, the set of all concepts of the context is partially ordered by extent inclusion and called *complete lattice*. For instance, Figure 7 illustrates concept lattice derived from the context of classes depicted in Figure 3. A concept lattice is drawn as a Hasse diagram (as shown in Figure 7), whereas typically concept intents and extents are compressed to gain in readability. Thus, an object (resp. attribute) is uniquely

Classes	'person'	'company'	'wg'	email	tel
Person	×			×	×
Company		×			
WorkingGroup (WG)			×		

Relations	'ha'	'hsm'
hasAssistant (ha)	×	
hasStaffMember(hsm)		×

Fig. 3. Contexts encoding classes and relations of the ontology depicted in Figure 2. The attribute country was removed to show the strength of RCA method.

shown at the minimal (resp. maximal) concept node whose extent (resp. intent) features it. Upwards/downwards inheritance between concepts is used to retrieve their full extents/intents. For instance, the full labeling of concept c_5 is $(\{\text{WorkingGroup}, \text{Company}\}, \{\text{source}:c_5\})$.

Lattice constructed on top of a set of object classes are typically interpreted as class hierarchy: concepts are seen as abstract classes and their *subconcept-of* relationships as generality links. For instance, c_5 factors out commonalities between c_4 (class `WorkGroup`) and c_3 (class `Company`). Hence $c_{\#5}$ represents an abstraction of both classes, e.g., the previously identified `Organization` class. However, without the presence of the attribute `country`, the lattice could not suggest creating the class `Organization` yet its presence in the refactored ontology is necessary to provide a common domain for relations `hasAssistant` and `hasStaffMember`. Here comes the need to abstract from various types of entities, in particular ontological relations.

Relational Concept Analysis (RCA) [11] was introduced to support FCA in extracting formal concepts from sets of individuals described by local properties and links (as illustrated in Figure 6). We will describe the RCA data of and its analysis process through its application to the refactoring of ontologies in the next section.

4 INUKHUK: Service Oriented Platform for Maintaining Ontologies Using RCA

We have started INUKHUK, an open source platform, which aims to provide a set of infrastructure services for ontology engineering. For instance, the services allow the construction of domain ontologies from text or through the assembly of a variety of semantic data, including ontology modules. In the latter case, additional services are available including modularization, merging and refactoring of ontologies. Figure 4 illustrates the use cases of the proposed platform.

4.1 Overall Process of Refactoring

To detect and remediate imperfections in the design of ontologies, INUKHUK relies on RCA framework. The global scenario of refactoring includes several steps including *alignment*, *encoding*, *analysis*, *reverse encoding* and *validation* of the refactored ontology.

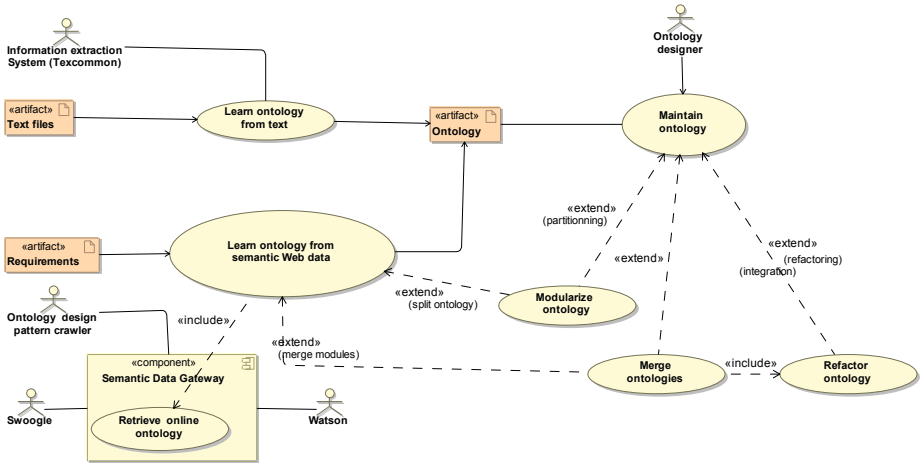


Fig. 4. Inukhuk platform services

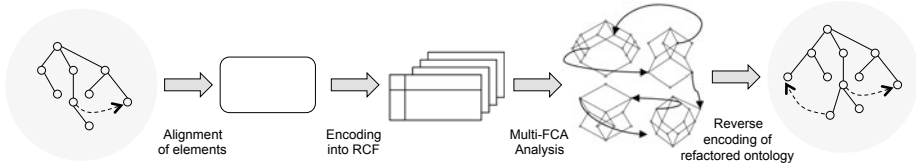


Fig. 5. Refactoring workflow

4.2 Aligning Ontology Elements

The alignment stage determines correspondences among ontology elements. This will avoid duplication in the translation of an ontology in the RCA data format by repeating similar specifications. The resolution of some naming conflicts takes place at this step. Name conflicts are resolved through name normalization based on linguistic resources such as electronic dictionaries and specialized domain ontologies. We have implemented ONTOALIGN, a tool that relies on third party packages including java-based api ALGINAPI⁴ and SIMPACK⁵ to detect potential name conflicts in the initial ontology.

4.3 Modelling the Data

In RCA input data are organized within a structure called relational context family (RCF) [11] that comprises a set of binary contexts $\mathcal{K}_i = (O_i, A_i, I_i)$ and set of binary relations $r_k \subseteq O_i \times O_j$, where O_i and O_j are the individual sets of \mathcal{K}_i (domain)

⁴ <http://alignapi.gforge.inria.fr/>

⁵ <http://www.ifi.uzh.ch/ddis/simpack.html>

source		
	'ha'	'hsm'
Person		
Company		×
WG	×	

target		
	'ha'	'hsm'
Person		
Company		×
WG	×	

dom			
	'person'	'company'	'wg'
ha			×
hsm	×		

ran			
	'person'	'company'	'wg'
ha	×		
hsm	×		

Fig. 6. The inter-context relations encoding incidence between ontological classes and ontological relations of Figure 2

and \mathcal{K}_j (range), respectively. Hence, during the encoding step, the initial ontology elements are transformed into a unique RCF where each context \mathcal{K}_i correspond to a sort of entity in the ontology meta-model, i.e., classes and properties. Moreover, the incidences between meta-entities (links between entity sorts in the ontology meta-model) are translated into binary relation r_k in the RCF, e.g., the relation `dom` in Figure 6. For sake of clarity of illustrations, we limit the encoding of our running example to context of classes and context of ontological relations. The class properties are encoded as formal attributes into the context of classes. Moreover, names of classes (resp. relations) are added as formal attributes in the underlying context and used to represent the inheritance links between classes (resp. relations). The encoding of the initial ontology depicted in Figure 2 is jointly represented by Figure 3 and Figure 6.

4.4 Analysis of the Relational Data

The input RCF is fed into the RCA engine whose respective output is a family of relational lattices (RLF). There is a lattice for each context in the RLF which is constructed step-wise and by a simultaneous expansion of all the member lattices (and their corresponding contexts). The process starts with all the initial lattices of the contexts, i.e., lattices reflecting uniquely non-relational descriptors of the formal objects. At each subsequent step, the attribute set of a context from the RCF will be completed with some new attributes, reflecting the relational links to objects from other contexts as well as the already known concepts on those contexts (e.g., the attribute `source:c5` linking objects representing ontology classes to a concept from the property ontology in Figure 8). The underlying attribute generation mechanism is called *relational scaling*. It is performed on every subsequent version of the lattices from the RLF. As scaling basically enhances object descriptions, a new concept construction step puts the RLF in line with the current state of the RCF. This in turn may (but need not) increase the total set of concepts and trigger a new scaling step. The analysis stops whenever the scaling fails to impact the lattice structure, i.e., the newly synthesized attributes do not induce

new concepts in any of the RLF lattices. In other terms, a stop means a fixed point is reached that materializes into an isomorphism between lattices at step $i + 1$ and their counterparts at step i (For more details, see [11]).

It is noteworthy that formal concepts in the final lattices are connected by the newly defined attributes much the same way description logics concepts are connected by property restrictions involving logical quantifiers (here existential quantification is implicit in the attribute construction). Figure 7 and Figure 8 show the final lattices of our running example.

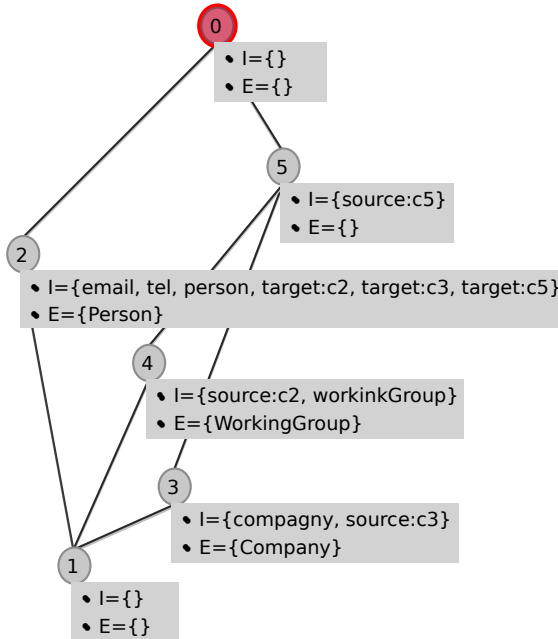


Fig. 7. Final lattice of classes

4.5 Generating the Refactored Ontology

An ontology is generated from the RLF by filtering out potentially useful abstractions and the reverse encoding of the corresponding ontological elements. Filtering is very important as interpreting the entire lattices as hierarchies (classes and relations) creates models of large size which contain irrelevant elements. The tool ONTODESIGNER is component of INUKHUK platform devised to the generation of ontologies based on lattices. It implements a set of pruning functions that discard spurious concepts from the lattices. The underlying mechanism assesses the relevance degree of each concept using criteria such as the presence of key domain information in the corresponding intent, e.g., class attribute. ONTODESIGNER operates model transformation based on meta-models. The input model is RLF, while output model is an ontology encoded in an ontology language, e.g., OWL. From the RLF whose lattices are depicted in Figure 7 and Figure 8, respectively, the generated ontology corresponds to the one in Figure 1.

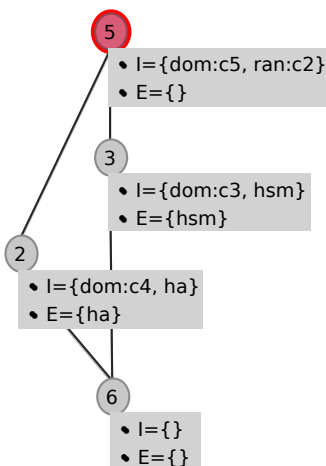


Fig. 8. Final lattice of roles

5 Implementing and Validating the Approach

5.1 Implementation

We have developed INUKHUK as set of services and tools for conducting refactoring of ontologies. INUKHUK is coupled with the RCA platform GALICIA⁶ and several open-source NLP resources including WORDNET⁷ and ANNIE⁸, the GATE-based information extraction system. Multi-tool open-source platform GALICIA provide RCA framework, while GATE is used to deal with linguistic aspects of encoding of the ontology into the RCA input data format. As part of INUKHUK, we have developed OPENQAM, an open-source suite of metrics for ontology quality analysis. Finally, we have implemented a semi-automatic process that aims at generating a poorly designed ontology based on the reference ontology by introducing redundancy of specification elements and incompleteness of abstraction.

5.2 Experiments

Several medium-sized ontologies hosted in *Protégé* repository have been considered for refactoring. Table 1 gives brief statistics on these ontologies.

OPENQAM implements several metrics that can be used to assess the structural quality of the design, such as the measures of Alani *et al.* [11] which rank a collection of ontologies according to a set of terms that usually form queries supported by these ontologies. In the problem of refactoring, the collection consists of the reference, initial and refactored ontologies. Alani *et al.* metrics include class match measure (CMM), density measure (DEM), semantic similarity measure (SSM), and betweenness measure (BEM). CMM is meant to evaluate the coverage of an ontology for the given

⁶ <http://sourceforge.net/projects/galicia/>

⁷ <http://wordnet.princeton.edu/>

⁸ <http://gate.ac.uk/ie/annie.html>

Table 1. List of ontologies used for validation of the approach

Name	Concept count	Object prop. count	Data prop. count	Individual count	Expressivity of DL
Tourism	78	26	27	57	SHIN (D) http://www.bltk.ru/OWL/tourism.owl
Travel	35	06	04	14	SOIN (D) http://protege.cim3.net/file/pub/ontologies/travel
People	60	14	1	21	ALCHIN (D) http://protege.stanford.edu/download/ontologies.html

search terms. DEM is intended to approximate the information-content of classes and consequently the level of knowledge detail (the number of subclasses, the number of attributes, number of siblings, etc.). SSM calculates how close the classes that matches the search terms are in an ontology. BEM measures the betweenness value of each queried class in the given ontologies. This value is calculated based on the number of shortest paths that pass through the class in the ontology graph.

The outcome experimental of our experiments as listed in Figure 9) draws some remarks: (i) class match measure CMM of initial ontologies is always higher. This is because the refactored ontology contains more abstractions than the initial ontology so that the rate of known terms in the initial ontology is high. According to our validation protocol these terms are obtained from reference ontology. An appropriate naming of new abstractions would reverse the situation. (ii) values of information-content DEM and betweenness BEM have both decreased since refactoring change significantly the number of links, graph paths, etc., whose quality is exclusively assessed by a domain expert, the only one able to capture the associated semantics. (iii) After refactoring, it may happen that the initial ontology remains closer to the reference ontology, than the refactored one. Such an outcome is particularly plausible when the reference ontology has some structural defects that are naturally corrected by RCA. This explains why the centrality measure SSM that is measured by the minimum number of links required to connect a pair of classes has decreased.

As illustrated by the table in the right-hand side of Figure 9, the values of some metrics have decreased in the restructured ontology w.r.t. its initial counterpart. This may happen in many cases. For instance, the initial ontology contains structural imperfections while refactored ontology has a better design since it is restructured by the tool and validated by the expert. The latter can remove structural elements from the refactored ontology that are not relevant and therefore influence some metrics including CMM which counts the number of similar terms in both ontologies.

For instance, Figure 10) shows a fragments of the refactored Tourism ontology. RCA-based refactoring process suggests to abstract the three restrictions `has_passport`, `has_int_passport`, and `has_ticket`, by the new restriction `C32` to which we have assigned the name 'has_travel_document'. The new restriction `has_travel_document` links `Tourist` to its travel documents (concept `Travel_document`) such as `passport` and `ticket`. The restriction `has_travel_document` is obtained by factorizing domain and range of subrestrictions `has_passport`, `has_ticket`, etc. Feeding the RCA process with additional features describing restrictions such as multiplicity and

Ontology	Concept count	Object property count	Data property count	Tourism		Travel		People	
				\mathcal{O}_I	\mathcal{O}_R	\mathcal{O}_I	\mathcal{O}_R	\mathcal{O}_I	\mathcal{O}_R
Tourism	99	63	27	1.0	.736	1.0	.725	1.0	.894
Travel	38	10	04	.393	1.0	.995	1.0	1.0	.971
People	64	16	1	1.0	.349	1.0	.07	1.0	.074
BEM				1.0	.08	1.0	.03	1.0	.05

Fig. 9. Left: Size of the refactored ontology. **Right:** Refactored (\mathcal{O}_R) v.s. initial ontology (\mathcal{O}_I).

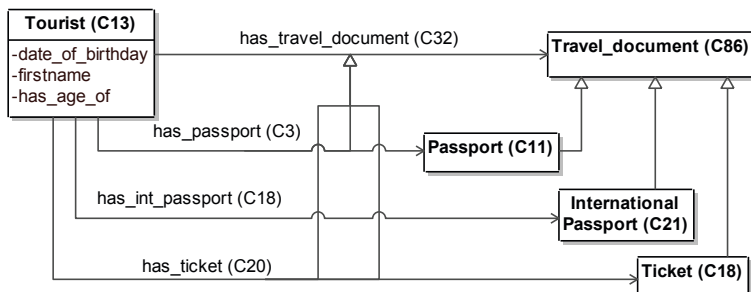


Fig. 10. The discovery of new restrictions by the tool for the Tourism ontology

direction necessarily lead to the discovery of further commonalities that can be represented through new potentially useful abstractions.

6 Related Work

Several studies have used FCA as part of a process of ontology engineering. Stumme *et al.* [17] have explored ontology construction by merging two existing ontologies provided with a corpus of textual documents. NLP techniques are used to capture the relationships between documents and classes from an ontology and organize them into a dedicated context. The two contexts are then merged and a pruned concept lattice is constructed which is further transformed into a merged ontology by a human expert.

In [13], Nanda *et al.* introduced a FCA-based methodology for the design of formal ontologies for product families. Terms describing the components in a product family along with the required properties are captured in a lexicon set and put into context. A class hierarchy is then derived from the concept lattice and exported into OWL. Unlike our own approach, the entire process relies heavily on human involvement while proposing no specific assistance for the extraction of transversal relations.

Several refactoring alternatives have been proposed. For instance, [2] parses ontologies using rules to detect anomalies. The parsing consists in investigating subset of OWL-DL that includes rules, is-a relations, and some property features, e.g., transitivity, disjunction, etc. Discovered anomalies includes syntactic inconsistency, redundancy of definitions, and cycles in definitions of rules and taxonomy. Each anomaly is provided with a formal definition that facilitates its detection in the ontology together with refactoring actions, e.g., creation of new class or property, change rules with abnormalities, etc.

Ostrowski *et al.* [14] have proposed a dynamic approach of refactoring that is based on $\langle \text{TBOX}, \text{ABOX} \rangle$ model of \mathcal{DL} ontologies as well as the set of target changes to be carried on the ontology. The required changes guide the definition of a set of transformation rules, e.g., remove of ontological components, inference of new classes, etc.

Conesa *et al.* [3] have proposed a semi-automatic method for pruning a general ontology that involves three steps: refinement, pruning and restructuring. The refinement step completes the ontology with new concepts while pruning discards irrelevant ones. The method applies a set of restructuring operations similar to those used in software refactoring.

In [19], authors have compared the result of mapping a pair of refactored ontologies with the result of mapping of the initial version of these ontologies. The proposed technique is able to reveal certain ontological defects based on the detection of patterns of names in the structure of the ontology. Several refactoring operations can be applied including adding and renaming concepts.

The work in [10] describes an approach for restructuring ontologies through knowledge discovery (ROKD). This approach is based on cooperative agent architecture where agents support (i) the discovery of preference information from query results and user profiles using data mining techniques; and (ii) the application of restructuring mechanisms to personalize the existing ontologies in a specific user ontology based on his preferences.

7 Conclusion

We have addressed the problem of refactoring a domain ontology. A refactoring approach and tools were proposed that relies on RCA framework. RCA framework supports concept analysis in mining relational data. Preliminary validation experiments conducted on various ontologies have demonstrated that the approach suits ontology refactoring that aims to improve the generalization level in class/relation hierarchies which would be one step further towards the quality assurance of ontology-based systems. Thus, these systems will rely on a stronger semantics.

RCA-based approach is accurate, exhaustive and formally founded. However, there are a number of limitations on both NLP and FCA parts of the approach. Indeed, the use of imprecise and incomplete linguistic resources for detecting similarities in class, relation and property names creates spurious abstractions that are hard to interpret. Moreover, the large size of the derived lattices may compromise the scalability of the approach. Our intuition is that pruning can improve the results. However, future work should focus on the design of effective pruning function.

References

1. Alani, H., Brewster, C.: Metrics for ranking ontologies. In: 4th Int. EON Workshop, 15th Int. WWW Conf. (2006)
2. Baumeister, J., Seipel, D.: Verification and refactoring of ontologies with rules. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 82–95. Springer, Heidelberg (2006)

3. Conesa, J., Olivé, A.: A method for pruning ontologies in the development of conceptual schemas of information systems. In: Spaccapietra, S., Atzeni, P., Chu, W.W., Catarci, T., Sycara, K. (eds.) *Journal on Data Semantics V. LNCS*, vol. 3870, pp. 64–90. Springer, Heidelberg (2006)
4. Dao, M., Huchard, M., Rouane-Hacene, M., Roume, C., Valtchev, P.: Improving Generalization Level in UML Models: Iterative Cross Generalization in Practice. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) *ICCS 2004. LNCS (LNAI)*, vol. 3127, pp. 346–360. Springer, Heidelberg (2004)
5. Fowler, M.: *Refactoring: improving the design of existing code*. Addison-Wesley, Boston (1999)
6. Ganter, B., Wille, R.: *Formal Concept Analysis. Mathematical Foundations*. Springer, Berlin (2001)
7. Godin, R., Valtchev, P.: Formal concept analysis-based class hierarchy design in object-oriented software development. In: Ganter, B., Stumme, G., Wille, R. (eds.) *Formal Concept Analysis. LNCS (LNAI)*, vol. 3626, pp. 304–323. Springer, Heidelberg (2005)
8. Gomez-Perez, A., Corcho, O., Fernandez-Lopez, M.: *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, 1st edn. *Advanced Information and Knowledge Processing*. Springer, Heidelberg (July 2004)
9. Grllner, G., Staab, S.: Categorization and recognition of ontology refactoring pattern. Technical Report 9/2010, University of Koblenz-Landau (2010)
10. Henshaw, S.M., El-Masri, A., Sage, P.A.: Restructuring ontologies through knowledge discovery. In: 5th IEEE Conference on Enterprise Comp., E-Commerce and E-Services, pp. 441–444. IEEE Computer, Washington, DC (2008)
11. Huchard, M., Rouane-Hacene, M., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. *Annals of Math. and AI* 49(1), 39–76 (2007)
12. Moha, N., Hacene, A.M.R., Valtchev, P., Guéhéneuc, Y.-G.: Refactorings of design defects using relational concept analysis. In: Medina, R., Obiedkov, S. (eds.) *ICFCA 2008. LNCS (LNAI)*, vol. 4933, pp. 289–304. Springer, Heidelberg (2008)
13. Nanda, J., Simpson, T., Kumara, S.R.T., Shooter, S.B.: A methodology for product family ontology development using concept analysis and web ontology language. *Journal of JCISE* 6, 103–113 (2006)
14. Ostrowski, D.A.: Ontology refactoring. In: *International Conference on Semantic Computing*, pp. 476–479 (2008)
15. Rouane-Hacene, M., Huchard, M., Napoli, A., Valtchev, P.: A proposal for combining formal concept analysis and description logics for mining relational data. In: Kuznetsov, S.O., Schmidt, S. (eds.) *ICFCA 2007. LNCS (LNAI)*, vol. 4390, pp. 51–65. Springer, Heidelberg (2007)
16. Snelting, G., Tip, F.: Understanding class hierarchies using concept analysis. *ACM Transactions on Programming Languages and Systems* 22(3), 540–582 (2000)
17. Stumme, G., Maedche, A.: FCA-merge: Bottom-up merging of ontologies. In: *Proc. of IJ-CAI*, pp. 225–234 (2001)
18. Surez-Figueroa, M.C., Gomez-Prez, A.: First attempt towards a standard glossary of ontology engineering terminology. In: 8th International Conference on Terminology and Knowledge Engineering, TKE 2008 (2008)
19. Sváb-Zamazal, O., Svátek, V., Meilicke, C., Stuckenschmidt, H.: Testing the impact of pattern-based ontology refactoring on ontology matching results. In: *Proc. 3rd Intl. WS on Ontology Matching (OM-2008)*, Karlsruhe (DE). *CEUR WS Proc.*, vol. 431 (2008)

Towards a Formalization of Individual Work Execution at Computer Workplaces

Benedikt Schmidt¹, Heiko Paulheim¹, Todor Stoitsev¹, and Max Mühlhäuser²

¹ SAP Research Darmstadt,
Bleichstrasse 8, 64285 Darmstadt, Germany
`{firstname.lastname}@sap.com`

² Technische Universität Darmstadt, Telecooperation Group
Schloßgartenstr. 7, 64289 Darmstadt, Germany
`{firstname}@informatik.tu-darmstadt.de`

Abstract. To better understand, analyze, and support work execution at computer workplaces, this paper presents a framework of ontologies. We analyze knowledge work at computer workplaces as weakly-structured processes by means of activity theory. Based on the analysis, we extend a set of upper ontologies to model the computer workplace and the process of work execution. We especially reflect the process of tool selection involved in work execution by a hierarchical analysis of involved planning activities and software tools, enabling a plan realization.

1 Introduction

Individual work execution processes in knowledge work are complex and weakly structured, i.e., they allow a large number of variations of the individual process steps and their execution order. A relevant environment of individual work execution processes is the computer workplace. This paper addresses the formalization of work execution at computer workplaces, i.e. the execution of regular office work.

Computer workplaces are multi-purpose workplaces, providing a set of software applications with numerous functionalities to enable and support a large variety of information creation and consumption activities. Individuals blend the use of software applications in individual processes of work execution which manifest expertise as well as experience. Description, analysis and support of such individual execution processes is a difficult task, as it comprises consideration of the software landscape with its capabilities – a highly structured domain – as well as the planning and execution of the work process – a weakly structured and highly individualized process.

The remainder of this paper is structured as follows. We give an overview of the requirements for an ontology of the domain and introduce a running example that will illustrate our work throughout the paper. The third section analyzes work execution processes at computer workplaces in terms of activity theory (AT). To formalize the processes, we extend the DOLCE upper ontology that

is introduced in Sect. 4. The original contribution of this paper, the computer work ontology (CWO) is presented in Sect. 5:

- Formalization of the computer workplace
- Formalization of work execution processes and tool selection

Finally, we review related ontologies and conclude with a summary and an outlook on future work.

2 Running Example and Requirements

Although the approach supports the formalization of arbitrary work execution processes at computer workplaces, we provide one running example that illustrates the core aspects of our work and that is used throughout the paper to illustrate formalizations.

2.1 Running Example: Pete Plans a Conference Travel

Pete works at the research department of a software company. Pete wants to visit the ICCS11 conference. Pete's original motive of visiting the conference is a strong interest in conceptual structures. Therefore, he has the objective to attend the ICCS11, which requires the creation of a travel request to be confirmed by his manager. The travel request is a standardized form which requires different information, e.g. travel destination, travel duration, and approximate costs for flight and hotel. Pete needs to find the travel request form, identify all required information, fill them into the form, and provide his manager with the filled out form. All these activities are performed on a windows PC with an office suite. To execute the task, Pete executes the following process:

- Browse for the document $\xrightarrow{\text{solve\textit{d}by}}$ Use the Windows Explorer to identify the form document by opening different folders (Step1).
- Consume the authoring form document $\xrightarrow{\text{solve\textit{d}by}}$ Open the form document with Microsoft Excel, as it is an .xls, file and identify required information (Step2).
- Browse for required information, conference data $\xrightarrow{\text{solve\textit{d}by}}$ Open a web browser and use a search engine to access the conference website (Step3).
- Author form document, conference data $\xrightarrow{\text{solve\textit{d}by}}$ Use copy and paste to transfer different information from the conference website to the form (Step4).
- Browse for required information, hotel and flight costs $\xrightarrow{\text{solve\textit{d}by}}$ Use a web browser to access web pages to book flights and hotels and identify approximate costs (Step5).
- Author form document, hotel and flight costs $\xrightarrow{\text{solve\textit{d}by}}$ Type identified costs into the respective input fields of the form document (Step6).
- Communicate the filled out form $\xrightarrow{\text{solve\textit{d}by}}$ Start Outlook, create an email and attach the filled out request form (Step7).

In the remainder of this paper, we will use this example process to illustrate our concepts and illustrate the ontology. Since we are dealing with weakly structured processes, however, we have to point out that this only one out of the numerous valid execution paths which exist due to individual requirements and selections of available functionality.

2.2 Modeling Requirements

The example illustrates a work execution process at the computer workplace and gives us some indications of what concepts a useful formal description may contain. The process of individual requirement generation and the selection of appropriate software to solve the requirements is of importance. This will facilitate the modeling of execution processes and may also be used to facilitate work execution by improved user support: proposing software functionalities and resources.

Altogether, the example indicates the following required aspects:

1. Individual work execution: Information about individuals, individual goals and the categories which are used to describe and execute work is required.
2. Tool/artifact: Software and its capabilities on an abstract and a functional level needs to be modeled. Information handling needs to be a specific focus with respect to the structured encoding systems and the content.
3. Connection between individual execution and tools/artifacts: As the individual execution process is constrained by usable tools and artefacts, the connection between both needs to be modeled.

Following [11], we define four additional requirements for the formalization: We need to (1) avoid conceptual ambiguity, (2) axiomatize our concepts, (3) avoid concepts that have no ontological meaning but exist for modeling reasons only and (4) provide the concepts without limiting future extensions of the ontology.

3 Activity Theory as Perspective on Work Execution

Work execution at computer workplaces can be explained by means of Activity Theory (AT) [7]. AT provides concepts to describe the interaction of an individual with the environment. In the following, we give a quick overview on concepts of AT which are relevant in the context of this paper. Then, we apply these concepts to the domain of work execution at the computer workplace.

3.1 Activity Theory and Situated Behavior

AT uses activities to provide a minimal meaningful context for human action. Thereby, an activity is a form of doing directed to an object. The relation of a subject to an object is mediated by a tool. The tool condenses the historical development of the relationship between subject and object. The role of tools shows the need to consider artifacts as integral and inseparable components of human functioning.

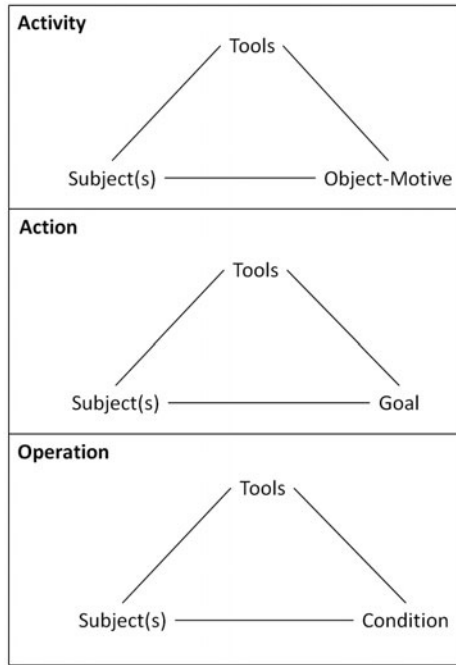


Fig. 1. Hierarchy of Behavior Situatedness in Activity Theory

Based on the activity, AT organizes situated behavior in a hierarchy, decomposing activities to actions and operations. Each element of the hierarchy is a maximally connected triad (see Fig. 1). Activities as long-term formations cannot be transformed directly into outcomes, but through a process. Thereby, activities are realized by a set of individual and cooperative actions. These actions are related to each other by the object/motive of the enclosing activity. Actions are executed by a set of operations, which are routines used subconsciously as answers to conditions faced during the performance of the action [6].

The borders between the levels of the hierarchy are permeable. Actions transform to operations, once the subject has gained experience to execute the action subconsciously. In case of changing conditions, the operation may transform back into an action. An activity can become an action, once a motive is lost, or the goal of an action may turn out as being a motive, transforming the action to an activity.

3.2 Situated Behavior at Computer Workplaces

Considering the computer workplace, situated behavior presents is constrained by the capabilities of the computer as tool. The computer transforms signs and has established as a tool for supporting the consumption, creation and transformation of data as information. Precisely, all behavior is channeled through software tools with respective functionalities.

As described above, AT mediates activities, actions and operations by tools. Work at a computer workplace as an individual and weakly structured process involving different applications to generate a specific outcome can be considered as action. We do not consider weakly structured computer work as an activity, as the limited execution time and the precise outcome does not fit the long-term motive perspective. On the other hand, we do not consider weakly structured computer work as an operation, as the execution is a knowledge-intensive act with characteristics of problem-solving, thus improbably unconsciously executed.

Considering weakly structured work at computer workplaces as an action does not provide information about the way execution processes develop. In the given example (see Sect. 2), the main goal of requesting a travel by the manager was decomposed into different subgoals. Each subgoal was executed by the functionality of a software application. We propose the decomposition of computer work actions into four categories that are connected in the sense of a hierarchical decomposition:

- **Task process:** The first level in a computer work execution hierarchy is the process of planning and executing a task. We follow a functional understanding of task as a logical unit of work that is performed by a series of actions in pursuit of a certain aim [8,15].
- **Knowledge action:** The second level in the hierarchy stands for the initial decomposition of the task. The subject identifies the main challenges of the task and tries to address these challenges by patterns of problem solving. We call these problem solving patterns *knowledge actions*, following existing works on reoccurring problem solving tasks in knowledge work [5]. Relevant knowledge actions in the domain of computer workplaces are 1) Browsing, 2) Consuming, 3) Authoring, 4) Communicating, and 5) Organizing.
- **Application action:** The third level in the hierarchy stands for execution activities in the context of a software and the identification of sets of functionalities that need to be performed to execute the knowledge action.
- **Desktop operation:** The fourth and lowest level of a computer work execution hierarchy stands for single functionalities that can be accessed within the context of a software and thus can be realized directly by an operation.

4 Modeling Basis

While domain ontologies focus on a minimal terminological structure, upper ontologies describe general concepts which are valid across all knowledge domains. The ontology we present in this paper is an extension of the DOLCE ontology.

DOLCE has been designed as a first module of a foundational Ontologies Library [3,9]. As an upper ontology, DOLCE describes relationships between enduring and perdurant particulars. Endurants are independent essential wholes that are in time, while lacking temporal parts, e.g. this paper. Perdurants, on the contrary, are entities that happen in time, and can have temporal parts, e.g. the process of reading this paper.

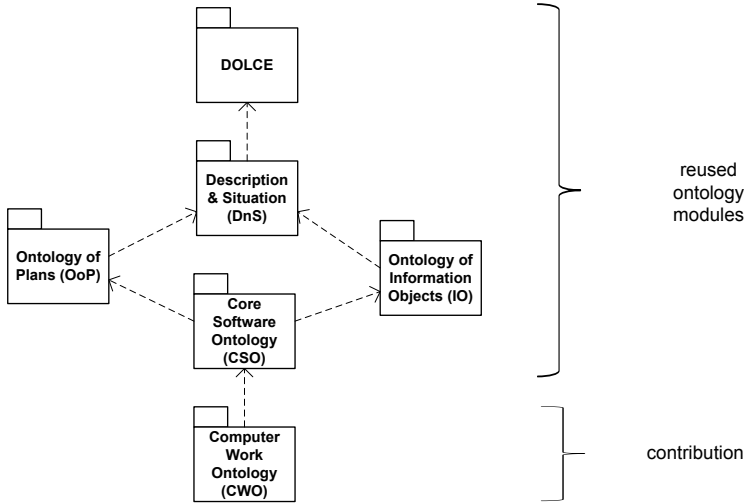


Fig. 2. Overview of the ontologies. Dotted lines represent dependencies between ontologies. An ontology O_1 depends on O_2 if it specializes concepts of O_2 , has associations with domains and ranges to O_2 or reuses its axioms.

The library of ontologies contains different systematically related modules and defines different design patterns to reuse the content for more specific domains [2]. For this paper, we reuse the following ontologies (c.f. Fig. 2):

- Descriptions and Situations (DnS): An ontological theory of contexts. DnS can be considered an ontology design pattern for structuring core and domain ontologies that require contextualization.
- Ontology of Plans (OoP): Formalization of a generic theory of plans.
- Ontology of Information Objects (IO): A semiotic ontology design pattern that assumes a content transferred in any modality to be equivalent to a social object called information object.
- Core Software Ontology (CSO): Formalization of fundamental concepts in the computer domain, e.g. software or data.

5 Computer Work Ontology (CWO)

The CWO extends the DOLCE ontology with respect to the domain of individual, weakly-structured desktop work. We present the CWO in terms of tool, subject, and object and describe their relationship as given in the AT triade. The tool is the computer workplace environment, which is described as an environment for the transformation of information by functionalities. The subject is the knowledge worker, who plans and executes work at a computer workplace. The object is the goal of the individual work.

5.1 Tool: Computer Workplace Environment

We model the computer workplace as an environment that offers functionalities of generating, displaying and transforming data which can be consumed as information. The functionalities and the available information defines a possibility-space for the execution of work. Functionalities are encapsulated in software tools and information is stored in files.

Software and Functionalities. To model software, we use the respective design pattern as described in [11]: `CSO:Software`¹ is defined as `CSO:Data` that `OIO:expresses` an `OoP:Plan`, itself sequencing a set of `OoP:Task` (see Fig. 3). We are interested in a perspective on software, as it is available to end-users. The functionalities offered by the software are modeled as `CWO:Functionality`, a specialization of `OoP:Task`. To describe the plans, describing the purpose-of-use of a software (e.g. word processing), we model `CWO:Scenario` as specialization of `OoP:Abstract-Plan`. The `CWO:Scenario` sequences a set of `CWO:Functionality`.

- (D1) $CSO:Functionality(x) =_{def} OoP:BagTask(x)$
 $\wedge \exists y (DOLCE:part-of(y,x) \wedge ComputationalTask(y))$
- (D2) $Scenario(x) =_{def} OoP:Abstract-Plan(x)$
 $\wedge \forall y (DnS:defines(x,y) \rightarrow Functionality(y))$
- (D3) $CSO:Application(x) =_{def} CSO:Software(x)$
 $\wedge \exists y (OIO:realizedBy(x,y) \wedge CSO:ComputationalObjects(y))$
 $\wedge \forall z (OIO:expresses(x,z) \rightarrow Scenario(z))$

As an example, we model the Windows Explorer functionality to open folders and display files, that was used to identify the form document. We relate a scenario with the software and assign different functionalities to the scenario.

- (Ex1) `CSO:Software(windowsExplorer)`
- (Ex2) `Scenario(folderStructureInteraction)`
- (Ex3) `Functionality(browseFolderStructure)`
- (Ex4) `Functionality(getElementDetails)`
- (Ex5) `Functionality(executeElementWithApplication)`
- (Ex6) `OIO:express(windowsExplorer, folderStructureInteraction)`
- (Ex7) `DnS:defines(folderStructureInteraction, browseFolderStructure)`
- (Ex8) `DnS:defines(folderStructureInteraction, getElementDetails)`
- (Ex9) `DnS:defines(folderStructureInteraction, executeElementWithApplication)`

Information Objects Represented by Files. Files realize a connection between meaningful information and software by data in a digital encoded representation. We model a `CWO:File` as a role played-by only `CSO:Data`. As `CSO:Software` is a subclass of `CSO:Data`, we cover software as files (see Fig. 3). `CSO:AbstractData` is another subclass of `CSO:Data`, containing data that identifies something different from itself, e.g., the word *tree* that stands for a mental

¹ Throughout the paper entities that belong to `CWO` are given without prefix. For all other entities, the respective prefix is given.

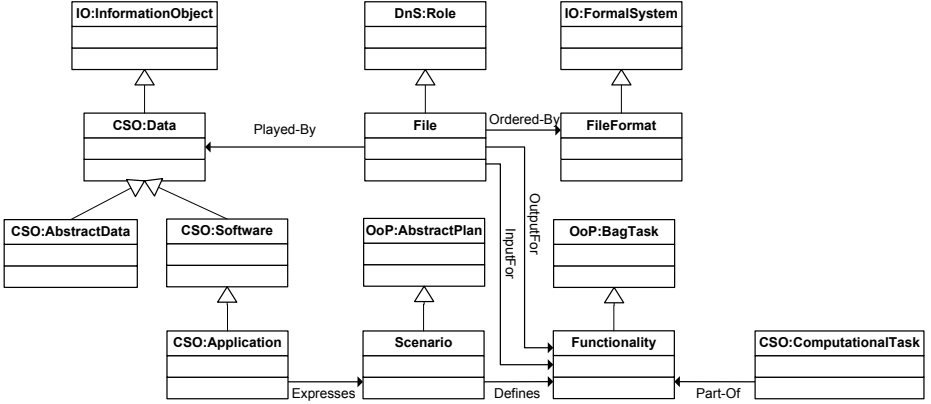


Fig. 3. The classification of software with scenarios, functionalities, and files. Concepts taken from DOLCE and accompanying ontologies are labeled with the respective name space.

image of a real tree. As a file may be abstract data or software, two aspects of files are supported: 1) being a static information object 2) being an information object for execution to make plans accessible in a runtime representation. A file as a static information object is modeled by relating the file as CSO:Data by DnS:about with a DnS:description. A file as an executable information object relates CSO:Software with OoP:Plan by the DnS:expresses relation.

A CWO:File is DnS:ordered-by a CWO:File-Format. A CWO:File with specific CWO:File-Formats can be input for CWO:Functionality. This connection organizes the file access by functionalities, which may range from opening the file to display content in a work processor to the interpretation of a web page by a web browser.

- (D4) $\text{File-Format}(x) \rightarrow \text{IO:Formal-System}(x)$
- (D5) $\text{specializes}(x,y) \wedge \text{File-Format}(x) \rightarrow \text{File-Format}(y)$
- (D6) $\text{uses}(x,y) \wedge \text{File-Format}(x) \rightarrow \text{File-Format}(y)$
- (D7) $\text{File}(x) =_{def} \text{DnS:Role}(x) \wedge \exists y(\text{ordered-by}(x,y) \wedge \text{File-Format}(y))$
 $\wedge \exists z(\text{played-by}(z,x) \wedge (\text{AbstractData}(z) \vee \text{Software}(z)))$
 $\wedge \forall f(\text{inputFor}(x,f) \rightarrow \text{Functionality}(f))$
 $\wedge \forall g(\text{outputFor}(x,g) \rightarrow \text{Functionality}(g))$

In the following, we give two examples for using CSO:File. The first example is for a file as role played by CSO:Data that is not CSO:Software. This means, the aspect of being an information object is of prime importance. For this purpose we model the form document which is identified in step 1 of the example (see Sect. 2) and show the connection to a word processor.

- (Ex10) $\text{IO:Information-Object}(\text{document-for-travel-request})$
- (Ex11) $\text{DnS:description}(\text{travel})$
- (Ex12) $\text{DnS:about}(\text{document-for-travel-request}, \text{travel})$

- (Ex13) File(TravelRequest.docx)
- (Ex14) DnS:played-by(Document-for-travel-request, travelRequest.docx)
- (Ex15) File-Format(docx)
- (Ex16) DnS:ordered-by(travelRequest.docx, docx)
- (Ex17) CSO:Software(microsoftWord)
- (Ex18) Scenario(textProcessing)
- (Ex19) DnS:expresses(microsoftWord,textProcessing)
- (Ex20) Functionality(openTextFile)
- (Ex21) DnS:defines(textProcessing, openTextFile)
- (Ex22) DnS:inputFor(openTextFile, travelRequest.docx)

The second example is for a file as a role played-by CSO:software. This means that the file as software gives access to functionalities. An interesting example are web applications interpreted from the perspective of a user. For a user, a web application is an address to be typed into a browser. By focusing on this aspect of consumption, the web application is a software that plays the role of a file. We use step 5 of the example (see Sect. 2), which is opening and interacting with a web application to book hotels.

- (Ex23) CSO:Software(hotelBooker)
- (Ex24) Scenario(searchHotel)
- (Ex25) DnS:expresses(hotelBooker,searchHotel)
- (Ex26) File(www.hotelbooker.net)
- (Ex27) File-Format(html4.0)
- (Ex28) DnS:ordered-by(www.hotelbooker.net, html4.0)
- (Ex29) DOLCE:played-by(www.hotelbooker.net, hotelBooker)
- (Ex30) CSO:Software(firefox)
- (Ex31) Scenario(webBrowsing)
- (Ex32) DnS:expresses(firefox,webBrowsing)
- (Ex33) Functionality(accessWebsite)
- (Ex34) DnS:defines(webBrowsing, accessWebsite)
- (Ex35) DnS:inputFor(openWebsite, www.hotelbooker.net)

5.2 Subject: Task Execution

Following our hierarchy for the action layer of AT (see Sect. 3.1), comprising task execution (CWO:TaskProcess), knowledge action (CWO:KnowledgeAction), application action (CWO:ApplicationAction), and desktop operation (CWO:DesktopOperation), we apply the plan pattern of the OoP 2 (see Fig. 4). Modeling the task execution based on the OoP:AbstractPlan stresses the weak structure and adaptation of execution processes based on constraints we want to stress. An OoP:AbstractPlan describes methods for the execution of a procedure. A CWO:TaskProcess is internally-represented in an agent, has a goal, and uses at least one CWO:KnowledgeAction. Following a hierarchical model, each CWO:KnowledgeAction uses an CWO:ApplicationAction, which uses a CWO:DesktopOperation. A CWO:KnowledgeAction references a description it is about. An CWO:ApplicationAction references a CWO:SoftwareClass, which organizes software that shares similarities with respect to the tackled scenarios.

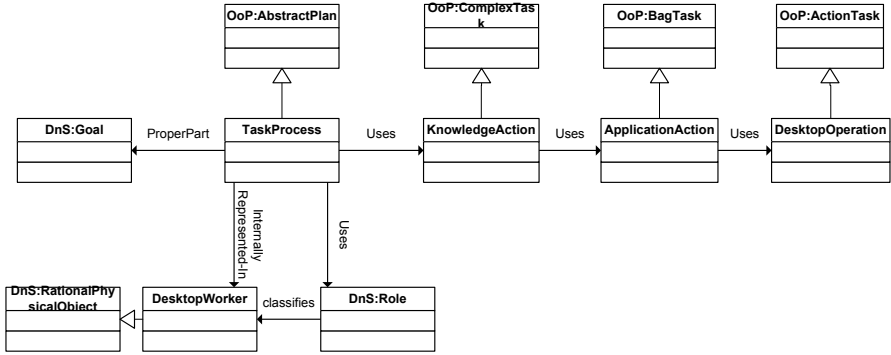


Fig. 4. The classification of the action hierarchy including TaskProcess, KnowledgeAction, ApplicationAction, and DesktopOperation and use of the planning pattern. Concepts taken from DOLCE and accompanying ontologies are labeled with the respective name space.

- (D8) $\text{DesktopWorker}(x) =_{def} \text{DnS:rational-physical-object}(x) \wedge \exists y(\text{internally-represented-by}(y,x) \wedge \text{TaskProcess}(y))$
- (D9) $\text{SoftwareClass}(x) =_{def} \text{DnS:Collection}(x) \wedge \forall y(\text{DnS:member}(x,y) \rightarrow \text{Software}(y))$
- (D10) $\text{TaskProcess}(x) =_{def} \text{OoP:AbstractPlan}(x) \wedge \forall y(\text{ComplexTask}(y) \wedge \text{uses}(x,y) \rightarrow \text{KnowledgeAction}(y))$
- (D11) $\text{KnowledgeAction}(x) =_{def} \text{ComplexTask}(x) \wedge \forall y(\text{uses}(x,y) \rightarrow \text{ApplicationAction}(y)) \wedge \exists z(\text{references}(x,z) \wedge \text{DnS:Description}(z))$
- (D12) $\text{ApplicationAction}(x) =_{def} \text{BagTask}(x) \wedge \forall y(\text{uses}(x,y) \rightarrow \text{DesktopOperation}(y)) \wedge \exists z(\text{references}(x,z) \wedge \text{CSO:SoftwareClass}(z))$
- (D13) $\text{DesktopOperation}(x) =_{def} \text{ActionTask}(x) \wedge \exists y(\text{uses}(x,z) \wedge \text{Functionality}(y))$

To give an example, we show the respective decomposition for step 1 in the initial example of creating a travel request for the manager (see Sect. 2).

- (Ex36) $\text{DesktopWorker}(\text{pete})$
- (Ex37) $\text{TaskProcess}(\text{createTravelRequest})$
- (Ex38) $\text{internally-represented-by}(\text{createTravelRequest}, \text{pete})$
- (Ex39) $\text{KnowledgeAction}(\text{browse})$
- (Ex40) $\text{description}(\text{travelRequest})$
- (Ex41) $\text{references}(\text{browse}, \text{travelRequest})$
- (Ex42) $\text{ApplicationAction}(\text{searchFile})$
- (Ex43) $\text{defines}(\text{browse}, \text{searchFile})$
- (Ex44) $\text{SoftwareClass}(\text{fileBrowser})$
- (Ex45) $\text{references}(\text{searchFile}, \text{fileBrowser})$

- (Ex46) DesktopOperation(openFolder)
- (Ex47) defines(searchFile,openFolder)

5.3 Object: Work Execution

We have described the decomposition of work into a hierarchy of actions, sequenced by a plan. Actions in AT are mediated by tools. In the CWO, we have modeled tools as software expressing scenarios that define functionalities. The mediation by a tool includes a process of tool selection, as the subject identifies a tool that sufficiently supports a given goal. To model this mediation process, we introduce the CWO:sufficient-implementation relation as a specialization of DnS:intensionally-references. CWO:sufficient-implementation expresses that a OoP:task can be adequately executed by using a respective DOLCE:endurant. We use the CWO:sufficient-implementation to connect the CWO:KnowledgeAction and the CWO:DesktopOperation with software and functionality as tools, to model the possible space of work execution (see Fig. 5).

Although the mediation process is modeled, the actual execution of work is not represented in the ontology. Such a modeling would require the description of the actual perdurants carried out by the user, such as clicking with a mouse or typing with a keyboard [12]. Since our focus is rather on the abstract work processes themselves than their modality-dependent execution, we have not included that level of detail in the CWO.

- (D14) KnowledgeAction(x) =_{def} OoP:ComplexTask
 $\wedge \forall y(\text{sufficient-implementation}(x,y) \rightarrow \text{Scenario}(y))$
- (D15) DesktopOperation(x) =_{def} OoP>ActionTask
 $\wedge \forall y(\text{sufficient-implementation}(x,y) \rightarrow \text{Functionality}(y))$

To illustrate the extension we again rely on the first step of the initial example (see Sect. 2). We connect the decomposition of the task to the different actions to the software chosen in the example.

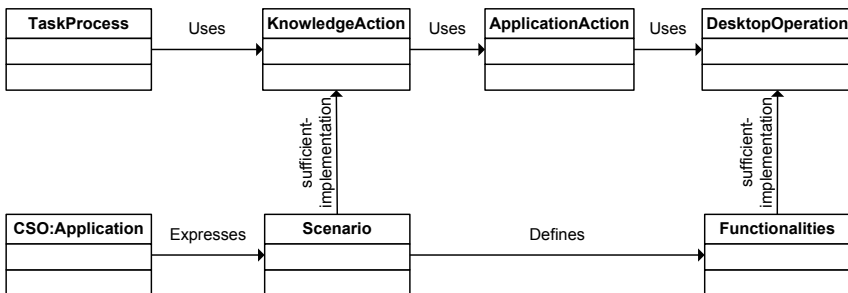


Fig. 5. The connection between the hierarchy of actions and software with scenarios and functionalities. Concepts taken from DOLCE and accompanying ontologies are labeled with the respective name space.

- (Ex48) TaskProcess(createTravelRequest)
- (Ex49) KnowledgeAction(browse)
- (Ex50) ApplicationAction(searchFile)
- (Ex51) defines(browse,searchFile)
- (Ex52) DesktopOperation(openFolder)
- (Ex53) defines(searchFile,openFolder)
- (Ex54) Software(windowsExplorer)
- (Ex55) Scenario(folderStructureInteraction)
- (Ex56) Functionality(browseFolderStructure)
- (Ex57) OIO:express(windowsExplorer,folderStructureInteraction)
- (Ex58) defines(folderStructureInteraction,browseFolderStructure)
- (Ex59) sufficient-implementation(searchFile,folderStructureInteraction)
- (Ex60) sufficient-implementation(openFolder,browseFolderStructure)

6 Related Work

The work presented in this paper uses the DOLCE ontology and describes concepts to model the computer workplace and to model work execution processes as selection of appropriate tools for fixed goals. In the following we give an overview of related work in the domain of computer workplace modeling and execution process modeling.

Computer workplace modeling: Computer workplace modeling exists as information object modeling and as software application modeling. The personal information model (PIMO) ontology [14] and the Attention Meta Data [16] are two examples for formalizations of the existing information objects. Both approaches capture file types and apply methods of classification and categorization to organize files with respect to the content. Definitions of transformations based on the type of files are not captured by such ontologies, as the application landscape is out of scope.

Modeling applications may focus on application classes, using taxonomies of software applications [1]. Another focus is a standalone application and the interaction of the user with the application. [13] provides the UICO ontology that connects basic actions with resources and information needs. As UICO focuses on input for trained machine learning, a detailed model of applications is out of scope.

Execution process modeling: Formalizations of execution processes generally provide a vocabulary to specify goals and realize a sequential or hierarchical task decomposition. The decomposition is realized by elements like *Object* and *Activity* in [4] or *Goal* and *Act* in the “Act Formalism” [10]. Modeling of the domain and knowledge-intensive planning are not tackled in depth by the reviewed approaches. More formalizations of execution processes can be found in [2].

Overall related work: The focus on a single subject that organizes a personal task execution in the sense of an execution process is not in the focus of the described approaches. Especially, the integration of the computer workplace as

domain and individual planning is not completely covered in any of the reviewed work. DOLCE with the DnS and the OoP extensions provides the necessary patterns, but requires additional classes to model the domain and additional properties to connect individual planning and the given domain, which has been realized by the CWO.

7 Conclusion

We have presented the Computer Work Ontology (CWO) that formalizes individual work execution in the domain of the computer workplace. Our ontology is grounded in foundational ontologies and enables the precise modeling of computer workplaces and a modeling of individual execution processes based on the definition of action in AT. We have shown the applicability of the CWO with a use case which was realized as running example in this paper. As we have applied modeling patterns belonging to DOLCE ontology, the CWO shows the applicability of those patterns for new domains.

The CWO focuses on a task perspective to describe work execution (endurants). As discussed above, user interactions like pressing a button on a keyboard, moving the mouse, etc. (perdurants) are out of scope for CWO. In the future, we are planning to connect our CWO with an already developed ontology of user interfaces and interactions [12]. As both ontologies are grounded the DOLCE ontologies, they are interoperable and can be integrated.

Currently, we use the CWO and DOLCE for two applications:

- *Task mining*: We enrich sensor events from a computer desktop with the respective data of the action hierarchy and create abstract patterns of work execution based on the captured work execution process.
- *Capturing document lifecycle*: We capture the transformation and dissemination of documents in a team of collaborative workers. The ontology captures the lifecycle of documents based on all acts of content enrichment, copying or disseminating.

In the future, we foresee to use CWO for providing proactive user support based on captured task instances. Thus, we will focus on the aspect of abstraction from specific work processes to more generic work processes.

Acknowledgements. The work presented in this paper has been partly funded by the German Federal Ministry of Education and Research under grant no. 01IA08006.

References

1. Forward, A., Lethbridge, T.C.: A Taxonomy of Software Types to Facilitate Search and Evidence-Based Software Engineering. In: Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds, pp. 1–13 (2008)

2. Gangemi, A., Borgo, S., Catenacci, C.: Task taxonomies for knowledge content. METOKIS Deliverable (2004)
3. Gangemi, A., Guarino, N., Masolo, C.: Sweetening ontologies with DOLCE, pp. 223–233. Springer, Heidelberg (2002)
4. Grüninger, M., Menzel, C.: Specification Language (PSL) Theory and Applications. *AI Magazine* 24(3), 63–74 (2003)
5. Hädrich, T.: Situation-oriented Provision of Knowledge Services. Dissertation, Martin Luther Universität Halle-Wittenberg (2008)
6. Kuutti, K.: Activity theory as a potential framework for human-computer interaction research. pp. 17–44. Massachusetts Institute of Technology, Cambridge (1995)
7. Leontiev, A.N.: Activity and consciousness. Progress Publishers (1977)
8. Marchionini, G.: Information Seeking in Electronic Environments. Cambridge University Press, Cambridge (1995)
9. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Horrocks, I.: WonderWeb Deliverable D18 Ontology Library (final) WonderWeb Project. Communities (2001)
10. Myers, K., Wilkins, D.: The Act Formalism, Version 2.2. SRI International Artificial Intelligence Center Technical Report (1997)
11. Oberle, D., Lamparter, S., Grimm, S., Vrande, D.: Towards Ontologies for Formalizing Modularization and Communication in Large Software Systems. Springer, Heidelberg (2006)
12. Paulheim, H., Probst, F.: A Formal Ontology on User Interfaces Yet Another User Interface Description Language? In: 2nd Workshop on Semantic Models for Adaptive Interactive Systems, (SEMAIS) (2011)
13. Rath, A.S.: UICO: An ontology-based user interaction context model for Automatic Task Detection on the Computer Desktop. In: CIAO 2009: Proceedings of the 1st Workshop on Context, Information and Ontologies, pp. 1–10 (2009)
14. Sauermann, L., Van Elst, L., Dengel, A.: Pimo-a framework for representing personal information models. *Proceedings of I-Semantics* 7, 270–277 (2007)
15. van der Aalst, W.M.P., van Hee, K.: *Workow Management. Models, Methods, and Systems*. MIT Press, Cambridge (2002)
16. Wolpers, M., Najjar, J., Verbert, K., Duval, E.: Tracking actual usage: the attention metadata approach, vol. 10, p. 106 (2007)

Semi-supervised Learning for Mixed-Type Data via Formal Concept Analysis

Mahito Sugiyama^{1,2} and Akihiro Yamamoto¹

¹ Graduate School of Informatics, Kyoto University
Yoshida Honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
mahito@iip.ist.i.kyoto-u.ac.jp,
akihiro@i.kyoto-u.ac.jp

² Research Fellow of the Japan Society for the Promotion of Science

Abstract. Only few machine learning methods; *e.g.*, the decision tree-based classification method, can handle *mixed-type data sets* containing both of discrete (binary and nominal) and continuous (real-valued) variables and, moreover, no *semi-supervised learning* method can treat such data sets directly. Here we propose a novel semi-supervised learning method, called SELF (SEmi-supervised Learning via FCA), for mixed-type data sets using *Formal Concept Analysis* (FCA). SELF extracts a lattice structure via FCA together with discretizing continuous variables and learns classification rules using the structure effectively. Incomplete data sets including missing values can be handled directly in our method. We experimentally demonstrate competitive performance of SELF compared to other supervised and semi-supervised learning methods. Our contribution is not only giving a novel semi-supervised learning method, but also bridging two fields of conceptual analysis and knowledge discovery.

Keywords: Semi-supervised learning, Classification, Mixed-type data, Formal Concept Analysis, Discretization, Concept lattice.

1 Introduction

The goal of this paper is to construct a novel semi-supervised learning method, called SELF (SEmi-supervised Learning via FCA), for *mixed-type data sets* including both discrete (binary and nominal) and continuous (real-valued) variables through *Formal Concept Analysis* (FCA) [5,10]. SELF makes the conceptual structure of a data set by FCA with discretizing real-valued variables, and learns classification rules with the class labels using the structure effectively. Figure 1 shows an overview of SELF.

Numerous mixed-type data sets are available these days. However, only few machine learning methods; *e.g.*, the decision tree-based classification method [20], can handle such data sets directly. In particular, no method can treat mixed-type data sets in *semi-supervised learning* [4,33]. This setting is a special form of classification; a learning algorithm uses both labeled and unlabeled data to obtain a classification rule. Only few labeled data are available in an actual situation since the task of labeling training data usually costs high. Thus machine learning and data mining methods, especially semi-supervised learning methods, for mixed-type data sets are required. Moreover, how to

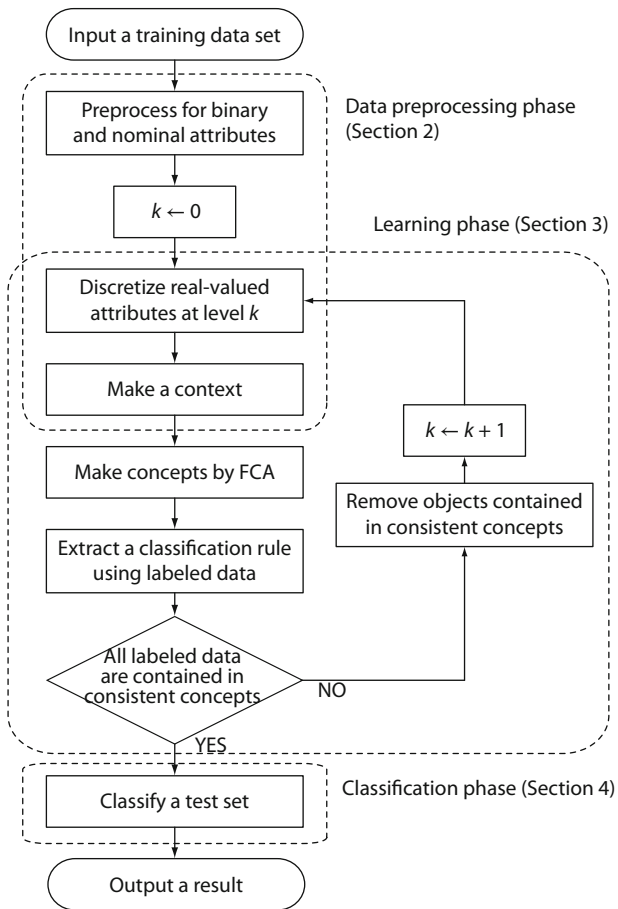


Fig. 1. A flowchart of the proposed SELF method. It learns classification rules from a training data set, and applies it to a test data set. The training data set contains (only few) labeled data and (lots of) unlabeled data. Here we say that a concept is consistent if all labels contained in the concept are same.

appropriately treat incomplete data sets containing missing values is one of crucial problems in knowledge discovery. In this paper, by applying FCA to semi-supervised learning, we directly treat such incomplete data sets and show the effectivity of FCA for mixed-type data sets in the setting of semi-supervised. Furthermore, we treat errors of discretized real-valued variables by embedding the discretizing process into the learning procedure and treat them directly in fully computational manner. Our results contribute to bridge two fields: conceptual analysis and knowledge discovery.

Our strategy is as follows: missing values are treated in the data preprocessing (Section 2) to make a context from a data set for applying FCA. Missing labels are included in the learning process (Section 3) and this is why we call our method semi-supervised. Discretization error is treated in the data preprocessing and embedded into learning phase to avoid overfitting by decreasing such error along with the learning process.

This paper is organized along the flowchart in Figure 1. Section 2 gives data preprocessing to construct a context. Section 3 describes our SELF method, and classification by obtained rules is considered in Section 4. Section 5 gives empirical evaluation of SELF. Section 6 gives related work, and the key points and future work are summarized in Section 7.

2 Data Preprocessing

The objective of data preprocessing is to construct a (formal) context for FCA from a training data set. A *dataset* X is given in the form of a *relation* [11]; i.e., a two-dimensional table with n tuples and d columns. Each column is called a *feature* 1 and each tuple corresponds to a datum (data point). Such dataset X can be expressed in the form of a matrix $[x_{ij}]_{n \times d}$ and the value of the i th datum for the feature j is denoted by x_{ij} . Missing values in X are allowed and denoted by the special symbol \perp . Three types of variables are considered: *binary*, *nominal*, and *real-valued*. For a dataset X , the domain of the feature j is denoted by D_j and throughout the paper, $D_j = \{\mathbf{T}, \mathbf{F}, \perp\}$ if j is binary, $D_j = \{1, \dots, v_j, \perp\}$ if nominal, and $D_j = \mathbb{R} \cup \{\perp\}$ if real-valued.

We call a triple (G, M, I) *context*. Here G and M are sets and $I \subseteq G \times M$ is a binary relation between G and M . The elements in G are called *objects*, and those in M are called *attributes*. Since G is always $\{1, 2, \dots, n\}$ for any dataset X with n data, we identify each object $g \in G$ with an identifier of each datum in X ; i.e., g indicates the g th datum (tuple) in X .

In the data preprocessing, for each feature $i \in \{1, 2, \dots, d\}$ of a dataset X , we independently construct a context (G, M_i, I_i) and combine them into a context (G, M, I) . In this process, we always *qualify* attributes to be disjoint by denoting each element m of the attribute M_j by $j.m$ followed by the way developed in database systems [11].

First, we focus on preprocessing for binary and nominal variables. For each feature $j \in \{1, \dots, d\}$, if it is binary and with no missing values; i.e., $x_{ij} \neq \perp$ for all $i \in \{1, \dots, n\}$, $M_j = \{\mathbf{T}\}$ and $(i, \mathbf{T}) \in I_j$ if $x_{ij} = \mathbf{T}$. If the variable j is binary with missing values, $M_j = \{\mathbf{T}, \mathbf{F}\}$, $(i, \mathbf{T}) \in I_j$ if $x_{ij} = \mathbf{T}$, and $(i, \mathbf{F}) \in I_j$ if $x_{ij} = \mathbf{F}$. Otherwise if j is nominal, $M_j = \{1, \dots, v_j\}$ and for all $i \in \{1, \dots, n\}$, $(i, m) \in I_j$ if $x_{ij} = m$. In this way, binary and nominal variables are directly translated into a context and missing values are naturally treated. Algorithm 1 performs this translation.

Second, we make a context from real-valued variables using discretization. This process is embedded in the learning process (see Figure 1) and discretizing resolution increases along with the process. The degree of resolution is denoted by a natural number k , called *level* of discretization, and in the following we explain how to discretize real-valued variables at fixed level k . First we use min-max normalization [12] so that every datum is in the closed interval $[0, 1]$. For every real-valued variable $X_j = (x_{1j}, \dots, x_{nj})$ of feature j in a dataset X , each x_{ij} is mapped to a value y_{ij} such that $y_{ij} = (x_{ij} - \min X_j) / (\max X_j - \min X_j)$. Next we discretize values in $[0, 1]$ and make a context using the encoding process for real numbers by base- β embedding [27]. If $\beta = 2$, intuitively this process coincides with the binary encoding of real numbers in

¹ It is usually called an *attribute*, but to avoid confusion with an attribute in a context, we use the word “feature”.

Algorithm 1. Data preprocessing for binary and nominal variables

Input: Dataset X with n objects and d features whose variables are binary or nominal

Output: Context (G, M_{BN}, I_{BN})
function CONTEXTBN(X)

```

1:  $G \leftarrow \{1, 2, \dots, n\}$ 
2: for each  $j$  in  $\{1, 2, \dots, d\}$ 
3:   if the feature  $j$  of  $X$  is binary and has no missing value then
4:      $M_j \leftarrow \{\mathbf{T}\}, I_j \leftarrow \{(i, x_{ij}) \mid i \in G \text{ and } x_{ij} = \mathbf{T}\}$ 
5:   else if the feature  $j$  of  $X$  is binary and has some missing value then
6:      $M_j \leftarrow \{\mathbf{T}, \mathbf{F}\}, I_j \leftarrow \{(i, x_{ij}) \mid i \in G \text{ and } x_{ij} \neq \perp\}$ 
7:   else // the feature  $j$  is nominal
8:      $M_j \leftarrow \{1, \dots, v_j\}, I_j \leftarrow \{(i, x_{ij}) \mid i \in G \text{ and } x_{ij} \neq \perp\}$ 
      // Note that we assume  $x_{ij} \in \{1, \dots, v_j\}$  for the feature  $j$ 
9:   end if
10: end for
11: combine  $(G, M_1, I_1), (G, M_2, I_2), \dots, (G, M_d, I_d)$  into  $(G, M_{BN}, I_{BN})$ 
12: return  $(G, M_{BN}, I_{BN})$ 

```

$[0, 1]$. At level k , $M = \{1, \dots, \beta^k\}$. For each x_{ij} , if $x_{ij} = 0$, then $(i, 1) \in I$. Otherwise if $x_{ij} \neq 0$, then $(i, m) \in I_j$ if and only if $x_{ij} \in ((m-1)/\beta^k, m/\beta^k]$. If $x_{ij} = \perp$, then $(i, m) \notin I_j$ for all $m \in M$. This means that if we encode x_{ij} by an infinite sequence $p = p_0p_1p_2\dots$, a context at level k is decided by the first k bits $p_0p_1\dots p_{k-1}$. Each value is converted to exactly one relation of a context. Algorithm 2 shows the above process for making a context from real-valued variables.

Example 1. Given a dataset

$$X = \begin{bmatrix} \mathbf{T} & 3 & 0.35 & 0.78 \\ \mathbf{F} & \perp & 0.813 & \perp \end{bmatrix},$$

where the first feature is binary, the second nominal with $D_3 = \{1, 2, 3\}$, and the third and the fourth real-valued. Assume that discretizing level $k = 1$. Then $G = \{1, 2\}$, $(M_1, I_1) = (\{\mathbf{T}\}, \{(1, \mathbf{T})\})$, $(M_2, I_2) = (\{1, 2, 3\}, \{(1, 3)\})$, $(M_3, I_3) = (\{1, 2\}, \{(1, 1), (2, 2)\})$, and $(M_4, I_4) = (\{1, 2\}, \{(1, 2)\})$. Thus we have the context $(G, M, I) = (\{1, 2\}, \{1.\mathbf{T}, 2.1, 2.2, 2.3, 3.1, 3.2, 4.1, 4.2\}, \{(1, 1.\mathbf{T}), (1, 2.3), (1, 3.1), (1, 4.2), (2, 3.2)\})$. It is visualized as a cross-table as follows:

	1. \mathbf{T}	2.1	2.2	2.3	3.1	3.2	4.1	4.2
1	×			×	×			×
2						×		

3 Learning via FCA

Here we propose a novel learning method, called SELF, that learns a set of classification rules via FCA using class labels. SELF makes a concept lattice from a context using

Algorithm 2. Data preprocessing for real-valued variables

Input: Real-valued dataset X and discretization level k **Output:** Context (G, M_R, I_R) **function** CONTEXTR(X, k)

```

1:  $G \leftarrow \{1, 2, \dots, n\}$ 
2: for each  $j$  in  $\{1, 2, \dots, d\}$ 
3:    $M_j \leftarrow \{1, 2, \dots, \beta^k\}$ 
4:   Normalize variables in the feature  $j$  of  $X$  by min-max normalization
5:    $I_j \leftarrow \emptyset$ 
6:   for each  $i$  in  $\{1, 2, \dots, n\}$ 
7:     if  $x_{ij} = 0$  then  $I_j \leftarrow I_j \cup \{(i, 1)\}$ 
8:     else if  $x_{ij} \neq 0$  and  $x_{ij} \neq \perp$  then
9:        $I_j \leftarrow I_j \cup \{(i, m)\}$ , where  $x_{ij} \in ((m-1)/\beta^k, m/\beta^k]$ 
10:    end if
11:  end for
12: end for
13: combine  $(G, M_1, I_1), (G, M_2, I_2), \dots, (G, M_d, I_d)$  into  $(G, M_R, I_R)$ 
14: return  $(G, M_R, I_R)$ 

```

FCA and finds classification rules from the concepts at each discretization level. Each rule is composed of a set of pairs of attributes and a label. This learning method is semi-supervised since it works even if few class labels are available.

3.1 Making Concept Lattices by FCA

First we summarize FCA (see literatures [5][10] for detail). We always assume that a given dataset X is converted into a context (G, M, I) by Algorithms 1 and 2.

For subsets $A \subseteq G$ and $B \subseteq M$, we define $A' := \{m \in M \mid (g, m) \in I \text{ for all } g \in A\}$ and $B' := \{g \in G \mid (g, m) \in I \text{ for all } m \in B\}$. Using these mappings, we can generate concepts from a given context; a pair (A, B) with $A \subseteq G, B \subseteq M$ is called a *concept* of a context (G, M, I) if $A' = B$ and $A = B'$. The set A is called an *extension* and B an *intension*.

The set of concepts over (G, M, I) is written by $\mathfrak{B}(G, M, I)$ and called the *concept lattice*. For a pair of concepts $(A_1, B_1), (A_2, B_2) \in \mathfrak{B}(G, M, I)$, we write $(A_1, B_1) \leq (A_2, B_2)$ if $A_1 \subseteq A_2$. Then we have the following: $(A_1, B_1) \leq (A_2, B_2)$ if and only if $A_1 \subseteq A_2$ (and if and only if $B_1 \supseteq B_2$). This relation \leq becomes an order on $\mathfrak{B}(G, M, I)$ in the mathematical sense and $(\mathfrak{B}(G, M, I), \leq)$ becomes a complete lattice. Let $\mathcal{C} \subseteq \mathfrak{B}(G, M, I)$. A concept $(A, B) \in \mathcal{C}$ is a *maximal element* of \mathcal{C} if $(A, B) \leq (X, Y)$ and $(X, Y) \in \mathcal{C}$ imply $(A, B) = (X, Y)$ for all $(X, Y) \in \mathcal{C}$. We write the set of maximal elements of \mathcal{C} by $\text{Max}\mathcal{C}$.

Many methods are available for making concept lattices, and the algorithm proposed by Makino and Uno [19] is known to be one of the fastest algorithms. Their algorithm enumerates all maximal bipartite cliques in a bipartite graph that coincides with the concept. Its computational complexity is theoretically bounded as $O(\Delta^3)$, where $\Delta = \max\{\#J \mid J \subseteq I, g = h \text{ for all } (g, m), (h, l) \in J, \text{ or } m = l \text{ for all } (g, m), (h, l) \in J\}$

($\#J$ is the number of elements in J). For empirical experiments, we use the program LCM [28] provided by the authors to enumerate all concepts.

If we see each concept as a cluster of a given dataset, then FCA can be viewed as a clustering method using only the closed property of the dataset. The concept lattice of (G, M, I) shows a lattice structure of clusters of objects and attributes. Note that each object usually belongs to more than one cluster, thus this is not “crisp” clustering.

3.2 Learning with Labels

For each object $g \in G$, we denote a *label*, an identifier of a class, of g by $\gamma(g)$, and if g is unlabeled; *i.e.*, the label information is missing, we write $\gamma(g) = \perp$. Moreover, we define $\Gamma(G) := \{g \in G \mid \gamma(g) \neq \perp\}$, hence objects in $\Gamma(G)$ are labeled data, and those in $G \setminus \Gamma(G)$ are unlabeled data.

Using labels, consistent concepts are defined. A concept $(A, B) \in \mathfrak{B}(G, M, I)$ is *consistent* if $\Gamma(A) \neq \emptyset$ and $\gamma(g) = \gamma(h)$ for all $g, h \in \Gamma(A)$. Note that a concept with $\Gamma(A) = \emptyset$ (all labels are missing) is not consistent.

Algorithm 3 is the main learning algorithm of SELF, which obtains a set of classification rules from a (training) dataset X . First it performs data preprocessing and makes the context (G, M, I) from X using the algorithms given in Section 2. Second it constructs the concept lattice $\mathfrak{B}(G, M, I)$ without class labels and finds consistent concepts using labeled data. If some objects that are not contained in consistent concepts remains, it refines discretization and repeats the above procedure.

This algorithm effectively uses the lattice structure of $\mathfrak{B}(G, M, I)$, that is, it traces the lattice from the top element, and all it has to do is to check the maximal consistent concepts. This lattice structure enables us to avoid overfitting since, informally, maximal concepts correspond to the most general classification rules.

Example 2. Given a dataset and its labels

$$(X, \gamma(X)) = \left(\begin{bmatrix} \mathbf{T} & 3 & 0.28 \\ \mathbf{F} & 1 & 0.54 \\ \mathbf{T} & 2 & \perp \\ \mathbf{F} & 1 & 0.79 \\ \mathbf{T} & 3 & 0.81 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ \perp \\ 2 \\ \perp \end{bmatrix} \right),$$

where the first feature is binary, the second nominal with $D_2 = \{1, 2, 3\}$, and the third real-valued. We fix the objects $G = \{1, 2, 3, 4, 5\}$. At level 1, we have the following context:

	1.T	2.1	2.2	2.3	3.1	3.2
1	×			×	×	
2		×				×
3	×		×			
4		×				×
5	×			×		×

Algorithm 3. Main learning algorithm of SELF; learning classification rules

Input: Dataset X with n objects and d attributes

Output: A set of classification rules \mathcal{R}

function MAIN(X)

- 1: Divide X into two datasets X_{BN} and X_{R} , where X_{BN} contains all binary and nominal variables in X , and X_{R} contains all real-valued variables in X
- 2: $(G, M_{\text{BN}}, I_{\text{BN}}) \leftarrow \text{CONTEXTBN}(X_{\text{BN}})$
// make a context from binary and nominal variables of X (see Section 2)
- 3: $k \leftarrow 1$ *// k is level of discretization*
- 4: $\mathcal{R} \leftarrow \text{LEARNING}(X_{\text{R}}, G, M_{\text{BN}}, I_{\text{BN}}, k, \emptyset)$ *// use this function recursively*
- 5: **return** \mathcal{R}

function LEARNING($X_{\text{R}}, G, M_{\text{BN}}, I_{\text{BN}}, k, \mathcal{R}$)

- 1: $(G, M_{\text{R}}, I_{\text{R}}) \leftarrow \text{CONTEXTTR}(X_{\text{R}}, k)$
// make a context from real-valued variables of X at level k (see Section 2)
 - 2: make (G, M, I) from $(G, M_{\text{BN}}, I_{\text{BN}})$ and $(G, M_{\text{R}}, I_{\text{R}})$
 - 3: build the concept lattice $\mathfrak{B}(G, M, I)$ from (G, M, I) (see Section 3)
 - 4: $\mathcal{C} \leftarrow \{(A, B) \in \mathfrak{B}(G, M, I) \mid (A, B) \text{ is consistent}\}$
 - 5: $\mathcal{R}_k \leftarrow \{(B, \gamma(a)) \mid (A, B) \in \text{Max}\mathcal{C} \text{ and } a \in \Gamma(A)\}$
 - 6: $\mathcal{R} \leftarrow \mathcal{R} \cup (\mathcal{R}_k, k)$ *// add the current result \mathcal{R}_k at this level k to \mathcal{R}*
 - 7: $G \leftarrow G \setminus \{g \mid g \in A \text{ for some } (A, B) \in \mathcal{C}\}$
 - 8: remove corresponding attributes and relations from M_{BN} and I_{BN} , respectively
 - 9: remove corresponding objects from X_{R}
 - 10: **if** $\Gamma(G) = \emptyset$ **then return** \mathcal{R}
 - 11: **else return** LEARNING($X_{\text{R}}, G, M_{\text{BN}}, I_{\text{BN}}, k + 1, \mathcal{R}$)
 - 12: **end if**
-

We show this lattice in the left-hand side in Figure 2. By our learning algorithm, we obtain $\mathcal{R}_1 = \{(\{1, \mathbf{T}\}, 1)\}$ since the concept $(\{1, 3, 5\}, \{1, \mathbf{T}\})$ is the maximal consistent concept, and there is no consistent concept that contains 2 or 4. Thus SELF removes objects 1, 3, and 5 and proceed to the next level. At level 2, we have the following context:

	1.T	2.1	2.2	2.3	3.1	3.2	3.3	3.4
2		×					×	
4		×						×

The right-hand side in Figure 2 shows the concept lattice of the above context, and we obtain $\mathcal{R}_2 = \{(\{2.1, 3.3\}, 1), (\{2.1, 3.4\}, 2)\}$. We therefore have the set of classification rules $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2\}$.

We show that SELF always stops in finite time if there are no conflicting objects.

Theorem 1. *Given a dataset X with the objects $G = \{1, \dots, n\}$. If there is no pair $g, h \in G$ such that $\gamma(g) \neq \gamma(h)$ and $x_{gj} = x_{hj}$ for all $j \in \{1, \dots, d\}$, Algorithm 3 stops in finite time.*

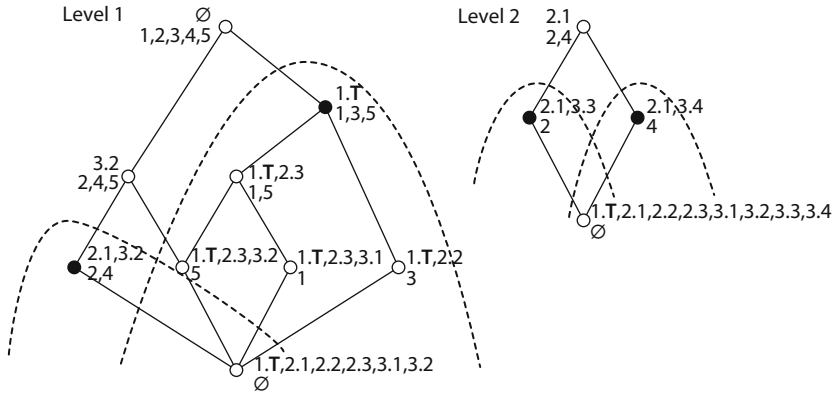


Fig. 2. The concept lattices constructed during the learning phase in Example 2. In these diagrams, each black dot is the maximal concept in the set of concepts covered by the dotted line.

Proof. If discretization level k is large enough, we have the concept lattice $\mathfrak{B}(G, M, I)$, where for every object $g \in G$, there exists a concept (A, B) such that $A = \{g\}$ since there is no pair $g, h \in G$ satisfying $x_{gj} = x_{hj}$ for all $j \in \{1, \dots, d\}$. Thus each object g with $\gamma(g) \neq \perp$ must be contained in some consistent concept, and the algorithm stops. \square

This theorem means that the algorithm works even if $\Gamma(G) = G$; *i.e.*, all objects have labels. Thus it also can be viewed as a supervised learning method.

The computational complexity of our learning algorithm is $O(nd) + O(\Delta^3) + O(\Lambda)$, where Λ is the number of concepts at level 1, since data preprocessing (Algorithm 1 and 2) takes $O(nd)$, making concepts takes $O(\Delta^3)$, and judging consistency of concepts takes less than $O(\Lambda)$.

4 Classification

Now we have a set of classification rules $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_K\}$ from a training dataset X by Algorithms 1, 2 and 3. In this section, we show how to classify a datum x using \mathcal{R} . We always assume that every datum has the same features $1, 2, \dots, d$ as a training dataset.

Algorithm 4 shows the classification algorithm using the obtained rules \mathcal{R} . It performs as a multi-class classification method, and we can handle multi-class datasets directly. It receives rules \mathcal{R} and a datum x , and produces the set of *label candidates* $L(x) = \{l_1, l_2, \dots, l_C\}$. The procedure is levelwise; *i.e.*, it checks rules from $\mathcal{R}_1, \mathcal{R}_2, \dots$, to \mathcal{R}_K , and at each level k , it makes a context (G, M, I) from the datum x . It finds all class labels l satisfying $R \in M$ for some $(R, l) \in \mathcal{R}_k$. This means that x has the same property as the concept that has attributes R . Note that the set G is always a singleton $\{1\}$ in the classification phase.

Algorithm 4. Classification of data**Input:** Set of classification rules $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_K\}$ and datum $x = [x_{ij}]_{1 \times d}$ **Output:** Label candidates $L = \{l_1, l_2, \dots, l_C\}$ **function** CLASSIFY(R, x)1: $L \leftarrow \emptyset$ 2: divide x into two data x_{BN} and x_{R} , where x_{BN} contains all binary and nominal variables in x and x_{R} contains all real-valued variables in x 3: $(G, M_{\text{BN}}, I_{\text{BN}}) \leftarrow \text{CONTEXTBN}(x_{\text{BN}})$ // make a context from binary and nominal variables of x (see Section 2)4: **for each** k in $\{1, 2, \dots, K\}$ 5: $(G, M_{\text{R}}, I_{\text{R}}) \leftarrow \text{CONTEXTTR}(x_{\text{R}}, k)$ // make a context from real-valued variables of x at level k (see Section 2)6: make the context (G, M, I) from $(G, M_{\text{BN}}, I_{\text{BN}})$ and $(G, M_{\text{R}}, I_{\text{R}})$ 7: add l to L if $R \in M$ for some $(R, l) \in \mathcal{R}_k$ 8: **end for**9: **return** L

Example 3. Let us consider about Example 2. A datum $x = (\mathbf{T}, 2, 0.45)$ is classified to the class 1 since the first value is \mathbf{T} , and a datum $y = (\mathbf{F}, 1, 0.64)$ is also classified to the class 1 since the second value is 1 and the third value is in the interval $(0.5, 0.75]$.

This algorithm is similar to one-against-all classification methods [31] since each pair $(R, l) \in \mathcal{R}_k$ at level k judges whether or not the object belongs to the class l . As a result, our method produces candidates of labels $L(x) = \{l_1, l_2, \dots, l_C\}$ (note that $L(x)$ may be empty), and cannot decide the unique label. All one-against-all classification methods have this problem intrinsically; e.g., one-against-all SVMs [1], and solving this problem is one of future works.

We can guarantee that every labeled datum in a given training dataset is always classified to some class using the set of classification rules \mathcal{R} .

Theorem 2. Let X be a dataset with n objects and d features and \mathcal{R} the obtained classification rules by Algorithm 3. For all data $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ in X with $\gamma(i) \neq \perp$, we have $L(x_i) \neq \emptyset$.

Proof. Algorithm 3 stops if and only if every labeled object is contained in some consistent concept. This means that if $\gamma(i) \neq \perp$, we have $L(x_i) \neq \emptyset$ for all $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ in X . \square

5 Experiments

Here we evaluate our SELF method empirically. We compare accuracy of classification using datasets from UCI Machine Learning Repository [7] and the benchmark datasets used in the literature [4].

5.1 Materials and Methods

A direct way to evaluate SELF is to compare accuracy of it to that of other semi-supervised learning methods using mixed-type datasets. However, to our best knowledge, no semi-supervised methods can treat such mixed-type datasets directly. Thereby we evaluated our method by two different experiments.

In the first experiment, we compared SELF to other supervised classification methods using mixed-type datasets. In the second experiment, we compared SELF to other semi-supervised learning methods using real-valued datasets.

SELF was implemented in R version 2.12.1 [24] and all experiments were performed in the R environment. For enumeration of all concepts from a context, we used LCM [28] distributed by Uno [3]. Throughout all experiments, we set $\beta = 2$ (see Section 2), that is, discretized real-valued variables by binary encoding of real numbers. In the classification phase, if $L(x) = \emptyset$; i.e., a data point x is not classified to any class, we set $\gamma(x)$ as the smallest mode of $\{\gamma(g) \mid g \in \Gamma(G)\}$, where G is the objects of a training dataset. Otherwise if $\#L(x) \geq 2$ ($\#L(x)$ denotes the number of elements in $L(X)$), we set $\gamma(x)$ as the smallest mode of $L(x)$.

For the first experiment, we collected nine mixed-type datasets from UCI repository [7]: *ad*, *allbp*, *anneal*, *australian*, *crx*, *echoc*, *heart*, *hepatitis*, and *horse*. For the second experiment, we adopted the benchmark datasets used in the literature [4]: *g241c*, *g241d*, *Digit1*, *USPS*, *COIL*, and *BCI*. All of them consist of only real-valued variables.

To analyze effectivity of unlabeled data, we performed SELF in two ways: one is using both labeled and unlabeled data for training, and the other is using only labeled data. For a control method in the first experiment, we used the decision tree-based method implemented in R, which is supplied in the `tree` package [25]. For reference, we also performed the nearest neighbor method (1NN) in the `class` package in R using only real-valued variables.

In the first experiment, we performed 10-fold cross-validation [12]. We divided the given dataset into 10 datasets randomly and set that one fold was a training labeled dataset, another one was a test dataset, and the rest of them was a training unlabeled dataset. Moreover, we selected from 10 to 100 data from the fold of training labeled dataset to check the effectivity of the number of labeled data. Note that control methods (decision tree and 1NN) did not use the unlabeled dataset.

The second experiment was carried out in the transductive setting [30] as in exactly the same way in the literature [4], that is, the test set coincides with the set of unlabeled data in the training dataset. The results of accuracy obtained from other semi-supervised learning method have already given [4] and we compared our result to the given results.

5.2 Results and Discussion

The experimental result for the first experiment is shown in Figure 3 and that of the second experiment in Table 1.

In the first experiment, accuracy of SELF is better than that of the decision tree-based method in more than half cases. This means that our method works well for mixed-type

² <http://research.nii.ac.jp/~uno/codes.htm>

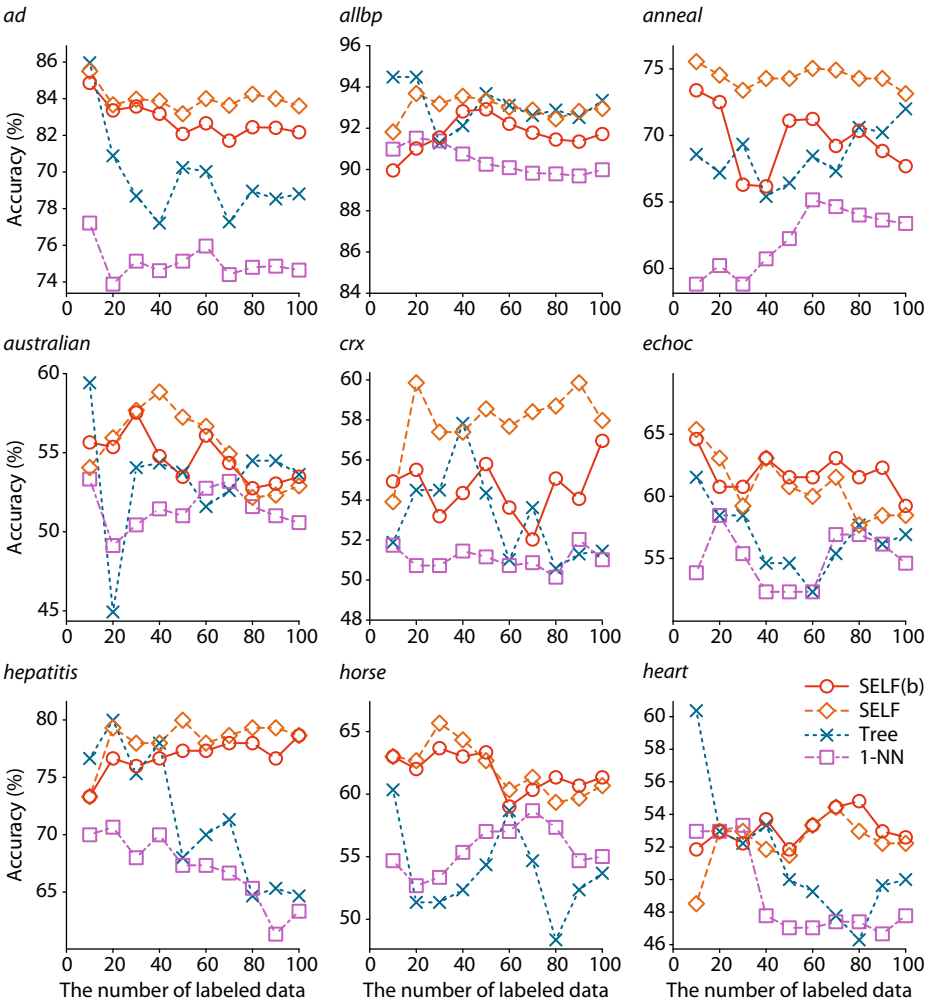


Fig. 3. The experimental result of accuracy (%) for mixed-type datasets from UCI repository. We performed our SELF method using both labeled and unlabeled data (SELF(b)) and only using labeled data (SELF), and compared them to the decision tree-based classification method (Tree) and the nearest neighbor method (1-NN).

datasets. However, in many cases, accuracy obtained using only labeled data is better than that obtained using both of labeled and unlabeled data. This shows the difficulty of effectively using unlabeled data.

In contrast, in the second experiment, accuracy obtained with both labeled and unlabeled data are better than that obtained with only labeled data for all datasets. The reason may be that unlabeled data were carefully prepared in the benchmark datasets by the donor, whereas generated randomly in the first experiment. Accuracy on 100 labeled data is better than those on 10 labeled data. Thus if the size of labeled data

Table 1. The experimental result of accuracy (%) on benchmark datasets. See the literature [4] or website³ for the results of other semi-supervised learning methods. To evaluate effectivity of unlabeled data, we performed our SELF method using both labeled and unlabeled data (SELF(b)), and only using labeled data (SELF).

	The number of labeled data is 10						The number of labeled data is 100					
	<i>g241c</i>	<i>g241d</i>	<i>Digit1</i>	<i>USPS</i>	<i>COIL</i>	<i>BCI</i>	<i>g241c</i>	<i>g241d</i>	<i>Digit1</i>	<i>USPS</i>	<i>COIL</i>	<i>BCI</i>
SELF(b)	52.33	52.41	58.36	75.12	39.91	58.67	67.01	67.03	72.62	83.19	70.18	88.08
SELF	50.55	51.27	53.03	75.04	23.42	50.44	54.37	53.87	59.98	77.44	46.09	64.56

increases, performance of SELF could become better. Compared to other semi-supervised learning methods, our method basically achieves competitive performance on most of datasets. Moreover, our performance is the best on the dataset *BCI*.

Therefore the first and second experiments indicate that SELF works better than existing supervised learning method for mixed-type datasets and competitive to existing semi-supervised learning methods.

6 Related Work

Many studies used FCA for machine learning and knowledge discovery [16], such as classification [8,9], clustering [32], association rule mining [13,21,29], and bioinformatics [2,14,17]. In particular, Ganter and Kuznetsov [8] attacked to the problem of binary classification for real-valued data and proposed algorithms based on the *JSM-method* that produce hypotheses (classifiers) using positive and negative examples. Their idea of using the lattice structure derived by FCA for classification is similar to our approach, but the way of treating real-valued variables is different. Their method discretizes real-valued variables by inequations, called *conceptual scaling* [10], that are given *a priori*, while SELF automatically discretizes them along with the learning process and no background knowledge for datasets is needed. Moreover, this paper is the first one that treats the modern machine learning problem semi-supervised learning.

In machine learning context, decision tree-based methods, *e.g.*, C4.5 [22,23], can treat mixed-type data by discretizing continuous variables, and there are several discretization techniques [6,18,26] to treat continuous variables in the discrete manner. Our approach is different from them since we integrate discretization process into learning process and avoid overfitting. Furthermore, no semi-supervised learning method integrates two processes of learning and discretization. Kok and Domingos [15] have proposed a learning method via hypergraph lifting, which constructs clusters by hypergraphs and learns on them. Their idea is similar to ours since we also “lift” raw data to the space of a concept lattice via FCA. However, it is difficult to treat continuous variables in their approach, thereby our approach can be more useful for knowledge discovery from actual mixed-type datasets.

³ <http://www.kyb.tuebingen.mpg.de/ssl-book/benchmarks.html>

7 Conclusion

We have proposed a novel semi-supervised learning method, called SELF, for mixed-type datasets (*i.e.*, datasets that have both continuous and discrete variables) using FCA and experimentally showed its competitive performance. To our best knowledge, this method is the first direct semi-supervised method for mixed-type datasets. Moreover, we can directly treat missing values on SELF. It uses only the algebraic structure of datasets, thus it can learn classification rules without any data distribution.

Refinement of discretization of real-valued variables must have some connection with *reduction* of a context [10] since if we extend a context by refining real-valued variables, the original attributes are removed by reduction. Thereby analysis of mathematical connection between them is a future work.

de Brecht and Yamamoto [3] have proposed *Alexandrov concept space* for learning from positive data in the computational learning theory context. Our proposed method might be an instance of the study, since the concept lattice is similar to the Alexandrov space. Thus theoretical analysis of our framework is an another future work.

Acknowledgment. This work was partly supported by Grant-in-Aid for Scientific Research (A) 22240010 and for JSPS Fellows 22-5714.

References

1. Abe, S.: Analysis of multiclass support vector machines. In: Proceedings of International Conference on Computational Intelligence for Modeling Control and Automation, pp. 385–396 (2003)
2. Blinova, V.G., Dobrynin, D.A., Finn, V.K., Kuznetsov, S.O., Pankratova, E.S.: Toxicology analysis by means of the JSM-method. *Bioinformatics* 19(10), 1201–1207 (2003)
3. de Brecht, M., Yamamoto, A.: Topological properties of concept spaces (full version). *Information and Computation* 208, 327–340 (2010)
4. Chappelle, O., Schölkopf, B., Zien, A. (eds.): *Semi-Supervised Learning*. MIT Press, Cambridge (2006), <http://www.kyb.tuebingen.mpg.de/ssl-book>
5. Davey, B.A., Priestley, H.A.: *Introduction to lattices and order*, 2nd edn. Cambridge University Press, Cambridge (2002)
6. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1029 (1993)
7. Frank, A., Asuncion, A.: UCI machine learning repository (2010), <http://archive.ics.uci.edu/ml>
8. Ganter, B., Kuznetsov, S.: Formalizing hypotheses with concepts. In: Ganter, B., Mineau, G.W. (eds.) *ICCS 2000*. LNCS, vol. 1867, pp. 342–356. Springer, Heidelberg (2000)
9. Ganter, B., Kuznetsov, S.: Hypotheses and version spaces. In: de Moor, A., Lex, W., Ganter, B. (eds.) *ICCS 2003*. LNCS, vol. 2746, pp. 83–95. Springer, Heidelberg (2003)
10. Ganter, B., Stumme, G., Wille, R. (eds.): *Formal Concept Analysis*. LNCS (LNAI), vol. 3626. Springer, Heidelberg (2005)
11. Garcia-Molina, H., Ullman, J.D., Widom, J.: *Database systems: The complete book*. Prentice Hall Press, Englewood Cliffs (2008)
12. Han, J., Kamber, M.: *Data Mining*, 2nd edn. Morgan Kaufmann, San Francisco (2006)

13. Jaschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS—An algorithm for mining iceberg tri-lattices. In: Proceedings of the 6th International Conference on Data Mining, pp. 907–911. IEEE, Los Alamitos (2006)
14. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences* (2010)
15. Kok, S., Domingos, P.: Learning Markov logic network structure via hypergraph lifting. In: Proceedings of the 26th International Conference on Machine Learning. pp. 505–512 (2009)
16. Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 287–312. Springer, Heidelberg (2004)
17. Kuznetsov, S.O., Samokhin, M.V.: Learning closed sets of labeled graphs for chemical applications. In: Kramer, S., Pfahringer, B. (eds.) ILP 2005. LNCS (LNAI), vol. 3625, pp. 190–208. Springer, Heidelberg (2005)
18. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. *Data Mining and Knowledge Discovery* 6(4), 393–423 (2002)
19. Makino, K., Uno, T.: New algorithms for enumerating all maximal cliques. In: Hagerup, T., Katajainen, J. (eds.) SWAT 2004. LNCS, vol. 3111, pp. 260–272. Springer, Heidelberg (2004)
20. Murthy, S.K.: Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery* 2(4), 345–389 (1998)
21. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient mining of association rules using closed itemset lattices. *Information Systems* 24(1), 25–46 (1999)
22. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann, San Francisco (1993)
23. Quinlan, J.R.: Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research* 4, 77–90 (1996)
24. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing (2011), <http://www.R-project.org>
25. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
26. Skubacz, M., Hollmén, J.: Quantization of continuous input variables for binary classification. In: Leung, K.-S., Chan, L., Meng, H. (eds.) IDEAL 2000. LNCS, vol. 1983, pp. 42–47. Springer, Heidelberg (2000)
27. Sugiyama, M., Yamamoto, A.: The coding divergence for measuring the complexity of separating two sets. In: Proceedings of 2nd Asian Conference on Machine Learning. JMLR Workshop and Conference Proceedings, vol. 13, pp. 127–143 (2010)
28. Uno, T., Kiyomi, M., Arimura, H.: LCM ver. 3: Collaboration of array, bitmap and prefix tree for frequent itemset mining. In: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, pp. 77–86. ACM, New York (2005)
29. Valtchev, P., Missaoui, R., Godin, R.: Formal concept analysis for knowledge discovery and data mining: The new challenges. *Concept Lattices*, 3901–3901 (2004)
30. Vapnik, V., Sterin, A.: On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control* 10(3), 1495–1503 (1977)
31. Vapnik, V.N.: The nature of statistical learning theory. Springer, Heidelberg (2000)
32. Zhang, Y., Feng, B., Xue, Y.: A new search results clustering algorithm based on formal concept analysis. In: Proceedings of 5th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 356–360. IEEE, Los Alamitos (2008)
33. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. Morgan and Claypool Publishers, San Francisco (2009)

Towards Structuring Episodes in Patient History

Galia Angelova¹, Svetla Boytcheva^{1,2}, and Dimitar Tcharaktchiev³

¹Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences, Sofia, Bulgaria

²University for Library Studies and Information Technology, Sofia, Bulgaria

³University Specialised Hospital for Active Treatment of Endocrinology,
Medical University Sofia, Bulgaria

Abstract. This article discusses current results in automatic Information Extraction (IE) of temporal markers from hospital Patient Records (PRs) texts. The aim is to construct a temporal sequence of important facts about phases in disease development, by recognising the main events that are described in the anamnesis (case history). We consider the conceptual structure of "episode", which is designed as a feature-value template enabling the further formalisation of the case history via generation of simple Conceptual Graph (CG). Evaluation results assessing the recognition of temporal markers are presented as well.

Keywords: Natural Language Processing (NLP), Event recognition, Temporal markers, Episode sequencing, Capturing structured information from free text.

1 Introduction

Defining events is difficult as the perspective can vary depending on people knowledge and task context. Events are observable or hypothetical occurrences in certain locations at particular moments of time. A medical event is, for instance, the diagnosing of a disease accompanied by a number of examinations – which can take weeks, but swallowing a single pill is an event as well. Events are described in texts with various depth and granularity; by default any verb refers to an event or state but an event instance can be also expressed by several consecutive sentences.

In Artificial Intelligence, events are treated as entities distinct from the things that participate in the happening or occurrence [1]. In linguistics, treating events as quantifiable entities enables to consider their instances as particular individuals with specific participants, time and location. Computational linguistics constructs fine-grained representations of event descriptions in text, viewing time as an essential aspect of text understanding. Recently the markup language TimeML for annotation of events and temporal relations was developed [2]. Events in TimeML are situations that happen or occur, they can be punctual or last some time, and may be expressed by means of verbs, nominalisations, adjectives, predicative clauses, or prepositional phrases. In this way TimeML suggests a text-based framework for detailed event annotation which facilitates event identification and extraction in automatic analysis of free text.

Our research project aims to discover patterns in disease developments by searching similar case histories. This includes, among others, extraction of structured event descriptions from free texts in order to explicate the temporal relations that hold between the events and participating things. In the area of biomedical NLP, the research on event recognition is a relatively recent activity. The article [3] analyses the potential of TimeML tags as annotation tool for clinical narratives. The paper [4] considers features of patient conditions which are described in clinical reports: they can be *negated*, *hypothetical*, *historical*, or experienced by someone *other* than the patient. The suggested algorithm ConText infers the status of a condition with regard to these properties from simple lexical clues occurring in the context of the condition. ConText is tested for 4654 annotations in 240 clinical reports using two kinds of local contexts: a six-token window (*stw*) and end-of-sentence context. The algorithm shows excellent performance in recognition of *negated conditions* (more than 97% f-score¹), good accuracy for *historical* conditions (more than 73% f-score), and about 50% f-score for the recognition of *hypothetical* conditions which are discussed within the sentences and processed in *stw* contexts. The authors conclude that “a comprehensive solution to the problem of determining whether a clinical condition is *historical* or *recent* requires knowledge above and beyond the surface clues picked up by ConText” [4]. The accuracy of 73% for *historical* conditions is a benchmark for the IE of temporal information. An alternative representation of five tags (*reference point*, *direction*, *number*, *time unit*, and *pattern*) for marking up temporal information in patient records is proposed in [5]. All listed approaches provide useful hints and design considerations for our task. In this paper we present current research results in the automatic identification of clinical episodes.

2 Episodes in Hospital Patient Records

In medicine, an episode comprises all activities that are performed between the diagnosis of disease and its cure; normally the episode is decomposed to goals and actions. The patient-related documentation is related to this default fragmentation of healthcare tasks [6]. For chronic diseases instead of cure we talk about stabilisation, e.g. the diabetes is compensated. Various approaches to event modeling are possible but we select those which are closest to the semantic structure of the discharge letters. Our goal is to determine and annotate the granularity of temporal intervals, when important clinical events occur, and we consider as episodes *sets of events defined via the explicit temporal markers uttered by the physicians* who examine and treat the patients. An episode is, for instance, the diagnosing of a disease at some moment of time; the treatment prescribed then can be viewed as a feature. The next episode might be the occurrence of certain disease complications after several years, which are treated by some drugs and so on. Thus we can consider the case history as a sequence of phases or episodes which summarise the chronology of disease-relevant events. Explicit temporal markers enable the identification of such events in the clinical records. Additionally, the narrative convention (to utter events in the sequence of appearance) helps much to capture structured temporal information.

¹ The IE accuracy is measured by the *precision* (percentage of correctly extracted entities as a subset of all extracted entities), *recall* (percentage of correctly extracted entities as a subset of all available entities) and *f-score* = $2 * Precision * Recall / (Precision + Recall)$.

We deal with a corpus of 6300 anonymised hospital PRs of diabetic patients, delivered by the University Specialised Hospital for Active Treatment of Endocrinology (USHATE) which belongs to Medical University Sofia. This hospital treats citizens with specific, complex history cases from all over the country. Most of the PRs present patients with diabetes diagnosed decades ago. The discharge letters summarise the most important facts and enumerate accompanying diseases. Some patients have up to 30 diagnoses listed in the hospital PRs (but most have up to 7 diagnoses). The case history is summarised with a specific level of granularity into several paragraphs (the text quality depends on the writer but the intention to provide it is always there). In general only the major illness phases are discussed together with the treatment and medication changes. Thus we actually work on specific sketchy abstracts typed in by human experts. The following sample is written in 2004:

Example 1. Diabetes Mellitus diagnosed in 2003, manifested by most symptoms - polyuria, polydipsia, lost 20 kg in 6 months with reduced appetite. Prescribed Maninil 3,5 mg 1+1 tabl. for a period of 3 months. Since then no blood sugar was tested and no further therapy was carried out. 20 years ago enlarged thyroid, sometimes the patient had suffocation and palpitation, but no examinations were made and no therapy was carried out.

This paragraph is entered in the zone *Anamnesis* of the respective PR. Other sections might be also included there with sub-headers: *Accompanying* or *Past diseases*, *Family history*, *Risk factors*, and patient *Allergies*. The *Anamneses* are automatically recognised with 100% accuracy as they are typed in as separate PR sections.

After manual investigation of numerous PRs and experiments in automatic extraction using a training set of 1300 PRs, we find out that our definition of episode is reasonable at least for the PRs of USHATE. They seem to contain a sufficient number of temporal markers which enable successful text splitting into episodes. We believe that human experts declare explicitly the most important temporal markers which are sufficient (in their view) to adequately communicate the case history to another medical doctor. Therefore, we consider these markers as primary signals for diseases progression phases. Our model is framed using three tags suggested in [5]: (i) *reference point*, (ii) *direction*, and (iii) *temporal expression* plus additional tags needed for structuring *episodes* in our project: (iv) *temporal marker for episode end*, (v) *diagnoses, complains or symptoms* (i.e. what happens, occurs or is found during the episode), (vi) *drugs/treatment* applied during the episode, and (vii) *treatment effect*. There could be several diagnoses and/or drugs enumerated in one episode.

The episodes of Example 1 are structured in Table 1. The conventional 'now' denotes the writing moment. Ideally, we want to build correct temporal sequences of all clinical events but [5] cites only 75% inter-annotators agreement for manual annotation of 254 temporal expressions in 50 discharge summaries. Note that 'since then' in episode 4 might be hard for humans too. It also remains unclear whether one should annotate periods when nothing happens (for instance episode 4).

There can be non-trivial temporal references, e.g. 'Two medication courses were made in 1990 and 1991'. The events, happening within episodes, are easier to position and interpret after the recognition of episode boundaries. To build a chronologic model, the relative temporal clauses need to be resolved by calculation of actual dates, e.g. Episode 1 starts '5-6 years ago' which can be interpreted as 5,5 years ago.

Table 1. Manually-constructed temporal event sequencing for the case presented in Example 1

Ep1	Reference point	Now minus 20 years
	Direction	forward
	Temporal expression	20 years ago
	Episode end	
	Diagnoses, complains, symptoms	enlarged thyroid, sometimes suffocation and palpitation
	Drugs/Treatment	
	Treatment effect	
Ep2	Reference point	2003 (diagnosis point)
	Direction	backward
	Temporal expression	in 6 months
	Episode end	2003 (diagnosis point)
	Diagnoses, complains, symptoms	lost 20 kg with reduced appetite
	Drugs/Treatment	no
	Treatment effect	
Ep3	Reference point	2003 (diagnosis point)
	Direction	forward
	Temporal expression	in 2003
	Episode end	2003 (diagnosis point) plus 3 months
	Diagnoses, complains, symptoms	Diabetes Mellitus, polyuria, polydipsia
	Drugs/Treatment	Maninil 3,5 mg 1+1 tabl. (for a period of 3 months)
	Treatment effect	
Ep4	Reference point	2003 (diagnosis point) plus 3 months
	Direction	forward
	Temporal expression	since then
	Episode end	Now (moment of hospitalisation)
	Diagnoses, complains, symptoms	-
	Drugs/Treatment	no
	Treatment effect	
Ep5	Reference point	Now (moment of hospitalisation)
	

3 Automatic Discovery of Episodes

The current IE prototype integrates previously developed software components: (i) a module for extraction of drug names, dosage, frequency and route, which identifies 1537 drug names in 6200 PRs with f-score 98,42% and dosage with f-score 93,85%, and (ii) a module for automatic recognition of diagnoses in a corpus of 6200 PRs with 84,5% precision [7]. These modules encode the diagnoses and drugs by the labels of international taxonomies: the International Classification of Diseases (ICD-10) and the Anatomical Therapeutic Chemical drug classification (ATC). In this way, given the episode description by 7 attributes, one can assume that the attributes 5th and 6th are automatically filled in by standardised labels to large extent. Thus our efforts are directed mostly to identification and conceptualisation of symptoms, complains and treatment effects, as well as to the systematic study of various temporal markers.

Regarding the automatic recognition of symptoms and complain in the free text of hospital PRs, these are described by a variety of expressions, ranging from the classical (rich) medical terminology to free explanations, using the patient wording and stories told when the patient is interviewed in the hospital admission office (the latter may also contain temporal references). In this way the automatic identification of

symptoms and complain is a complicated task which requires incremental construction of lexicons and task-specific training corpora. The conceptualisation of treatment effects is easier as they are presented mostly by typical phrases.

We have performed experimental tests with 1375 discharge letters where the IE prototype discovers 29178 key terms or markers (in average 21,22 key words per PR). The distribution of these terminologies and temporal markers is the following one:

- 7092 occurrences of drug names were met in 1213 discharge letters,
- 6436 diagnoses are referred to in 1292 discharge letters,
- 1274 complains are recognised in 841 discharge letters and
- 7149 temporal markers were identified in 1374 discharge letters.

It turns out that the hospital PRs contain a significant amount of temporal information (about 33% of all extracted key terms and markers). Major errors in the automatic recognition of episode markers and/or their correct interpretation are due to:

- often use of *abbreviations* to denote intervals of time: 'y./'ye./'yea.' for 'year', followed or not by full stop or other punctuation mark. The same holds for 'month', 'week', 'day' etc.;
- sophisticated *prepositional phrases* for marking start, duration, cycle or interval: 'in 3 months', 'per 3 months', 'for 3 months' etc. Often the understanding is possible only after interpretation in the context;
- *ambiguity* in the use of temporal phrases, e.g. 'per day' may participate in the dosage of some drug and then 'day' should not be treated as a temporal marker, which signals a new episode;
- *reference to multiple moments* in one token, e.g. '5-6 years ago', '2001-02';
- *variety of tokens denoting the same time*, e.g. 'September 2009', 'm09.2009', 'M09.2009', 'Sept.2009', '09/2009' and so on;
- *fuzzy and non-determined references*, like e.g. 'few months ago';
- *anaphoric references* to previously introduced moments of time, e.g. 'since then', 'simultaneously', 'before', 'after', 'after that', 'several months after that', 'about the end of the year', 'at the same time' etc.;
- *spelling errors*.

The temporal markers are identified by an empirically-elaborated context-free grammar, which is run initially with simple rules and is under incremental development. The present recall in the temporal information recognition is about 57% and the precision is 84% (f-score 68%). There is some over-generation too, i.e. the system generates more temporal markers than appropriate. We consider our present achievements as work in progress, which has to be developed further.

Episodes are ordered in a sequence by a simple procedure which tries to calculate the actual date and constructs a list of linearly-ordered reference points. Note that the actual discharge letters in the hospital information system contain more dates than the anonymised version shown in Example 1. Our current research is focused on the investigation of the relative temporal references like “since then” in Example 1 which refers to the time of the previous episode. In general complicated time reasoners are needed to cope with the interpretation of temporal information in clinical narratives

but according to [8] these research tasks are at their embryonic stage. There is a need for theoretic models as well as large training corpora of annotated texts which are too expensive to construct. Temporal modelling of clinical texts can borrow theories from computational linguistics and contextualise them accordingly. A deeper study of the specific clinical discourse would help to acquire empiric rules for temporal reasoning in this domain.

4 Conclusion

Extraction of time-related information is a challenging research task. It is very important in medical informatics because time in medicine is essential to assess the speed of disease manifestation and development, the progress and effectiveness of treatments and so on. Success in extraction of temporal information would improve the clinical decision support systems. For the specific tasks of our projects, representing patient histories as simple CGs would enable comparison of cases. Unfortunately the automatic extraction of temporal information is a very difficult task and does not become simpler when the considerations are narrowed down from general NLP to medical texts. Much work is needed to develop the necessary corpora and conceptual resources which might support the implementation of advanced prototypes. We have achieved some progress due to already implemented components that deliver reliable information about the diagnoses and medication events. The elaboration of a component for relative temporal ordering of episodes is a target for our future work.

Acknowledgements. The research work presented in this paper is supported by grant DO 02-292 "Effective search of conceptual information with applications in medical informatics", funded by the Bulgarian National Science Fund in 2009-2012.

References

1. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove (2000)
2. Sauri, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML annotation guidelines, Version 1.2.1, (January 31, 2006), http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf.
3. Savova, G., Bethard, S., Styler, W., Martin, J., Palmer, M., Masanz, J., Ward, W.: Towards Temporal Relation Discovery from the Clinical Narrative. In: Proc. AMIA Annual Symposium, pp. 568–572 (2009)
4. Harkema, H., Dowling, J., Thornblade, T., Chapman, W.: Context: An Algorithm for Determining Negation, Experiencer, and Temporal Status from Clinical Reports. *J. Biomed. Inform.* 42(5), 839–851 (2009)
5. Hyun, S., Bakken, S., Johnson, S.B.: Markup of temporal information in electronic health records. *Stud. Health Technologies and Informatics* 122, 907–908 (2006)
6. Tcharaktchiev, D.: Hospital Information Systems. Sofia, Kama, (in Bulgarian) (2003)
7. Tcharaktchiev, D., Angelova, G., Boytcheva, S., Angelov, Z., Zacharieva, S.: Completion of Structured Patient Descriptions by Semantic Mining. To appear in 2nd Int. PSIP Workshop on Patient Safely through Intelligent Procedures in Medication. IOS Press, Paris (May 2011)
8. Zhou, L., Hripscak, G.: Temporal reasoning with medical data - a review with emphasis on medical natural language processing. *J. Biomed. Inform.* 40(2), 183–202 (2007)

Rigorous, and Informal?

David Love

Department of Computing, Sheffield Hallam University

Abstract. Most debates on conceptual structures have focused on (Hilbert) formal theories, due to the absence of stronger theoretical frameworks (despite nearly 80 years of searching). In this paper we look for new conceptual structures by looking for *weaker* theories. We contend both that weaker theories are possible, and that they offer useful alternatives to the better known formal theories.

1 Introduction

In 1925 John von Neuman showed that for a limited class of numerical systems, all objects of those systems can be extended from the null case [9], creating a self-referential, closed system. This idea was later extended by Alonzo Church and Stephen Kleene (amongst others), showing the creation of ‘non-numerical’ systems through a one-to-one association between those ‘non-numerical’ systems and limited classes of numerical ones [1][2][7]. Based on that earlier work by Church, Kleene claimed the structures identified through this process were general to *all* systems claimed as *formal* in the sense of David Hilbert [7, §60, §62]. Although this claim has yet to be proven, most mathematicians accept both the original claim of Church and later re-statements.

The details of these formal systems are unimportant[4]; of greater interest for this paper is the identified structure of formal systems and the consequences in terms of descriptive power. If we assume that every system of interest (natural, mathematical, computational or otherwise) follows the structure of formal systems, then the correspondence between the formal theory (i.e. the theory used to describe those formal systems) and the formal system is perfect. We would have no need of any other theory when describing any conceptual structure of interest.

Nonetheless, experience with computational systems (a close relative of formal systems) suggests that some other class of theory might also be of use in describing conceptual structures [6]. We also know from the work undertaken in the 1930s that formal theory is a *sub-class* of all possible mathematical theories [7, Chapter 5]. In other words, theoretical evidence suggests there should be a non-formal class of mathematical theory: and practical experience suggests such theories might be of interest to describing some conceptual structures.

¹ A good summary of the development of formal theory, and particularly the interrelationship between formal theory and computation, can be found in *The Universal Turing Machine — A Half Century Survey* [5], and especially the survey by Robin Gandy [4].

Thus far, however, the search for ‘non-formal’ theories has focused on *more* powerful theories: i.e. theories which re-state the conclusions of formal theory in even more universal terms [3,10,11]. Little work has been undertaken on *less* powerful theories: those which cannot describe even the systems described by formal theory [8], which are the focus of this paper.

2 Boundaries of Formal Theory

Theories described as formal in the sense of Hilbert have both a very clear structure, and a very clear boundary for theories describing those structures. Mathematically, we have a very good idea of the objects and structures described by formal theory. Therefore the easiest place to look for ‘non-formal’ theories lies just outside the boundaries of formal theory.

One of the key observations from the work of Kurt Gödel and Alonzo Church is the importance of a *one-to-one* correspondence between objects in the formal system. If the theory can maintain a one-to-one correspondence between objects of the system described *by* the theory and objects *in* the theory, and if neither manipulation of the objects of the theory, nor manipulation of the objects of the system can perturb that one-to-one correspondence, then the objects of the theory and the objects of the system become *equivalent* or *interchangeable*. In other words, the system *of* the theory, and the system *described* by the theory become one and the same — it makes very little sense to distinguish between the object of the theory and the theory itself, because any object can be described perfectly by the theory and and theoretical description has a perfect representation in the object system.

Where this one-to-one correspondence between theory and object holds, ‘boundaries’ between two systems become particularly important. If the two object systems share the same structure, then we would expect to be able to create two theory systems maintaining a one-to-one correspondence with the respective object system. The question now becomes, *can we go further?* Given the similarities of structure in both the object and theory system, can we describe *both* systems using a single theory?

The agreed answer to this question, accepting Church’s Thesis, is ‘yes’. We can create a *universal* formal theory, which is capable of describing any such system (or combination of systems) within a single theoretical structure. Further, we **assume** that since this single, universal, theory exists, as long as the object system(s) maintain a one-to-one correspondence with the theory system, *any* such object system can be described by this theory system. In other words, the precise details of the object system are entirely irrelevant: all that matters is that the structure of the system is maintained, and the one-to-one correspondence is preserved.

The existence of a single, unified and universal formal theory capable of describing any system maintaining a particular structure, gives formal theory an immense power and scope. But what if we want to develop less powerful theories?

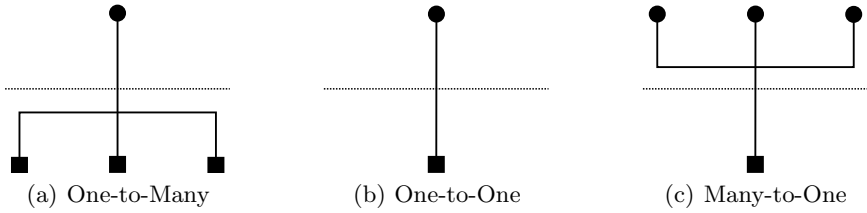


Fig. 1. Relationships Between Objects in Two Systems

We know from work undertaken on the limits of formal theory that a broader class (or classes) of mathematical theory should exist². So what type of system might *not* be describable using formal theory? If we knew the answer to that question, we would have the ability to be much more confident in our use of formal theory as the basic system for conceptual structures. For instance, if we knew that such systems were rare, then we could be confident that formal descriptions would adequately describe a large range of conceptual structures.

3 Nearly Formal Systems

An obvious first question is thus ‘*What systems cannot be described by formal theory?*’

In answer to this question we will refer to the boundaries of two simple systems illustrated in Figure 1. From the discussion in §2, it should be obvious that Figure 1(b) is perfectly describable by a formal theory. The one-to-one correspondence between objects in the system, even across the boundary, can be perfectly captured. But what of Figures 1(a) and 1(c)?

Here we must be careful. At first glance both Figures 1(a) and 1(c) would seem completely describable by formal theory. Further study, however, reveals that both Figure 1(a) and 1(c) are only describable using formal theory *in one particular case*. That case is the one we are most familiar with: where the relationships shown in Figures 1(a) and 1(c) can be broken down into a series of one-to-one relationships. **Only** in this case are the systems describable by formal theory: in *general* they are not.

The key to forming the special case is a *copy* operation. If we can create an equivalent *copy*, such that the overall behaviour of both the ‘original’ and ‘copy’ remains the same, then we can indeed reduce Figures 1(a) and 1(c) to Figure 1(b).

This process is illustrated for one of the systems in Figure 2. First we create a copy of the original object, then we create a one-to-one relationship between this copy and the object in the second system (Figure 2(b)): repeating this

² Assuming a radical re-statement of Hilbertian formal theory is not found. And nearly 60 years of intensive mathematical research found neither such a re-statement, nor evidence that such a re-statement should exist. Hence the confidence most mathematicians have in Church’s Thesis, even though a definitive proof is lacking.

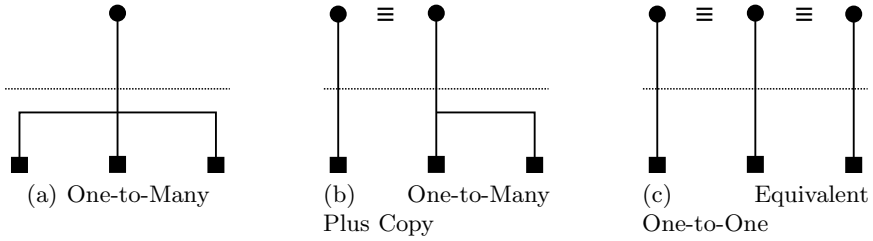


Fig. 2. Equivalent Object Systems

operation results in the final case, where we have re-established a one-to-one correspondence between all objects in the two systems.

So the potential descriptive power of formal theory relies on the existence of the copy operation in the object system. If we can copy objects reliably, then we can break apart relationships in the object system into a series of one-to-one relationships — and the resultant system(s) are then describable by formal theory.

If we have no such copy operation, the object system must remain *informal*. It may be describable by *some* theory, but that theory **cannot** be formal. And *if* we choose to use a formal theory to describe such a system, we must accept an inevitable breakdown between the system described by the formal theory and the system itself.

In such a case, *verification* of the theory system may be undertaken by formal theory. But *validation* of the relationship between the theory system and the object system **cannot** be undertaken by formal theory. Practically, then, other (non-formal) theories are required.

4 Conclusion

Basic assumptions in formal systems have become very familiar, and are often unquestioned. Practical questions have been raised about the suitability of formal theory for the *validation* of conceptual structures [6, Chapter 4]. Grounding such discussions in both theoretical and practical contexts, however, has proved difficult.

We propose that instead of searching for more powerful formal theories to undertake validation work, we look again at *ah hoc* informal theories. ‘Informality’ does not have to imply a lack of rigour. Rigorous informal theories should exist: and should not be too difficult to find.

References

1. Church, A.: A note on the Entscheidungsproblem. *Journal of Symbolic Logic* 1(1), 40–41 (1936)
2. Church, A.: An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58, 345–363 (1936)

3. Copeland, B.J.: Hypercomputation. *Minds and Machines* 12(4), 461–502 (2002)
4. Gandy, R.: The Confluence of Ideas in 1936. In: Herken [5], pp. 55–111 (1988)
5. Herken, R. (ed.): *The Universal Turing Machine — A Half Century Survey*. Oxford Science Publications (1988)
6. King, D.: *Parting Software and Program Design*. Ph.D. thesis, University of York (2004)
7. Kleene, S.C.: *Introduction to Metamathematics*, *Biblioteca Mathematica*, 6th edn. A Series, of Monographs on Pure and Applied Mathematics, vol. 1. Wolters-Noordhoff Publishing (1971)
8. Love, D.: From hypocomputation to hypercomputation. *International Journal of Unconventional Computing* 5(3–4), 339–367 (2009)
9. von Neumann, J.: *On the Introduction of Transfinite Numbers*. chap. (1923), pp. 346–354. Harvard University Press, Cambridge (1967)
10. Stannett, M.: Computation and hypercomputation. *Minds and Machines* 13(1), 115–153 (2003)
11. Teuscher, C., Sipper, M.: Hypercomputation: Hype or computation? *Communications of the ACM* 45(8), 23–24 (2002)

OpenSEA – Using Common Logic to Provide a Semantic Enterprise Architecture Framework

Jeffrey A. Schiffel¹ and Shaun Bridges²

¹The Boeing Company – Wichita Division
jeffrey.a.schiffel@boeing.com

²Open-SEA.org
Shaun.Bridges@Open-Sea.org

Abstract. The ISO 24707:2007 Common Logic Standards are designed to provide an abstract syntax for logic systems, in order to provide a commonality for interaction between different systems. OpenSEA uses these open standards as the foundation of a framework for Semantic Enterprise Architectures by combining them with definitions for enterprise architecture provided by The Open Group Architecture Framework. By using abstract syntax and semantics based on standards that are free to extend and specialize, OpenSEA provides the possibility for systems to interact by providing common generalisations for all conceptual structures that adhere to the framework.

1 Introduction

This paper introduces the architecture framework called Open Semantic Enterprise Architecture (OpenSEA), intended to enable the semantics of enterprise architectures for information access and knowledge generation. It uses tools that adhere to Common Logic (CL) [1] standards and unites the information through The Open Group Architecture Framework (TOGAF) [2].

As early as 1992, Sowa and Zachman wrote that rapidly improving price-performance and increasing rates of changes in information technology will continue [3]. Nineteen years later a workable solution remains unrealised. With data growth and information consumption growing at an explosive pace, organizations find it increasingly more difficult to sort out useful information from a sea of data. Similarly, the concept of a Semantic Web of linked data continues to be discussed and developed but, at its heart, relies on a broad range of logic languages and systems that use different ontologies and knowledge bases. Communication between two logic systems using different languages, logic, ontologies and terminologies is difficult and remains an obstacle to the notion of a global pool of information from which knowledge can be generated. In the last few years two standards have been agreed that can be combined to address all these problems simultaneously into the OpenSEA framework: TOGAF and CL.

TOGAF has become a widely accepted methodology for defining the business processes, objectives, data structures, systems and interfaces that enterprises of all sizes and sectors can use. Common Logic provides an abstract syntax for logic that can also be freely extended and adapted so any systems that have been built using the

standards can agree on a common understanding and communicate without loss of information. OpenSEA captures the abstract semantics of enterprise architectures using tools that adhere to Common Logic standards. TOGAF provides the Upper-Ontology and CL provides the Meta-Ontology. This set of abstract definitions, rules and terminologies may be adapted to create different templates, retaining a chain of generalisations and specialisations that all adhering definitions could follow to agree to a shared, abstracted understanding of each other.

2 Formalising the Enterprise Architecture

Different tools exist for different types of developers, but all suffer from the lack of a common language required to pull a system model together. Without a coherent system model, an integrated, interoperating system is not possible. This is especially true of dispersed subsystems, whether separated geographically or by functionality. It is difficult, if not impossible, in the current state of the tools market to have one tool interoperate with another tool [2, 4]. An enterprise will form a free market structure if the nature of the transaction between two organization units is simple, well defined, and universally understood [3]. In this case, the organization (or person) with work to assign would survey all possible workers to find one who is acceptable in terms of availability and cost. Formalising Zachman's Information System Architecture (in TOGAF) with Conceptual Graphs provides a means of capturing and expressing Peircian logic in a way that can be readily consumed by humans and predicate logic systems without compromising expression or logic [4]. Sowa is the driving force behind Common Logic; Zachman's Information System Architecture is influential in developing or evolving enterprise architectures [5]. Their investigations and proposals are central to the ideas behind OpenSEA.

2.1 A Common Toolset and a Common Language

Conceptual Graphs allow a common tool to capture and express information across the different sections of a model. A process flow diagram describes *how* an *enterprise* operates, a data flow diagram describes *how* a *system* captures this process and a master schedule document expresses *when* an *enterprise* undertakes the process. Traditionally these are all the domains of different experts using different tools, and thereby the barrier to full, clear, meaningful interaction between these domains is created. By using a common toolset and a common language this can be avoided. It is worth noting that seventeen years on The Open Group identified the same weakness: Tools such as online help are there specifically for users, and attempt to use the language of the user. Many different tools exist for different types of developers, but the tools suffer from the lack of a common language that is required to bring the system together. It is difficult, if not impossible, in the current state of the tools market to have one tool interoperate with another tool [2].

3 Gathering Knowledge across the Enterprise

Building on a common toolset and language allows an enterprise architect to see how changes in one model affect another. If the nature of dependency between enterprise

elements could be stored in some data repository along with the element models, it would be easier to assess the impact of a change to any one of the models. [3].

3.1 Semantic Business Process Management

In 2008 the International Research Forum discussed making semantically enriched service descriptions, and for services to become aware of their environment and their role within it: “If you want really to bring services on the Web then you need to have this kind of Semantic Web” [6]. The advantages of semantically enabling web services include an improved opportunity for interoperation [10], a means of providing accurate, meaningful descriptions, better rates of discovery [11], and improved security [12]. The key factors behind the weaknesses of the current service discovery are the quality of the syntactic data and variations in meaning of the metadata. Business Process Management is “the approach of managing the execution of IT-supported business operations from a managerial process view rather than from a technical perspective” [13]. Business process modelling is widely used, but is usually limited to simplified work-flows. Little attention is paid to capturing the over-riding reasons for modelling a process or constituent submodels. [14] proposed Semantic Business Process Management to improve the communication between the business requirements and the composite resources, systems and labour. Aligning with this idea, OpenSEA enables a semantic enterprise architecture that bridges the business needs and underpinning technology.

3.2 TOGAF and Common Logic

TOGAF [2] has been designed to be specialised for different industries via templates. It is widely used and accepted already. Using a language that is already widely used addresses one of the common problems facing the ontology engineer: acceptance. OpenSEA utilises the widely used language of TOGAF to provide a small vocabulary for a large variety of users. At the same time it uses generalisation relationships to extend an upper-ontology, whilst retaining a traceable link between all concepts that adhere to the framework. A short sample is provided later to show how this is done.

Common Logic [1] has been established so that the content of any system using first-order logic can be represented. It facilitates interchange of first-order logic-based information between systems. CL does not require a specific syntax; rather, it provides an abstraction of syntaxes to allow languages to be developed independently, while allowing them to be expressed without compromise or confusion. Just as TOGAF provides a scalable language, CL provides a scalable logical standard, allowing disparate data and knowledge bases to be combined within a distributed, boundaryless knowledge base.

4 The Extended Ontology

Central to OpenSEA is the notion of all definitions, facts and rules being specialised to suit a given enterprise architecture template, sector specific template, industry, organisation, team, process, or individual. This is achieved using the generalisation relations to create a web of connected concepts where all artefacts that adhere to the

framework can be traced back to a common generalisation. This simple approach is much the same way that the Internet can resolve host names across different domains, although in OpenSEA a concept or relation may be a specialisation of one or more generalisations. Sowa provided the following example for how this may be expressed in the CL compliant CGIF format.

```
CLIF:
  (forall ((R1 MonadicRelation) (R2 MonadicRelation) (x) (y))
    (if (and (GeneralizationOf R1 R2) (R2 x y)) (R1 x y)))

CGIF:
  [MonadicRelation @every *R1] [MonadicRelation @every *R2]
  [Entity: @every *x] [Entity: @every *y]
  [If (GeneralizationOf ?R1 ?R2) (#?R2 ?x ?y) [Then (#?R1 ?x ?y)]]
```

That is, for all monadic relations $R1$ and $R2$ and any x and y , if $R1$ is a generalization of $R2$ and $R2(x,y)$, then $R1(x,y)$. Once the `GeneralizationOf` statement is made then the type hierarchy can be listed as a simple collection of assertions:

```
CLIF;
  (and (GeneralizationOf Architect Business_Analyst)
    (GeneralizationOf Architect Information_Analyst)
    (GeneralizationOf Information_Analyst Data_Analyst)
    (GeneralizationOf Information_Analyst Technical_Analyst))
```

[8] built on this example to show how a doctor and patient could be seen to be specialisations of the same TOGAF definitions “Agent”, “PerformsTaskIn” and “Role”:

<pre>TOGAF: [Agent: @every *t] (PerformsTaskIn ?t [Role]) HealthCare (GeneralizationOf Agent Doctor) (GeneralizationOf Role Healthcare) Sales (GeneralizationOf Agent Salesman) (GeneralizationOf Role Sales)</pre>	<pre>Translated to the CLIF form: CLIF: [Doctor: @every *t] (PerformsTaskIn ?t [Healthcare]) And [Salesman: @every *t] (PerformsTaskIn ?t [Sales])</pre>
---	--

[7] drew on the TOGAF “attributes” used to define and document all artefacts that are contained within a TOGAF based architecture (specifically `ID`, `Name`, `Description`, `Category`, `Source`, and `Owner`). He proposed that these same attributes could be used to provide the information required to build the web of interlinks between the member concepts and relations (through “`Category`” connecting each entry to its parent entries), maintaining unique URLs to identify the entry and locate the metadata (`ID` and `Source`) including Definition (a CL definition of the “object,” i.e., how it is defined by other relations and concepts), `Name` (a “friendly” name), `Description` (a human readable free text) and `Owner` (the governing body to maintain the object). These metadata can be represented in CLIF as:

```
[Universal: @every *t]
(chrc ?t [Category])
(chrc ?t [Description])
(chrc ?t [ID])
(chrc ?t [Name])
(chrc ?t [Owner])
(chrc ?t [Source])
```

For example, the TOGAF definition of “data-entity” is “An encapsulation of data that is recognized by a business domain expert as a thing. Logical data entities can be tied to applications, repositories, and services and may be structured according to implementation considerations”. In OpenSEA this could be formalised in the upper ontology as:

```
CGIF:
[DEFINITION: " [DATA_ENTITY:*x1] [SERVICE:*x2]
(isAccessedAndServicedThrough ?x1 ?x2) "]
[NAME: Data Entity]
[CATEGORY: OpenSEA.org/ENTITY]
[SOURCE:
"http://www.opengroup.org/architecture/togaf9-
doc/arch/index2.html"]
[ID: "OpenSEA.org/DATA_ENTITY"]
[OWNER: OpenSEA]
[DESCRIPTION: "AN ENCAPSULATION OF DATA..."]
[DATA_ENTITY: *x1]
(chrc ?x1 OpenSEA) (chrc ?x1 Universal) (chrc ?x1
"http://www.opengroup.org/architecture/togaf9-
doc/arch/index2.html") (chrc ?x1 "AN ENCAPSULATION OF DATA")
(chrc ?x1 "OpenSEA.org/DATA_ENTITY") (chrc ?x1 Data Entity)
(chrc ?x1 ?x1] [SERVICE:*x2]
(isAccessedAndServicedThrough ?x1 ?x2) ")
```

In this example “DATA_ENTITY” is related to “SERVICE” with the “IsAccessedAnd-ServicedThrough” and is a specialisation of the “ENTITY” concept ([CATEGORY: OpenSEA.org/ENTITY]); the source is maintained as the Open Group web page that contained the definition in this case as this ; the ID (OpenSEA.org/DATA_ENTITY) is maintained by a body (OpenSEA in this case) and the TOGAF free text description is contained in the description.

4.1 Stability and Agility: Formalising TOGAF

Ontologies can be seen on a spectrum ranging from global ontologies that are massive, stable and involve significant effort to design to small, agile ontologies that are changed frequently yet relevant to very few [9]. OpenSEA embraces this spectrum through the notion of the “Owner” owning and governing the domain within which the specialisations are based. In this way the upper ontology would remain static and provide the stability required, yet small teams could adapt their ontology rapidly, absorbing and extending other objects when required. Conceptual Graphs can be used to show how the terms provided by TOGAF could be captured within a type hierarchy and basic definitions created from these concepts and relations.

5 Further Research

Future work will show a demonstration of an application in a specified domain, illustrating how development tools interoperate in OpenSEA. Work will also address the business case for OpenSEA, such as the costs and benefits of adoption, the organizational economics and cultural impacts to an organization when introducing a formal framework, and governance issues.

References

- [1] ISO/IEC 24707: Common Logic (CL): A framework for a family of logic-based languages (2007), <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html> (accessed November 1, 2009)
- [2] The Open Group: TOGAF Version 9, Zaltbommel. Van Haren Publishing, Netherlands (2009)
- [3] Zachman, J., Sowa, J.: Extending and formalizing the framework for information systems architecture. *IBM Syst. J.* 31(3), 590–617 (1992)
- [4] Sousa, P., Pereira, C., Vendeirinho, R., Caetano, A., Tribolet, J.: Applying the Zachman framework dimensions to support business process modelling. In: *Digital Enterprise Technology, (Session 3)* pp. 359–366. Springer, Heidelberg (2007)
- [5] Emery, D., Hilliard, R.: Every architecture description needs a framework: Expressing architecture frameworks using ISO/IEC 42010. In: *Joint Working IEEE/IFIP Conference on Software Architecture 2009 and European Conference on Software Architecture*, Cambridge, UK, September 14-17, pp. 31–40. IEEE Xplore, Los Alamitos (2009)
- [6] Heuser, L., Alsdorf, C., Woods, D.: *International Research Forum 2008*, 1st edn. Evolved Technologies Press, New York (2009)
- [7] Bridges, S.: *The Extent and Appropriateness of Semantic Enterprise Interoperability with TOGAF9 and ISO Common Logic*. Unpublished Dissertation. Sheffield Hallam University, Sheffield, UK (2010)
- [8] Bridges, S., Schiffel, J., Polovina, S.: *OpenSEA: A Framework for Semantic Interoperation Between Enterprises* (in press)
- [9] Berners-Lee, T., Kagal, L.: The fractal nature of the semantic web. *AI Magazine* 29(3), 29 (2008)
- [10] Bussler, C., Fensel, D., Maedche, A.: A conceptual architecture for semantic web enabled web services. *ACM Sigmod Rec.* 31(4), 24–29 (2002)
- [11] Sabou, M., Pan, J.: Towards semantically enhanced web service repositories. *Web Semant.: Sci., Serv. and Agents on the World Wide Web* 5(2), 142–150 (2007)
- [12] Alam, A.: Reasoning with semantics-aware access control policies for geospatial web services. In: *Proceedings of the 3rd ACM Workshop on Secure Web Services*, pp. 69–76 (2006)
- [13] Smith, H., Fingar, P.: *Business Process Management: The Third Wave*. Meghan-Kiffer Press, Tampa (2003)
- [14] Hepp, M., Roman, D.: An ontology framework for semantic business process management. In: *Proceedings of Wirtschaftsinformatik* (2007)

An Android Based Medication Reminder System: A Concept Analysis Approach

Ray Hashemi¹, Les Sears¹, and Azita Bahrami²

¹Department of Computer Science,
Armstrong Atlantic University, Savannah, GA, USA

Ray.Hashemi@armstrong.edu

²IT Consultation, Savannah, USA

Azita.G.Bahrami@gmail.com

Abstract. Failure to take medication as prescribed is one of the leading issues in health care today. A class of applications designed to remind people to take their medication as prescribed. While there are quite a few medication reminder systems available, they all require the user to enter the data manually. To simplify the use of these applications, development of a system is presented in this paper that allows the user to take a picture of his/her prescription medication labels and have reminders automatically generated for them. The system is developed for Android OS powered mobile devices and it employs image processing and a concept analysis approach. The accuracy for parsing dosing instruction text from images of medication labels and creating reminder events is over 90%.

Keywords: Medication Reminder System, Concept Analysis, Android OS, Dose Scheduling System, and Dosing Instruction Extraction.

1 Introduction

As cell phones evolve into networked mobile computing platforms, these devices bring conveniences and efficiencies into the daily lives of millions that could barely have been imagined just a few years ago. Devices such as Apple's iPhone, XDA based devices and those based on the Google Android Operating System, not only provide traditional cell phone services, but also applications designed to assist people in managing many aspects of their lives. One such class of applications is a prescription medication reminder system. Considering the fact that failure to take medication as prescribed is one of the leading issues in health care today [1][2][3], more than fifteen such applications currently exist.

In reviewing the available medication reminder applications, such as Wonderful Solutions' Medication Reminder[4], Pillbox Alert by Sartuga Software LLC[5], Get Pills by Rafal Rzepecki [6], and others, we noticed that each application requires the user to manually enter the prescription dosing information. We imagined that young adults may be too busy to want to take the time and that older adults may find it difficult and tedious to enter the information. If the input process could be automated,

perhaps these applications would be more widely adopted, and improve adherence to prescription medication dosing instructions. What if the user could simply take a picture of the label and have reminders generated automatically? With that thought in mind, the goal of this project is to demonstrate the ability to develop an automated prescription medication reminder system for Android based mobile devices.

Android based mobile devices present many unique challenges. Among them are the device’s limited processing power and available memory.

The rest of the paper is organized as follow. Concept definition for digitized images is introduced in section 2. Concept analysis is presented in section 3. Concept-row analysis is the subject of section 4. Calendar event creation is the subject of section 5. Experimental results are discussed in section 6. Conclusion and future Research are covered in section 7.

2 Concept Definition for Digitized Images

In a grayscale digitized image, if a set of pixels, P , have the same properties, t , then they make a *blub*. x_{min} , and x_{max} are the smallest and largest x coordinates among the x coordinates of pixels in P . y_{min} , y_{max} are the smallest and largest y coordinates among the y coordinates of pixels in P . A *concept* is the smallest encompassing box, that encompasses the blub and it is represented by the coordinates of its left-top and right-bottom vertices, V_{lt} and V_{rb} , where, coordinates for V_{lt} and V_{rb} are (x_{min}, y_{min}) and (x_{max}, y_{max}) , respectively. By having the coordinates of V_{lt} and V_{rb} , one can calculate the length of each side of the concept—Size, S . A concept may also include a set of pixels, q , that are not a part of the embedded blub. Therefore, a concept is a quadruple $C = (P, V, S, T)$, where:

$P = p \cup q$ is a set of pixels in which p is the set of pixels in the embedded blub and q is the set of pixels that are not a part of the blub. Thus,

$$p \cap q = \emptyset, |p| > 0, \text{ and } |q| \geq 0.$$

$V = \{V_{lt}, V_{rb}\}$. $V_{lt}(x_{min}, y_{min})$ and $V_{rb}(x_{max}, y_{max})$ are the coordinates of the left-top and right-bottom vertices.

$S = \{s_w, s_h\}$ and s_w and s_h are the size of the width and height of the concept expressed in number of pixels.

T = a set of properties that pixels in p have in common.

Figure 1 displays an image before and after concepts are identified.

Two concepts, $C_i = (P_i, V_i, S_i, T_i)$ and $C_j = (P_j, V_j, S_j, T_j)$, are *horizontally* related if: $T_i = T_j$, $S_i \approx S_j$, and $horizontalDistance(C_i, C_j) \leq \tau$, where τ is a threshold. Two concepts, $C_i = (P_i, V_i, S_i, T_i)$ and $C_j = (P_j, V_j, S_j, T_j)$, are *vertically* related if: $T_i = T_j$, $S_i \approx S_j$, and $verticalDistance(C_i, C_j) \leq \tau$, where τ is a threshold.

If for concept C one of the following conditions is true, then the concept is considered an *outlier concept* and it is filtered:

- $s_h > \hat{s}_h + d_h$ and $s_w > \hat{s}_w + d_w$, where \hat{s}_h and \hat{s}_w are the average of all concepts’ s_h and s_w . $d_h \geq \sigma_h$ and $d_w > \sigma_w$, where σ_h and σ_w are standard deviations for all concepts’ s_h and s_w , respectively.

2. $g > \hat{g} + d_g$, where, g is the average gray level of the concept and \hat{g} is the average gray level of the concept and $d_g \geq \sigma_g$, where σ_g is the standard deviation of all concepts' gray levels.
3. Concepts with minimum distance from the top or bottom edges.
4. Concept for which $q = \emptyset$.

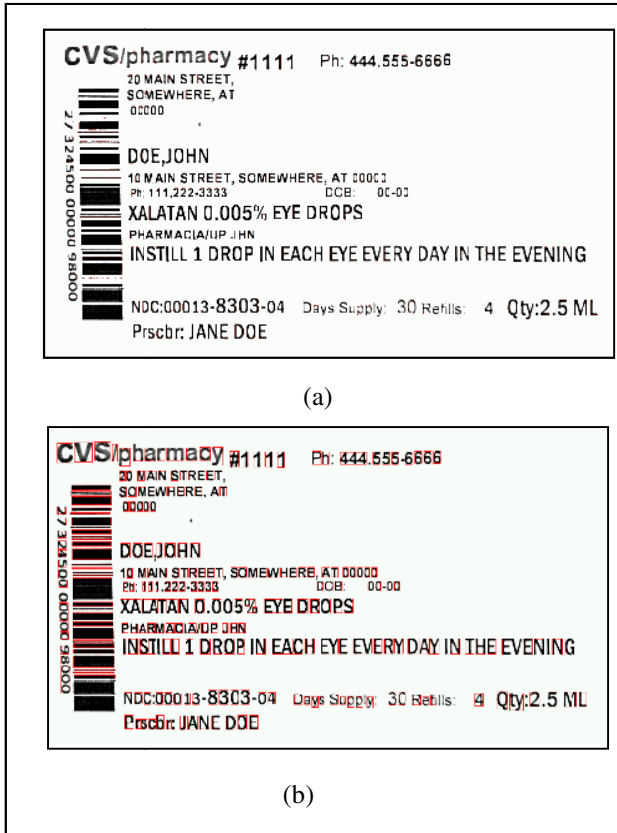


Fig. 1. An image before and after its concepts are identified: (a) An image, (b) The image and its concepts

A set of concepts make a *concept-row*, R , if every two adjacent concepts in the set are horizontally related and a set of concepts make a *concept-column*, K , if every two adjacent concepts in the set are vertically related. A concept-row that is made up of n concepts is defined as, $R = (P, V, S, T)$, Where,

$$P = \{ \{p_1, \dots, p_n\} \cup \{q_1, \dots, q_n\} \}$$

$$V = \{ v_{lt}(x, y), v_{rb}(x', y') \}, \text{ where: } x \text{ and } y \text{ are } \min(x) \text{ and } \min(y) \text{ among all } x \text{ and } y \text{ coordinates in } \{ v_{lt}(x_1, y_1), \dots, v_{lt}(x_n, y_n) \}, x' \text{ and } y' \text{ are } \max(x) \text{ and } \max(y) \text{ among all } x \text{ and } y \text{ coordinates in } \{ v_{lt}(x_1, y_1), \dots, v_{lt}(x_n, y_n) \}.$$

$S = \{s_w, s_h\}$ and s_w and s_h are the size of the width and height of the concept-row using V and it is expressed in number of pixels, and $T = T_1$.

A concept-column is also defined as $K = (P, V, S, T)$, where P, V, S , and T have the same meaning as they have in a concept-row, R .

Let $\mathbf{R} = \{R_1, \dots, R_m\}$ be a set of concept-rows. If every two adjacent concept-rows are horizontally related to each other, then collectively make a *super concept*. This is also true for a set of concept-columns of $\mathbf{K} = \{k_1, \dots, k_n\}$. Pixels that are not a part of a concept and they are not a part of the background also change to background.

Super concepts that are horizontally or vertically related make a new super-concept. However the new super concept belongs to a higher level of concept hierarchy of the image.

3 Concept Analysis

We start with introducing some terminologies and then presenting an algorithm for the concept analysis of digital images.

Inner concept for a given concept $C = (P, V, S, T)$ is $C' = (P', V', S', T')$ and it is defined as follows:

$$\begin{aligned}
 &P' \subset P. \\
 &V' = \{v'_{lt}, v'_{rb}\}, \text{ where: coordinates for } v'_{lt} \text{ and } v'_{rb} \text{ are } v'_{lt}(x_{min} + \delta, y_{min} + \delta) \\
 &\quad \text{and } v'_{rb}(x_{max} - \delta, y_{max} - \delta) \text{ and } \delta = \min(0.3*s_w, 0.3*s_h) \\
 &S' < S. \text{ That is, } s_w - s'_w = 2\delta \text{ and } s_h - s'_h = 2\delta. \\
 &T' = T.
 \end{aligned}$$

A blub, its embedded concept and its inner concept are illustrated in Figure 2(a-c). A blub within a concept may be situated in different ways. Let us assume that a part of the blub is sandwiched between two horizontal top sides of its concept and inner concept. If the number of pixels in the longest horizontal sequence of pixels in the sandwiched part is greater than a threshold then the top of the blub is not encompassed by the inner concept, thus, it is coded as '1'; Otherwise, it is coded as '0'. As a result, the situation of a bulb in its concept could be coded by a 4-digit binary number and it is referred to as *concept signature*. Signature of concept in Figure 2.a is shown in Figure 2.d.

Concept's Mesh is a grid made up of a small number of horizontal and vertical lines imposed on a concept, Figure 2.e.

Concept's Vector for concept $C = (P, V, S, T)$ is a vector of integer numbers such that the i -th value in the vector stands for the number of times that i -th line of the concept's mesh passes through dark pixels. If the concept's mesh is made up of five horizontal and five vertical lines, then the concept vector of C has ten values, Figure 2.f.

Database of Concepts' Vector is a database of expected concept's vector for the 36 symbols. Although it is not required by regulation [7], for every label examined, the dosing instruction was a plain sans serif font of only capital letters A through Z and the numerals 0 through 9, the total of 36 symbols. In addition, the database includes signature for each symbol.

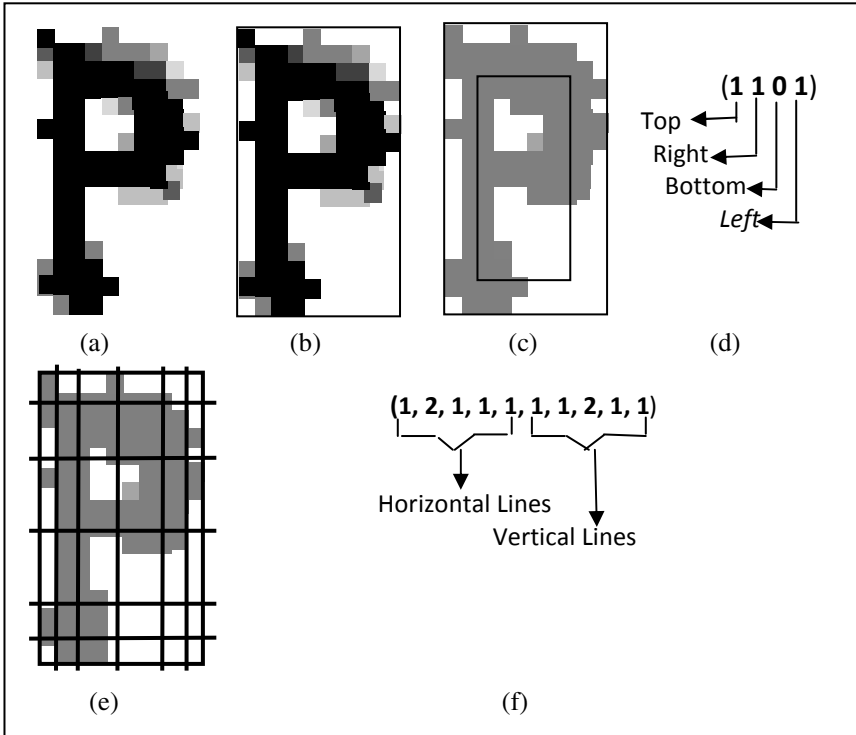


Fig. 2. Blub and Concept: (a) a blub, (b) concept of the bulb, (c) the concept and its inner concept, (d) concept's signature, (e) concept's mesh with five horizontal and five vertical lines, and (f) concept's vector

3.1 Concept Mapping

Let W and Sig be the vector and signature for the concept $C=(P, V, S, T)$. The Euclidian distance of the W from the entries in the database with the same signature is measured. C is the same as the concept with the smallest distance. Therefore, the blub in concept C is recognized. The details are presented by the following algorithm.

Algorithm: Concept Mapping

Given: Concepts in a digitized image. A database, D , of concepts vector of the capital letters and digits.

Objective: Map every concept onto a character in the database.

Repeat for each concept C in the image.

Determine the concept signature, Sig ;

Determine the concept vector, W ;
 $D' \leftarrow$ All the entries of the database that their signatures are equal to Sig ;
 Repeat for each entry in D'
 Calculate the Euclidian distance of W from the entry;
 C is the same as the entry in D' with the smallest Euclidian distance;
End;

In this stage, another level of filtering of the concepts takes place based on the ratio of the width to height of the concept. Concepts that contain letter “I” or digit “1” are exception. Therefore, a concept is filtered if the ratio is different from a predefined value, unless the concept mapped on either “I” or “1”.

4 Concept-Row Analysis

After the concept mapping is complete, the concept-rows are identified. Each concept-row represents a token. However, the token may or may not be an expected token to be present on a medical label dosing instructions. The accuracy of concept-rows is essential for reaching our goal. To increase the accuracy, each concept-row is compared to a known lexicon, Figure 3, expected to be present on the label of instructions.

Prescription Medication Label Lexicon							
Word	Type	Word	Type	Word	Type	Word	Type
APPLY	Verb	TABLETS	Item	ONE	Number	5	Number
TAKE	Verb	DOSE	Item	TWO	Number	6	Number
SWALLOW	Verb	DOSES	Item	THREE	Number	7	Number
EAT	Verb	LOZENGE	Item	FOUR	Number	8	Number
INSTILL	Verb	LOZENGES	Item	FIVE	Number	9	Number
CONSUME	Verb	EVERY	Adj.	SIX	Number	DAY	Period
DEVOUR	Item	EACH	Adj.	SEVEN	Number	DAYS	Period
CAPSULE	Item	ONCE	Adj.	EIGHT	Number	HOUR	Period
CAPSULES	Item	TWICE	Adj.	NINE	Number	WEEK	Period
PILL	Item	TIMES	Adj.	1	Number	WEEKS	Period
PILLS	Item	DAILY	Adj.	2	Number		
DROP	Item	HOURLY	Adj.	3	Number		
TABLET	Item	A	Number	4	Number	FOR	Prepos.

Fig. 3. Dosing Instruction Vocabulary

For the actual comparison, the Levenshtein Distance method is used [10]. The Levenshtein Distance method compares the number of simple operations of insertions, deletions and substitutions to compute what is known as the *edit distance*. The edit distance is a numerical value of the number of times the characters in one word are different in one of the ways mentions above from another word. The known word with the smallest edit distance from the unknown word is considered the most likely match. The actual algorithm used is an adaptation of the Wagner-Fisher dynamic programming algorithm for computing edit distance and this project uses the Apache commons-lang string utility package [11].

5 Calendar Event Creation

The super concepts of the concept-rows are identified. Contextual logic is applied to each super concept to parse out the frequency and duration using the vocabulary given in Figure 3. This figure not only includes the expected lexicon on a label, it also includes the role of the word in the text.

For creating calendar event, frequency and duration are needed. Both are determined using the following algorithm. The parameter “Code” is either “F” for requesting determination of “frequency”, or “D” for requesting determination of “duration”.

Algorithm Frequency_Duration (Code)

Given: The super concepts of a digitized medication label’s image and Prescription Medication Label Lexicon.

Objective: Determine the frequency or duration.

If Code = “F”

Then pattern = <number>[<item>]<adjective><time_period>

Else pattern = <for> <number> < time_period>;

Repeat for each super concept of the image.

If (pattern exist) Then return (<number>) ; break;

End;

Using this information, calendar events are created for the given frequency and duration. For example, the following super concept: “TAKE 2 PILLS EVERY DAY FOR THE NEXT 10 DAYS” has a frequency of twice a day and duration of ten days. In this example, two calendar events are added to the mobile device’s calendar each day, starting with today, for the next ten days for a total of 20 calendar events. The Android calendar API is not documented or published and was implemented with information from [12].

6 Experimental Results

The particular device used for this project was the Motorola Droid. It has an ARM Cortex A8 processor running at 550Mhz, 256 MB of internal RAM and a 5 megapixel camera. The Android OS provides a Java Software Development Kit for creating custom applications which further constrains the application environment by limiting the Java heap size to a maximum of 17MB.

We measured the system’s ability to create calendar events to remind the user to take their prescription medication. We used the following three criteria for evaluation:

- Accuracy of the dosing instructions in the calendar events.
- Accuracy of the frequency of the dosing instructions.
- Accuracy of the duration of the dosing instructions.

The label represents the dosing instructions that are given as part of the receipt when filling a prescription that are not affixed to the prescription bottle itself.

	Dosing Instruction	Dose Frequency	Does Duration	Average
Accuracy	93%	97%	93%	94.4%

Fig. 4. Results

Thirty labels were tested with variations of dosing instruction under the same illumination and the mobile device was oriented in such a way as not to produce any shadows on the label. The results are presented in Figure 4.

7 Conclusion and Future Research

The results of the accuracy of the prescription medication reminder system reveals that using pictures of medication labels is a feasible of entering dosing instructions for the patient.

It was noted during testing that proper and even illumination was required to obtain consistent results. In addition, the camera on the mobile device is better suited for taking snapshots and is difficult to hold steady when taking close-ups of the labels. These issues could be remedied by providing an inexpensive stand to hold the camera and the medication label in the proper position.

As future research, separating connected blubs is in progress. This effort ultimately reduces the number of unwanted concepts, improves concept-rows, and influences, both concept and concept-row analysis.

References

- [1] Conn, V.S., Hafdahl, A.R., Cooper, P.S., Ruppar, T.M., Mehr, D.R., Russell, C.L.: Interventions to Improve Medication Adherence Among Older Adults: Meta-Analysis of Adherence Outcomes Among Randomized Controlled Trials. *The Gerontologist* 49(4), 447–462 (2009)
- [2] Chia, L.R., Schlenk, E.A., Dunbar-Jacob, J.: Effect of personal and cultural beliefs on medication adherence in the elderly. *Drugs Aging* 23(3), 191–202 (2006)
- [3] Neupert, S.D., Patterson, T.R., Davis, A.A., Allarie, J.C.: Age Differences in Daily Predictors of Forgetting to Take Medication: The Importance of Context and Cognition. *Experimental Aging Research* (in press)
- [4] Wonderful Solution, <http://wonderfulsolution.blogspot.com> Page accessed (November 12, 2010)
- [5] Sartuga Software. Pillbox Alert, <http://pillboxalert.com> Page Accessed (November 12, 2010)
- [6] Rzepecki, R.: Get Pills, <http://sourceforge.net/projects/getpills> Page accessed (November 12, 2010)
- [7] Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60(2), 91–110 (2004)
- [8] Navaro, G.: A Guided Tour to Approximate String Matching. *ACM Computing Surveys* 33 (1999)
- [9] Apache Commons. “commons-lang”, <http://commons.apache.org/lang> Page accessed (November 14, 2010)
- [10] Conder, S., Darcey, L.: Working with the Android Calendar. *developer.com* (2009), <http://www.developer.com/article.php/3850276>

System Decomposition for Temporal Concept Analysis

David Luper¹, Caner Kazanci², John Schramski³, and Hamid R. Arabnia⁴

¹ Department of Computer Science, University of Georgia, Athens, GA, USA
luper.david@gmail.com

² Department of Mathematics, University of Georgia, Athens, GA, USA
caner@uga.edu

³ Department of Engineering, University of Georgia, Athens, GA, USA
jschrams@uga.edu

⁴ Department of Computer Science, University of Georgia, Athens, GA, USA
hra@cs.uga.edu

Abstract. Temporal concept analysis is an extension of formal concept analysis (FCA) that introduces a time component to concept lattices allowing concepts to evolve. This time component establishes temporal orderings between concepts represented by directional edges connecting nodes within a temporal lattice. This type of relationship enforces a temporal link between concepts containing certain attributes. The evolution of concepts can provide insight into the underlying complex system causing change, and the concepts evolving can be seen as data emission from that complex system. This research utilizes models of complex systems to provide frequency histograms of activity in well-defined sub-networks within a system. Analyzing systems in this way can provide higher levels of contextual meaning than traditional system analysis calculations such as nodal connectedness and throughflow, providing unique insight into concept evolution within systems.

Keywords: Data Mining, Systems Analysis, Knowledge Extraction, Graph Mining, Sequence Mining.

1 Introduction

FCA is a principled way of deriving ontological structures from a set of data containing objects and attributes [1]. It establishes concepts from collections of objects exhibiting a certain group of attributes. In a database void of time, these concepts appear without change, however, in temporal concept analysis [2][3][4] time is taken into account and concepts can evolve to take on different meaning. As an example take a database where people are objects possessing the attributes of either young or old. If time steps are present in this database a person p could have entries at different time steps, $t1$ and $t2$, where $p t1$ is labeled young and $p t2$ is labeled old. This would highlight that people objects can morph from young to old over sequential time steps. This serves to establish a temporal link from time step t to $t + 1$ between the attributes young and old. This is a simple example where a one way transition from young to

old occurs, but temporal relationships between attributes can be far more complex involving a sophisticated network of transitions encompassing very complex systems of interaction. The underlying system causing the evolution can be modeled in an adjacency matrix of transition probabilities from one attribute to another. This adjacency matrix can be seen as a kind of Markov model outlining the attribute transition probabilities for objects in a concept lattice. A temporal concept attribute model (TCAM) is a modeling of a complex system where nodes in the system are attributes and flows in the system are probabilities that objects possessing an attribute at time t will possess another attribute at time $t + 1$.

Modeling complex systems occurs across a wide variety of scientific disciplines [5][6][7][8][9] including economics, computer science, ecology, biology, sociology, etc. Models (networks) help understand systems that are too complex for deterministic behavior to be recognized, like a person's movement (i.e. tracking a person's GPS data)[10][11], ecosystem food webs [12], rhythm patterns within music [13] or financial volatility within economic systems [14]. Network analysis historically involves the evaluation of network structure and function through the calculation of such metrics as nodal connectedness or compartmental and total system throughflow. This approach is helpful, but lacks the ability to analyze groupings of connected nodes interacting with each other. Perhaps a more complete method for analyzing a network includes taking into account a node's sphere of influence within defined sub-networks. This approach can serve to contextualize the behavior of specific nodes providing a more complete understanding of their role in the network. The goal of this research is to quantify a measure of flow for nodal groupings, signifying levels of importance in a network. The two main obstacles include structurally decomposing a network into groupings of nodes (sub-networks) and calculating the amount of flow that passes through these derived subsets.

2 Temporal Concept Attribute Models (TCAM)

An attribute transition model can be constructed from a time stamped database by isolating all instances of object transition from one attribute to another over a given window of time in the database. Strategies for constructing this model include, but are not limited to, the following method. First a group of attributes A must be defined where A contains all the attributes being modeled in the TCAM over a defined time window T . For completeness A may need a null attribute value representing an object having an attribute in the model at time step t and then having no attribute from the model at time step $t + 1$. Once A and T are defined a set of objects O must be assembled that will be used to build the TCAM. O can be any logical grouping of objects. With A , T and O defined all entries in the database for each object in O over the time window T must be enumerated in time step order. For any object being labeled with attribute a_1 at time step t and a_2 at time step $t + 1$ a frequency of occurrence value for the edge between a_1 and a_2 in the TCAM is incremented by one. In this example we are seeking only transitions and constrain the methodology by saying an attribute is not permitted to have an edge looping back to itself. If an object stays in possession of a particular attribute for multiple time steps nothing is modified in the transition matrix. Once every time step in T for every object in O is enumerated the transition

matrix can be normalized to reflect the probability of transition between attributes. As an important note the set of attributes in A must never appear in any combination at the same time step for a single object. If attributes a_1 and a_2 both appear at time step t for object o a new element must be added to A called a' . This new element represents the occurrence of both attributes at the same time step. This maintains consistency in A such that elements of A are attribute state groupings that objects transition in an out of.

3 Network Decomposition

This research will pursue a computational algorithm to determine the partial through-flow for meaningful groupings of weakly connected nodes (sub-networks) in a complex system (network). Decomposing a network involves both structural and functional steps that will now be introduced, but flow vectors, sub-network vectors, and the sub-network matrix must be introduced first. A flow vector can be constructed from a network if every edge in the network is labeled with a sequential integer and an edge's magnitude of flow is stored at its respective index (Fig. 1). A sub-network vector (Fig. 2) is a binary vector with a size equal to the flow vector, where for any edge used in the sub-network a 1 is stored at the corresponding vector index and

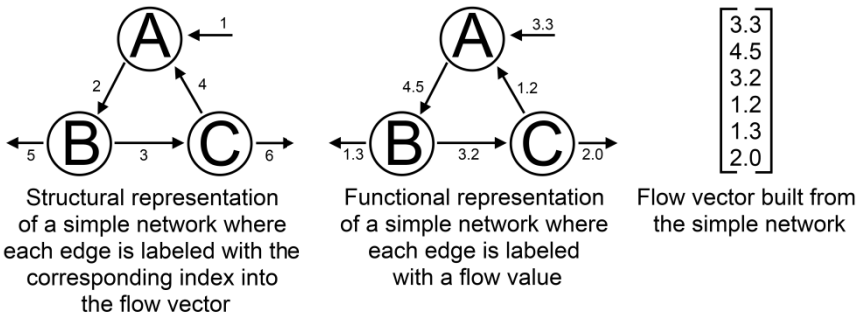


Fig. 1. Flow vector

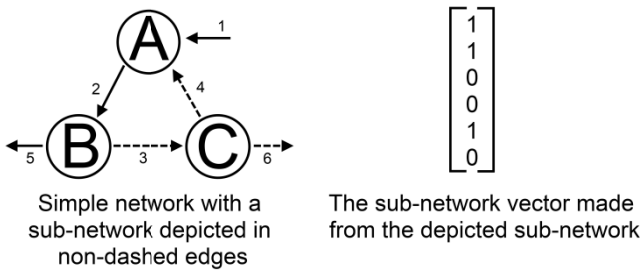


Fig. 2. Sub-network vector

all other elements are zero. A sub-network matrix is a matrix of n rows and m columns where n equals the number of edges in the network and m equals the number of decomposed sub-networks. Each column of the sub-network matrix is a sub-network vector.

3.1 Structural Decomposition

Structural decomposition is the process of finding all sub-networks within a network. The definition of a sub-network relies on understanding key concepts of network decomposition and throughflow.

- The network, and each sub-network, are assumed to be at steady state (input in equals input out, compartmental storage is ignored).
- Edges in network decompositions are unweighted. The sub-network matrix is the output from a network decomposition, and any flow across the edges is abstracted into coefficient terms pursued later.
- For a sub-network to be dissected from the rest of the network it must be self-sustaining, ensuring that any agent traversing a sub-network will remain in that sub-network without getting lost to some other sub-network. This effectively binds an agent solely to a particular sub-network.
- A constraint is placed on the decomposition of a network, that once decomposed the network (including flows) must be able to be recomposed. This constraint can be met by applying the derived coefficients (discussed later) to their respective columns in the sub-network matrix. Then the rows of the sub-network matrix can be summed to produce the original network flow vector.
- Because a sub-network is unweighted and at steady state, each node in a sub-network must have only one input and one output. If nodes in a sub-network had multiple inputs or multiple outputs an agent traversing the sub-network would have to choose which path to take and the sub-network could not remain unweighted.

Accounting for these concepts defines a sub-network as any path through the network that starts where it ends, has no duplicate nodes and each node has exactly one input and one output. This is the definition of a simple cycle.

A structural decomposition algorithm can now be outlined. Let M be an adjacency matrix for a network. If there are inputs to or outputs from the network, an additional start/stop state (compartment) must be added and its edges must be listed in M . The decomposition algorithm takes M as input and places every network compartment into a path of length 1. All the paths are placed in a queue. Until the queue is empty path p is removed from the queue and inspected to see if it is a simple cycle. If p meets this criterion it is output as a sub-network. After inspection, paths of length $len(p) + 1$ are constructed using every edge stored in M for the last node in p . Any new simple path (one that has not been created prior to this and has no duplicate nodes) is placed on the end of the queue and the loop continues.

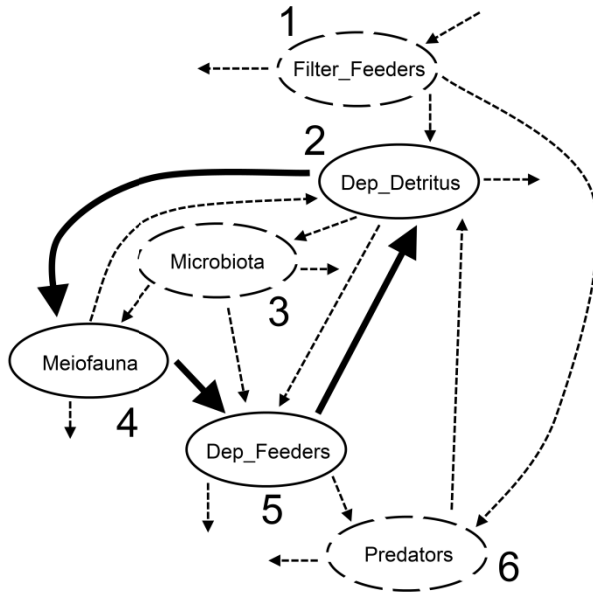


Fig. 3. Sub-network (compartments 2-4-5) of an ecological model depicting energy flow in an oyster reef habitat (Dame and Patten 1981)

3.2 Functional Decomposition

Functional decomposition is the process of assigning magnitude values to the sub-networks identified in the structural decomposition phase. These magnitude values can represent frequency of occurrence values for sub-networks, or portions of total system throughflow each of sub-networks is responsible for. This research presents a computational framework that analyzes simulated data from a network model and computes a coefficient for each sub-network in a system. It requires as input a weighted adjacency matrix of transition probabilities between compartments in a network. The output is a coefficient vector.

After structural decomposition, a data distribution containing pathways agents took through the system can be simulated using the transition probability matrix, and subsequently analyzed. This allows a histogram to be computed tracking the sub-networks used to interpret each of the data instances. Interpreting a pathway means viewing it as a combination of sub-networks rather than a combination of individual nodes [15], and an interpretation vector is the result of interpreting a pathway. It is a vector of length n , where n is the number of sub-networks in the decomposed system. Each index in the interpretation vector represents a particular sub-network, and each value in the vector represents the number of times a sub-network was used in an interpretation. This methodology calculates an interpretation vector for each pathway in a distribution and adds it to a histogram to keep track of how many times each sub-network is used throughout interpretation of the entire distribution of data.

During interpretation of data instances a problem can arise that certain pathways through a network can be interpreted using different sets of sub-networks. An example of a path with multiple interpretations is seen in Fig. 4. If all interpretations for a given pathway are added to the histogram, pathways containing multiple interpretations would have a greater impact on the histogram. Conversely, if only one interpretation is used, sub-networks can be viewed as being responsible for more or less flow depending on a pathway’s chosen interpretation. Ultimately this causes multiple correct coefficient vectors to exist, and they constitute a solution space of possible coefficient vectors. This research uses an averaging technique to deal with pathways that contain multiple interpretations. Every interpretation vector for a pathway is added to the histogram after dividing each of them by the total number of interpretations for that pathway. This gives equal weight to every possible interpretation for a pathway, while adding the equivalent of a single interpretation to the histogram.

A Particle Pathway through the Oyster Reef Energy Model

→ 1 → 2 → 3 → 4 → 5 → 2 → 4 → 2 → 3 → 4 →

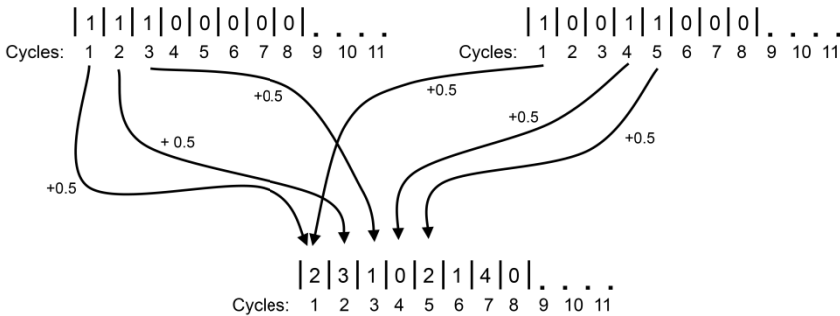
Two Interpretations of the Example Pathway

Cycle 1 (1 - 2 - 3 - 4)
 Cycle 2 (4 - 5 - 2)
 Cycle 3 (4 - 2 - 3)

Cycle Assignment : 1 1 1 2 2 2 3 3 3 1
 Pathway : 1 2 3 4 5 2 4 2 3 4

Cycle 1 (1 - 2 - 3 - 4)
 Cycle 4 (2 - 3 - 4 - 5)
 Cycle 5 (2 - 4)

Cycle Assignment : 1 4 4 4 4 5 1 1 1
 Pathway : 1 2 3 4 5 2 4 2 3 4



Sub-Network Histogram

Fig. 4. An overview of how to determine coefficient vectors

4 Discussion

Decomposing a TCAM and computing coefficients for its sub-networks is a novel way to analyze complex systems behind conceptual evolution. However, much work

is needed to determine useful ways of applying this methodology. Using the information for data mining holds potential to be very transformative and is an interesting way of allowing machines to understand concept evolution. One interesting usage of this methodology would be to compare different windows of time within a time step database using the computed histogram in a distance metric. Complex systems in different states (i.e. they are modeled with different transition probability matrices) would have different coefficients, and the Euclidean distance between those coefficients in a particular feature space could be an invaluable source of information for data mining and knowledge extraction.

The coefficients computed by this methodology hold more information than the transition probabilities alone because they reflect all relationships each of the transition probabilities are involved in. Research needs to be devoted to finding ways to exploit the additional information embedded in this representation of system activity.

5 Conclusion

A TCAM is transition probability matrix that models attribute transition within temporal concept analysis. This work has shown how to construct a TCAM from a time stamp database. A TCAM models complex systems driving concept evolution within temporal concept analysis. A methodology was presented for decomposing a TCAM both structurally and functionally. First, an exhaustive set of unique groupings of sub-networks from the TCAM are found. After this, magnitude coefficient values are computed detailing the frequency of occurrence for each sub-network in simulated data from TCAM. This methodology can be seen as a transform that takes as input a transition probability matrix, and outputs a histogram of magnitude values representing frequency of occurrence for each simple cycle in the system. Stated another way, this transform takes a sequential grouping of compartment nodes in a time series (a pathway through a complex system) and maps it out of the temporal domain to a domain representing frequency of occurrence for each sub-network in the system. Viewing system activity in this different domain allows new information about the system to be ascertained specifically related to its decomposed set of sub-networks.

Research is being devoted towards two applications of the proposed methodology. First, effort is being made towards identifying important relationships within ecological models for differentiating seasonal variance. This research uses support vector machines to classify histograms computed from seasonal variations of ecological models. Second, this methodology is being utilized to provide impact analysis on an ecosystem. For this work an ecological model with two different sets of flow values (pre and post impact) are compared and a novel distance metric is outlined, applying the proposed methodology to find how much each of the sub-networks in the model is affected by some impact on the system.

This methodology has great potential, and the diverse range of problems to which it can be applied is a major strength of the work. It holds great potential for systems analysis because it provides a new domain in which system activity can be viewed.

References

1. Priss, U.: Formal concept analysis in information science. *Annual Review of Information Science and Technology* 40, 521–543 (2006)
2. Neouchi, R., Tawfik, A.Y., Frost, R.A.: Towards a Temporal Extension of Formal Concept Analysis. In: *Proceedings of the 14th Canadian Conference on Artificial Intelligence*, Ottawa, Ontario (2001)
3. Wolff, K.E.: Interpretation of Automata in Temporal Concept Analysis. In: Priss, U., Corbett, D.R., Angelova, G. (eds.) *ICCS 2002. LNCS (LNAI)*, vol. 2393, p. 341. Springer, Heidelberg (2002)
4. Wolff, K.E.: Temporal Concept Analysis. In: MephuNguifo, E., et al. (eds.) *ICCS-2001 International Workshop on Concept Lattices-Based Theory, Methods and Tools for Knowledge Discovery in Databases*, pp. 91–107. Stanford University, Palo Alto (2001)
5. Batagelj, V., Mrvar, A.: Pajek Program for large network analysis. *Connections* 21(2), 47–57 (1998), Project home page at, <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>
6. Smith, D.A., White, D.R.: Structure and Dynamics of the Global Economy: Network Analysis of International Trade, 1965-1980. *Social Forces* 70, 857–893 (1992)
7. Wellman, B., Salaff, J., Dimitrova, D., Garton, L., Gulia, M., Haythronwaite, C.: Computer networks as social networks: collaborativework, telework and virtual community. *Annu. Rev. Sociol* 22, 213–238 (1996)
8. Ammann, P., Wijesekera, D., Kaushik, S.: Scalable, Graph-Based Network Vulnerability Analysis. In: *Proceedings of CCS 2002: 9th ACM Conference on Computer and Communications Security*, Washington, DC (November 2002)
9. Thibert, B., Bredesen, D.E., del Rio, G.: Improved prediction of critical residues for protein function based on network and phylogenetic analyses. *BMC Bioinformatics* 6, 213 (2005)
10. Luper, D., Chandrasekaran, M., Rasheed, K., Arabia, H.R.: Path Normalcy Analysis Using Nearest Neighbor Outlier Detection. In: *ICAI 2008*, pp. 776-783 (2008); In: *Proc. of International Conference on Information and Knowledge Engineering (IKE 2008)*, Las Vegas, USA, July 14-17, pp. 776-783 (2008) ISBN #: 1-60132-075-2
11. Luper, D., McClendon, R., Arabia, H.R.: Positional Forecasting From Logged Training Data Using Probabilistic Neural Networks. In: *Proc. of International Conference on Information and Knowledge Engineering (IKE 2009)*, Las Vegas, USA, July13-26, pp. 179–189 (2009) ISBN # for set: 1-60132-116-3
12. Rohr, R.P., Scherer, H., Kehrl, P., Mazza, C., Bersie, L.: Modeling Food Webs: Exploring Unexplained Structure Using Latent Traits. *The American Naturalist* 176(2), 170–177 (2010)
13. Temperley, D.: Modeling Common - Practice Rhythm. *Music Perception: An Interdisciplinary Journal* 27(5), 355–376 (2010)
14. Marcelo, C.: Medeiros and Alvaro Veiga, Modeling Multiple Regimes in Financial Volatility with a Flexible Coefficient GARCH(1, 1) Model. *Econometric Theory* 25(1), 117–161 (2009)
15. Luper, D., Kazanci, C., Schramski, J., Arabia, H.R.: Flow Decomposition in Complex Systems. *ITNG*, Las Vegas, USA (2011)

Modeling UAS Swarm System Using Conceptual and Dynamic Architectural Modeling Concepts

Hassan Reza and Kirk Ogaard

Department of Computer Science,
University of North Dakota,
Grand Forks, ND, USA
reza@aero.und.edu

Abstract. In this paper, three conceptual architectures for modeling of control architectures for UAS swarms are evaluated according to the criteria of reliability, availability, performance, and adaptability. The digital pheromone conceptual control architecture is proposed as optimal solution satisfying the above criteria. The conceptual architecture for the digital pheromone control architecture could be implemented using dynamic three-tier software architecture. The three-tier architectural style provides a context in which quality attributes such as reliability, availability, performance, and adaptability can be realized. The proposed three-tier software architecture is then specified in the Darwin architecture description language.

Keywords: UAS swarms, software architecture, ADL, Darwin, Dynamic Software Architecture, Conceptual Structure.

1 Introduction

Software system's conceptual architecture is similar to a conceptual structure [14]; they too can be represented by the following concepts: components (i.e., active or computational nodes), connectors (i.e., passive or communicational nodes), glue (i.e., links connecting components and connectors), and configurations (i.e., overall structure) [13]. Conceptual architecture is a knowledge driven process that plays a significant role in the development of a software intensive system because it documents early design decisions using architectural knowledge, facilitates the communications among stakeholder, and promotes reused at architectural level [12]. The design of a successful conceptual architecture for software products should begin by identifying architecturally significant requirements (ASR) of a system. Examples of architecturally significant requirements are security, performance, availability, etc. The software design attempts to construct the whole system by gluing and interconnecting, the components implementing ASR. During the back-end engineering, the developers are attempting to validate and verify the software product according to user requirements.

In this paper, three conceptual control architectures for UAS swarms are evaluated according to the criteria of reliability, availability, performance, and adaptability. The

digital pheromone conceptual control architecture is proposed as optimal solution satisfying the above criteria. The conceptual architecture for the digital pheromone control architecture could be implemented using dynamic three-tier software architecture. The three-tier architectural style provides a context in which quality attributes such as reliability, availability, performance, and adaptability are realized. The proposed three-tier software architecture is then specified in the Darwin architecture description language.

2 Control Architecture for UAS Swarm

Instead of using a single unmanned aerial system (UAS) to accomplish military objectives, a swarm of many small UASs could be used. UAS swarms have many advantages over only operating a single UAS. For example, the large number of identical UASs in a homogeneous swarm makes it highly redundant. Thus, if one of the UASs in the swarm fails, the remaining UASs can still accomplish the mission objectives. UAS swarms are thus highly available.

Two critical requirements for UAS swarms are: 1) the ability to continue operating even if some fraction of the UASs in the swarm fails and 2) the ability to adapt to changes in the operating environment. While many notations exist for modeling mission-critical software at low level of design, such as colored Petri nets [6] and predicate/transition nets [7], few of these conceptual architectural modeling notations known as ADLs [11, 13] are capable of modeling dynamisms required by for UAS swarms. However, the Darwin architecture description language (ADL) [8], which is based on the π -calculus [9] for modeling concurrent processes, is capable of modeling dynamic and distributed software architectures.

The control architectures for UAS swarms, for most part, are motivated by the swarming behaviors demonstrated by insects (e.g. ants or bees). Such control architectures for UAS swarms have many advantages over other types of control architectures [1]. For example, insects in a swarm have a limited degree of autonomy, but they are not fully autonomous. Thus, they are not intelligent agents. UAS swarms modeled after insect swarms do not require sophisticated artificial intelligence for the embedded control software executing on each UAS in the swarm. This eases the task of verification for the embedded control software. The verification of mission-critical software used in military operations is critical. Also, training such UAS swarms to perform useful tasks is easier. Genetic algorithms, for example, might be used to train UAS swarms. And finally, control architectures for UAS swarms modeled after insect swarms have good scalability.

The behavior of UASs in the swarms was controlled through a combination of co-fields [3] and deterministic finite automata (DFAs). A co-field is a reactive control system for UAS swarms where the behavior of each UAS in the swarm is influenced by the behavior of all the other UASs in the swarm. The directional influence exerted on the velocity of some UAS x_i by some other UAS x_j under the co-fields control

architecture is a function of the distance between x_i and x_j . The states in the DFAs used in this control architecture represent the possible types of behaviors or objectives for UASs in the swarm, e.g. “search” or “return to base.” Transitions occur between states in the DFAs when the gradient of the co-field for the respective UAS exceeds some threshold value. Different types of behavior (e.g. patrolling, searching, or protecting) result from using different co-field functions and DFAs.

The particle swarm control architecture described in [4] models the UASs in a swarm in a manner analogous to how gas molecules are modeled in the gas model in physics. Similar to the co-fields control architecture, the UASs in the swarm follow vector fields in the environment. Additionally, each UAS in the swarm is considered to be surrounded by a bubble. Swarms of insects in nature use pheromones to mark important parts of their environment. Digital pheromones, described in [5], were used to replicate this behavior in the control architecture for UAS swarms. The vector field model provides an efficient means for UASs in the swarm to be attracted to targets.

A HOST (Hostility Observation and Sensing Terminals) is a small computer that stores the digital pheromones. It wirelessly broadcasts the strength of its digital pheromones to nearby UASs in the swarm. A UAS in the swarm will only listen to the closest HOST to its current georeferenced position. Each HOST can hold any number of digital pheromone types with varying strengths. The strength of a particular digital pheromone at a HOST can be altered in response to nearby UASs in the swarm depositing the same type of digital pheromones, changes in the HOST’s local environment, or messages received from nearby HOSTs. Digital pheromones at a HOST will also “evaporate” over time. Thus, a digital pheromone will be deleted from the HOST if it is not periodically reinforced by deposits of new digital pheromones with the same type. HOSTs diffuse their digital pheromones to other nearby HOSTs based on the strength of those digital pheromones. Thus, if a HOST holds a strong digital pheromone, it will propagate that digital pheromone to a greater number of nearby HOSTs. Weaker digital pheromones will propagate to fewer numbers of nearby HOSTs.

3 Analysis of Control Architectures

Due to the difficulty in defining co-field functions for complex behaviors, the co-fields control architecture [2, 3] may be limited to only basic swarming behaviors. Furthermore, the use of deterministic finite automata (DFAs) to represent objectives for the UAS swarms makes co-fields somewhat inflexible. A DFA is a static model, since the states and transitions cannot change (i.e., the structural elements of model such as states/arcs can be changed). This feature, in turn, results in unacceptable adaptability for the co-fields control architecture.

While a control architecture based on digital pheromones [5] meets all the critical requirements for UAS swarms, the requirement for digital pheromone holders (i.e. HOSTs) to be physically deposited in the operating environment is too restrictive for dynamic scenarios (e.g. the rapid deployment of a UAS swarm to a novel environment).

Table 1 compares the three control architectures using the criteria of reliability, availability, performance, and adaptability. The co-fields and particle swarm control architectures provide little or no support for adapting to changes in the operating environment. The co-fields control architecture is likely to have poor performance due to its communication model, which requires new information to be propagated through the UAS swarm via wireless broadcasts. The particle swarm control architecture is probably less reliable than the other two, because its behavior is nondeterministic. Hence, its ability to reliably accomplish objectives is also unpredictable.

Table 1. A comparison of three control architectures for UAS swarms

	Co-Fields	Particle Swarms	Digital Pheromones
Reliability	+	–	+
Availability	+	+	+
Performance	–	+	+
Adaptability	n/a	–	+

+: Good support for the quality attribute; –: Poor support for the quality attribute; n/a: Not applicable.

A control architecture for UAS swarms based on digital pheromones meets all the necessary criteria—reliability, availability, performance, and adaptability. Although no software architecture for digital pheromones was explicitly specified in [5], one possibility is a three-tier software architecture (see figure 1) consisting of tiers of components for controlling: 1) the UASs in the swarm, 2) the digital pheromones stored on the digital pheromone holders, and 3) the georeferenced positions of the digital pheromone holders.

The following assumptions were made with the three-tier software architecture: 1) the initial number of UASs in the swarm is known, 2) the number of UASs in the swarm may decrease (due to failure) or remain the same but cannot increase, 3) the number of digital pheromone holders is known, 4) the number of digital pheromone holders remains constant throughout the operation of the UAS swarm, 5) the digital pheromone holders are deposited in the operating environment prior to the deployment of the UAS swarm, and 6) the UASs in the swarm are programmed with their mission objective prior to deployment (i.e. the objective cannot be changed dynamically).

The deployment architecture shows the mapping between processes represented by rounded rectangular to the processors represented by simple rectangular. The architecture consists of each UAS in the swarm executing a *UASController* component and each digital pheromone holder executing corresponding *PheromoneSet* and *PheromoneHoldersPositions* components. The pseudo-code for the three components can be briefly described as follows:

- The *UASController* component periodically broadcasts the UAS's georeferenced position (*positionForUAS*) to nearby digital pheromone holders using the UAS's wireless telemetry protocol.

- The *PheromoneSet* components (executing on the nearby digital pheromone holders) receive the georeferenced position broadcast by the UAS. Each *PheromoneSet* component sends its unique identifier (*pheromoneHolderID*) and the georeferenced position of the UAS (*positionForUAS*) to its corresponding *PheromoneHolderPositions* component.
- Its corresponding *PheromoneHolderPositions* component replies with a Boolean value (*isClosestHolderFlag*) indicating whether that digital pheromone holder is currently the closest digital pheromone holder to the UAS's georeferenced position.

Thus, each nearby digital pheromone holder independently determines whether it should respond to the UAS's request based on its proximity to the UAS. The response from the digital pheromone holder (*velocityForUAS*) is a wireless transmission to the UAS containing the new velocity vector for the UAS to follow.

Although the georeferenced positions for UASs in the swarm will vary, the georeferenced positions for the digital pheromone holders should remain constant. By calculating the distances between the current georeferenced position of the UAS and the georeferenced positions of all the digital pheromone holders, the *PheromoneHolderPositions* components determine whether a particular digital pheromone holder is the closest to a particular UAS.

The most important part of the velocity vectors provided to the UASs by the digital pheromone holders is the direction. This indicates the direction the UAS must fly to reach the next point in the digital pheromone trail. However, the magnitude could be used to represent the airspeed the UAS is required to maintain. If the magnitude is used in this fashion, then stronger digital pheromone trails will produce faster velocity vectors. Thus, once one or more UASs discover the target, the remaining UASs in the swarm will rapidly converge on that target.

4 The Formal Description of Conceptual Software Architecture

The three-tier software architecture can be shown to support the four critical quality attributes of reliability, availability, performance, and adaptability. In the three-tier software architecture, all the layers are redundant—the *UASController* components connectors (i.e., links between elements of a systems), and configuration (i.e., overall conceptual structure of a system).

Darwin ADL [8] is an ADL that can be used to model distributed and/or dynamic software systems. Although the Darwin ADL supports dynamic instantiation, it does not support other important dynamisms, such as dynamic reconfiguration of connectors. A more precise description for a software architecture based on digital pheromones would have reconfigurable connectors between the *UASController* components and the *PheromoneSet* components. Since all communication between the UASs in the swarm and the digital pheromone holders is wireless, in a strict sense the *UASController* components are only directly connected to proximate *PheromoneSet* components. As the georeferenced position of a UAS changes relative to the georeferenced positions of the digital pheromone holders, the connectors also change

due to the limited range of wireless transmissions from the UASs in the swarm. However, since the Darwin ADL doesn't support dynamic reconfiguration of connectors, this architectural aspect was represented in the three-tier software architecture by connectors between every *UASController* component and every *PheromoneSet* component. Obviously, this may not always literally be the case.

Dynamic instantiation is implicitly used in the three-tier software architecture. The initial number of UASs in the swarm and the number of digital pheromone holders are specified prior to instantiation. After the initial creation of these components, the failures (i.e. deletions) of *UASController* components are handled seamlessly by the architecture. The failure of *PheromoneSet* components was not considered, because this type of failure requires dynamic reconfiguration of the connectors in the architecture. Also, if failure of *PheromoneSet* is considered, there must be a mechanism for detecting such types of failures. This is necessary so the reconfiguration of the connectors can be performed at the appropriate times. The databases stored in the *PheromoneHolderPositions* components would also need to be updated after failure of a *PheromoneSet* component. The failure of a *PheromoneHolderPositions* component would simply result in the failure of its corresponding *PheromoneSet* component.

Such dynamic reconfiguration of connectors would likely involve a special component exclusively dedicated to dynamically changing the configuration, such as the configurator component proposed for the Wright ADL in [10]. Few ADLs offer complete support for representing dynamic architectures such as the three-tier software architecture for UAS swarms [11]. C2 SADL provides complete support for dynamisms [11], but only if the C2 architectural style is used. However, the C2 architectural style is not well suited for implementing control software for UAS swarms. Figure 1 depicts the software architecture of a proposed USA swarm system.

The three-tier software architecture can be described in the Darwin ADL as follows:

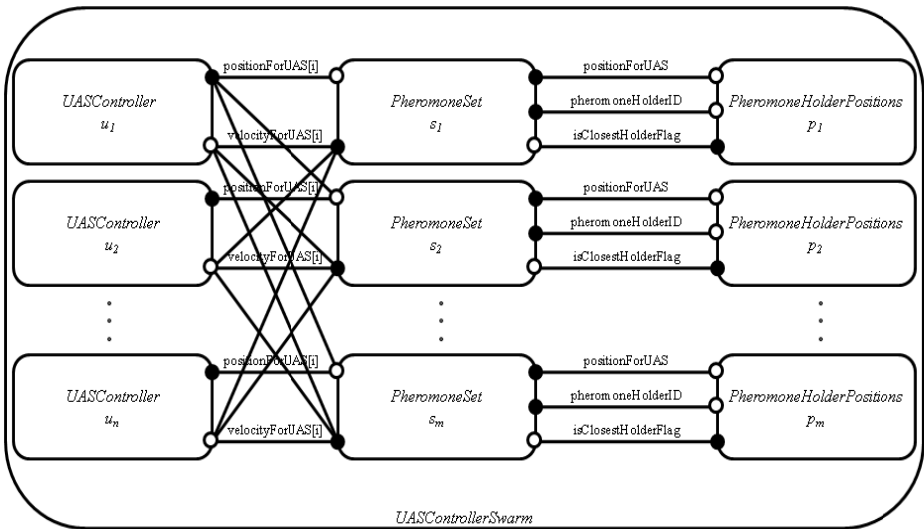


Fig. 1. The proposed three-tier software architecture of UAS swarm systems

```

component PheromoneSet(int n)
{
  provide velocityForUAS[n], positionForUAS, pheromoneHolderID;
  require positionForUAS[n], isClosestHolderFlag;
}
component PheromoneHolderPositions
{
  provide isClosestHolderFlag;
  require positionForUAS, pheromoneHolderID;
}
component UASController(int m)
{
  provide positionForUAS[m];
  require velocityForUAS[m];
}
component UASControllerSwarm(int n, int m)
{
  array U[n]: UASController;
  array S[m]: PheromoneSet;
  array P[m]: PheromoneHolderPositions;
  forall i:0..n-1
  {
    inst U[i] @ i + 1
  }
  forall i:0..m-1
  {
    inst S[i] @ n + i + 1;
    inst P[i] @ n + i + 1;
    bind
      P[i].isClosestHolderFlag -- S[i].isClosestHolderFlag;
      S[i].positionForUAS -- P[i].positionForUAS;
      S[i].pheromoneHolderID -- P[i].pheromoneHolderID;
  }
  forall i:0..n-1
  {
    forall j:0..m-1
    {
      bind
        U[i].positionForUAS[j] -- S[j].positionForUAS;
        S[j].velocityForUAS[j] -- U[i].velocityForUAS;
    }
  }
}

```

Fig. 2. Specification of controlling UAS Swarms Software Architecture in Darwin

5 Conclusion and Future Work

Using UAS swarms in mission-critical environments would have many advantages over only using a single UAS, including improved reliability, availability, performance, and adaptability. When three possible control architectures for UAS swarms were evaluated according to these criteria, the control architecture based on digital pheromones was the only one that satisfied all of four critical requirements. The software for such control architecture could be efficiently implemented using dynamic three-tier software architecture.

Future work could include an architectural representation for the dynamic reconfiguration of connectors between *UASController* and *PheromoneSet* components, as well as accounting for the possibility of *PheromoneSet* and *PheromoneHolderPositions* components failing during UAS swarm operations. Future work could also consider cases where the objective for the UAS swarm can be changed dynamically for optimal performance, reliability, adoptability and/or availability. Validation of the three-tier software architecture using tools for the Darwin ADL is also important. Therefore, extensive simulations and experiments are needed to demonstrate the feasibility of this proposed solution.

References

- [1] Parunak, H.: Making swarming happen. In: Proceedings of the Conference on Swarming and Network Enabled Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (January 2003)
- [2] Chalmers, R., Scheidt, D., Neighoff, T., Witwicki, S., Bamberger, R.: Cooperating unmanned vehicles. In: Proceedings of the AIAA First Intelligent Systems Technical Conference (September 2004)
- [3] Mamei, M., Zambonelli, F., Leonardi, L.: Co-fields: A physically inspired approach to motion coordination. *IEEE Pervasive Computing* 3(1), 52–61 (2004)
- [4] Feddema, J., Schoenwald, D., Parker, E., Wagner, J.: Analysis and control for distributed cooperative systems (September 2004), <http://est.sandia.gov/consequence/docs/DistributedCoopSystems.pdf> (last accessed October 2010)
- [5] Parunak, H., Purcell, M., O’Connell, R.: Digital pheromones for autonomous coordination of swarming UASs. In: Proceedings of the First AIAA Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference (May 2002)
- [6] Fukuzawa, K., Saeki, M.: Evaluating software architectures by coloured Petri nets. In: Proceedings of the Fourteenth International Conference on Software Engineering and Knowledge Engineering, pp. 263–270 (July 2002)
- [7] Xu, D., Volz, R., Ioerger, T., Yen, J.: Modeling and verifying multi-agent behaviors using predicate/transition nets. In: Proceedings of the Fourteenth International Conference on Software Engineering and Knowledge Engineering, pp. 193–200 (July 2002)
- [8] Magee, J., Dulay, N., Eisenbach, S., Kramer, J.: Specifying distributed software architectures. In: Proceedings of the Fifth European Software Engineering Conference, pp. 137 – 153 (September 1995)
- [9] Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, parts I and II. *Journal of Information and Computation* 100, 1–40, 41 – 77 (1992)
- [10] Allen, R., Douence, R., Garlan, D.: Specifying dynamism in software architectures. In: Proceedings of the Foundations of Component-Based Systems Workshop (September 1997)
- [11] Medvidovic, N.: A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering*, 70–93 (August 2002)
- [12] Ali babar, M., Dingsoyr, T., Lago, P., Vliet, H.: *Software Architecture Knowledge Management: Theory and Practice*. Springer, Heidelberg (2009)
- [13] Taylor, R., Medvidovic, N., Dashpfi, E.: *Software Architecture: Foundations, Theory, and Practice*. Wiley, Chichester (2010)
- [14] Sowa, J.: *Conceptual Structures: Information Processing in Mind and Machine*. Addison Wesley, Reading (1984)

Name Extraction and Formal Concept Analysis^{*}

Kazem Taghva, Russell Beckley, and Jeffrey Coombs

School of Computer Science,
University of Nevada, Las Vegas
kazem.taghva@unlv.edu

Abstract. Many applications of Formal Concept Analysis (FCA) start with a set of structured data such as objects and their properties. In practice, most of the data which is readily available are in the form of unstructured or semi-structured text. A typical application of FCA assumes the extraction of objects and their properties by some other methods or techniques. For example, in the 2003 Los Alamos National Lab (LANL) project on Advanced Knowledge Integration In Assessing Terrorist Threats, a data extraction tool was used to mine the text for the structured data. In this paper, we provide a detailed description of our approach to extraction of personal names for possible subsequent use in FCA. Our basic approach is to integrate statistics on names and other words into an adaptation of a Hidden Markov Model (HMM). We use lists of names and their relative frequencies compiled from U.S. Census data. We also use a list of non-name words along with their frequencies in a training set from our collection of documents. These lists are compiled into one master list to be used as a part of the design.

1 Introduction

Formal Concept Analysis (FCA) is a mathematical theory based on the concept of Galois Lattice [1]. The methodologies based on FCA facilitate the identification of concepts (or grouping of objects) based on their attributes. It is widely used in many applications such as information extraction, data mining, and software engineering [6]. In FCA methodology, it is assumed that objects and their attributes are already identified by other means. One of the objects playing important role in some applications of FCA is the object of personal name. In some of our applications we were interested in identification of personal names for subsequent processing including FCA and other relational extractions. We therefore developed a name finding approach that was robust in presence of noisy text that were produced as a result of OCR (Optical Character Recognition) conversion.

Our name finder was designed to find personal names in unstructured text having moderate OCR noise. Our initial motive for the name finder was to enhance our privacy act classifier (PAC) [7][8]. Finding personal names can help to identify private information in two ways [4][3]. First, some types of personal information are not legally protected unless the information is associated with a personal name. Second,

^{*} International Workshop on the Concept Formation and Extraction in Under-Traversed Domains (CFEUTD-2011).

even in some cases where personal names are not necessary, they can be a strong indicator of nearby relevant personal information.

Finding names in unstructured noisy text is difficult for several reasons[9][5]. Most obviously, noise might mutilate the text of the personal name or useful context. Also, many first and last names in our collection are rare, and come from a variety of ethnicities. Furthermore, many names are composed of common words e.g. 'Mark Hill' may refer to a person. Conversely, non-personal-name references may juxtapose to resemble personal names e.g. 'will cook' may be a description of a housekeeper.

This paper consists of five other sections in addition to this introduction. Section 2 is a short description of our approach. Section 3 describes our information lists that are used in our HMM. Sections 4 and 5 are description of our main techniques. Finally section 6 is our conclusion and future work.

2 Overview of Our Approach

Our basic approach is to integrate statistics on names and other words into an adaptation of a hidden markov model. We use lists of names and their relative frequencies compiled from U.S. Census data[2]. We also use a list of non-name words along with their frequencies in a training set from our collection of documents from a federal agency. These lists are compiled into one master list to be used at runtime.

Our adaptation of a hidden markov model includes an extra factor to account for the punctuation and whitespace between tokens. If this falls outside the definition of a hidden markov model, even if we attach the preceding inter-word symbols to each token, because the probability of the transition emission is a function of multiple states. We use a slightly modified Viterbi algorithm to find the most probable path, and select tokens that are emitted by name states in the most probable path.

3 The Information Lists

We used six word lists to compile a single master list to be used at runtime.

The first list is the non-name go-list, which we used to help build the second list, the non-personal-name list. The non-name go-list is a list of words that, in our collection, are often capitalized but rarely refer to personal names. It consists mostly of names of governmental and commercial organizations, and geographical names e.g. "River". To make this list, to identify possible names, we use the regular expression $\wedge\text{b}[A-Z][a-z]^+$, which we also later used to reject tokens from our non-personal-name list tally. We tracked the frequency of each string matching $\wedge\text{b}[A-Z][a-z]^+$ and put the 7000 most common strings into an initial list, which we then manually purged of words that are more common as personal-names than as non-personal-names. Furthermore, as we developed the name-finder we added many other words that had been underrepresented in previous versions of the non-personal-name list.

The next step was to form the non-personal-name list. We went through the training set and tallied every word occurrence that either did not match $\wedge\text{b}[A-Z][a-z]^+$ or did match an entry in the non-name go-list. For each word that made the list, we record edit frequency: the number of occurrences divided by the total number of

non-personal-name word occurrences in the text. The list was sorted in descending order of frequency, and, based on this order, each entry had a cumulative frequency.

The next three lists—female first names, male first names, and last names—were taken from the 1990 U.S. census. Like the non-personal-name word list, these included the frequency and cumulative frequency for each entry.

Finally, the list of titles—such as 'Ms.' and 'Senator'—was manually compiled and had the same kind of information as the name lists and the non-personal name word list.

4 Recipe for the Master List

Consolidating the word lists into one master list served three objectives. First, it facilitated choosing the n most indicative words of any type, rather than some given number of words for each type. Secondly, it put all of the frequency information for a single word (which could occur in multiple lists) into one spot, so that at run time, multiple searches for the same word were unnecessary. Thirdly, it facilitated the calculation of emission probability components in advance.

5 Criteria for Selecting Words from Word Lists for Master List

The formula for selecting list words is based on two criteria. First we are concerned with how well the word signifies a word class e.g. 'Johnson' strongly suggests a name, while 'Hill' is somewhat more ambiguous. Secondly, we prefer words that are more common so that the list will cover a large percentage of the lexicon. The word 'the' is strong by both measures. The word 'Polychloroflourinate' is strong by the first measure but weak by the second. The word 'will' is weak by the first measure but strong by the second. The word 'ther't' is poor by both measures.

The formula for the worthiness of each word, for any list of type t in which it is found, is as follows:

Define nf the name-frequency:

$$nf = \frac{\text{occurrences as type } t \text{ in a personal name}}{\text{total number of word occurrences in text.}}$$

Define wf , the word-frequency:

$$wf = \frac{\text{occurrences other than in a personal name(any type)}}{\text{total number of word occurrences in text.}}$$

Define tf , the total frequency:

$$tf = nf + wf$$

Let p_n be the probability that a random token in the text is used as type t as part of a personal name, then

$$\begin{aligned} \text{name value} &= 2 * n_f * p_n - t_f \\ \text{word value} &= 2 * w_f * p_n - t_f \\ \text{value} &= \max(\text{name value}, \text{word value}) \end{aligned}$$

The master list contains the ten-thousand words with the highest value.

For each word in the master list, we have five values corresponding to the five word types. For case-folded word w and word type t ,

$$\text{masterlist number}(w,t) = P(w \mid \text{state of type } t)$$

is a factor in the emission probability for occurrences of w from states of type t . $P(w \mid \text{state of type } t)$ depends on its inclusion in, frequency in, or exclusion from the word lists of every type.

Calculating $\text{masterlist number}(w,t)$ proceeds as follows:

First, for types t_1, t_2 , and word w , define:

- $\text{coverage}(t_1)$ = the proportion of occurrences of t_1 referents that are found in the t_1 list e.g. the number of people in the U.S. whose last names occur on our name list.
- $\text{intersection}(t_1, t_2) = \text{freq}(w, t_1)$

$$\diamond_{\{w \mid w \in t_1 \text{ AND } w \in t_2\}}$$

- $\text{overlap}(t_1, t_2) = P(\text{word } w \text{ is in list } t_2 \mid \text{a word occurrence with a referent of type } t_1)$.
- $\text{co-verlap}(t_1, t_2) = P(\text{word } w \text{ is a given entry in list } t_2 \mid \text{a word occurrence with a referent of type } t_1)$

Calculate thusly:

- $\text{intersection}(t_1, t_2) = 0.0$
- For each word in type t_2 list, w :
 - if w is in t_1
 - * $\text{intersection}(t_1, t_2) = \text{intersection}(t_1, t_2) + \text{freq}(t_1, w)$
- $\text{overlap}(t_1, t_2) = \frac{\text{intersection}(t_1, t_2)}{\text{coverage}(t_1, t_2)}$
- $\text{co-verlap}(t_1, t_2) = \frac{\text{intersection}(t_1, t_2)}{\text{coverage}(t_1, t_2) | t_2 |}$

Now, calculate $\text{master list value}(w, t_1)$ for word w and each type.

```

for each type t1.
  sum = 0.0
  for each type t2
    if t1 != t2
      ifwis NOTin t2
        sum = sum + log(1 -overlap(t1,t2))
      else
        sum = sum + log(co-verlap(t1,t2))
    else
      ifwis NOTin t2
        sum = sum + log(1 -coverage(t1,t2))
      else
        sum = sum + log(frequency(w, t1))
master list value(w,t1)= sum

```

Finally, for each type, t , we need a reject probability: $\text{reject}(t) = p(w \text{ is not in master list } | t)$. To get this we look at the sum of the frequencies of words that are in list but not in the master list, and add the complement of $\text{coverage}(t)$ as defined above. You can also think of this as $1.0 - \sum w \text{ in } t \text{ AND in master list frequency}(w,t)$. Compute as follows:

```

sum = 0
for each word w in list t
  if w is not in master list
    sum = sum + frequency(w,t)
reject(t) = sum + 1.0 -coverage(t)

```

OTHER FACTORS

Whitespace and Punctuation

The hmm looks at the symbols between words to modify the probabilities associated with any path. We define a partition I , on inter-word string classes e.g. the literal string $'.'$ forms a class, as does the regular expression $/: s^*/$. For every arc in the hmm and every i in I , we associate a probability that factors into every path once for each time the path travels the arc and the associated inter-word string is a match for i . This is complicated by the fact that when initials are used we expect a different distribution of the inter-word classes. Most importantly, we expect the string $'.'$ to be much more common following an initial than following a complete name word. We keep a global variable that is true if and only if the previous token consisted of exactly one letter.

Capitilization pattern

We take it for granted that personal names are more likely to be capitalized than randomly chosen words that are not personal names. The ideal capitalization pattern is $Xxxxx...$, but there are normal exceptions such as MacDonalld and DeLouise etc. Also, we assume that names without vowels are highly improbable. All these considera-

tions are integrated to form a partition on all words such that each class has an associated probability for personal-name types and an associated probability for non-personal-name words.

EXECUTION

First, we prime for the viterbi algorithm by setting the initial values for each state, based on all factors except the emission transition. Then we continue with viterbi combining all factors to assess the probability for each state X phase. We update the maximal path scores for each state in each phase i , where the word-string is s and the preceding inter-word string is g , thusly:

definelist value(w,t)= master list value(w,t) if w is in the masterlist
 log(reject(t))
 otherwise

define candidate score($s1,s2,i$) = list value($w,type(s1)$)+log(capitalization(w) +
 log(inter word emission score($s1, s2,g$)+ max path score($s1,i -1$)) for each state $s1$
 max path score($s1,i$) = max subs2 in S [candidate score($s1,s2, i$)]

The Viterbi algorithm returns a sequence of states comprising the maximal path, in which every occurrence of a non-background state we take to signify part of a name in the text. That is, with a couple of exceptions. If the name is Washington D.C. or P.O.Box, we reject it without further ado.

6 Conclusion and Future Work

This paper is a preliminary report on our work for identification of personal names. It is anticipated to extend this work on other entities for FCA. It is also anticipated to further report on the use of FCA in our future work on privacy act classification.

References

- [1] Ganter, B., Wille: Formal concept analysis. Springer, Heidelberg (1999)
- [2] U.S. Government. Frequently occurring first names and surnames from the 1990 census, <http://www.census.gov/genealogy/www/freqnames.html> (viewed August 2005)
- [3] U.S. Government. The freedom of information act 5 U.S.C. sec. 552 as amended in 2002, <http://www.usdoj.gov/oip/foiaupdates/VolXVII4/page2.htm> (viewed June 30, 2004)
- [4] U.S. Government. The privacy act of 1974 5 u.s.c. sec. 552a, <http://www.usdoj.gov/04foia/privstat.htm> (viewed August 22, 2005)
- [5] Miller, D., Boisen, S., Schwartz, R., Stone, R., Weischedel, R.: Named entity extraction from noisy input: Speech and OCR. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, pp. 316–324 (2000)
- [6] Rocha, L.M.: Proximity and semi-metric analysis of social networks. Report of Advanced Knowledge Integratio In Assessing Terrorist Threats LDRD-DR Network Analysis Component. LAUR 02-6557

- [7] Taghva, K., Beckley, R., Coombs, J.: The effects of OCR error on the extraction of private information. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 348–357. Springer, Heidelberg (2006)
- [8] Taghva, K., Beckley, R., Coombs, J., Borsack, J., Pereda, R., Nartker, T.: Automatic redaction of private information using relational information extraction. In: Proc. IS&T/SPIE 2006 Intl. Symp. on Electronic Imaging Science and Technology (2006)
- [9] Taghva, K., Borsack, J., Nartker, T.: A process flow for realizing high accuracy for ocr text. In: SDIUT 2006 (2006)

Towards the Development of an Integrated Framework for Enhancing Enterprise Search Using Latent Semantic Indexing

Obada Alhabashneh, Rahat Iqbal, Nazaraf Shah, Saad Amin, and Anne James

Faculty of Engineering and Computing,
Coventry University, UK

{alhabaso, r.iqbal, n.shah, s.amin, a.james}@coventry.ac.uk

Abstract. While we have seen significant success in web search, enterprise search has not yet been widely investigated and as a result the benefits that can otherwise be brought to the enterprise are not fully realized. In this paper, we present an integrated framework for enhancing enterprise search. This framework is based on open source technologies which include Apache Hadoop, Tika, Solr and Lucene. Importantly, the framework also benefits from a Latent Semantic Indexing (LSI) algorithm to improve the quality of search results. LSI is a mathematical model used to discover the semantic relationship patterns in a documents collection. We envisage that the proposed framework will benefit various enterprises, improving their productivity by meeting information needs effectively.

Keywords: Enterprise Search (ES), Latent Semantic Indexing (LSI), Search Context, Document Ranking.

1 Introduction

Enterprises have a rich and diverse collection of information resources. Such resources can be divided into two categories: structured, which is encoded in the databases; and unstructured, which is encoded in documents.

The retrieval of structured information has been well investigated [1], and several search tools or products are available in the market, such as the traditional database engines (e.g. Oracle, Microsoft SQL server, MySQL). However, retrieval of unstructured information is still a challenging task due to several problems associated with the search of unstructured information. For example, lack of anchor text (e.g., hyperlinks) and the heterogeneous formats of documents.

Enterprise Search Engines (ESEs) are still not sufficiently mature to provide high quality results that fully meet its users' needs. According to the International Data Corporation (IDC) report, there are significant economic losses caused by poor quality of enterprise search. The report also noted that there is dissatisfaction by the enterprise executives about the performance and information quality of the available ESEs [2, 3].

Although a wide range of commercial enterprise search products are available from various vendors, such as, Google, Verity, IBM, Oracle, Microsoft and Panoptic, none of the existing enterprise search products provide an effective solution [2, 4, 5].

Enterprise search has attracted relatively little interest in the research community, particularly in the area of unstructured information retrieval [4, 6]. A number of researchers have attempted to address enterprise search problems with a varying degree of success [6, 7, 1, 8, 9, 10, 11, 25]. For example, Dmitriev et al used the implicit and explicit annotation to substitute the lack of anchor texts in order to enhance the ranking of documents in search results [6]. Mangold et al proposed a framework to extract the search context and the logical structure of the information from enterprise databases to enhance the quality of search results [1]. Zhu et al attempted to disambiguate users queries using dictionaries to enrich those queries with additional keywords [7].

In this paper, we present an integrated framework for enhancing the enterprise search. The framework is based on open source technologies which include Apache Solr, Lucene, Tika, and Hadoop. The selection of open source technologies is made in order to address the issues concerning scalability, and enabling incorporation of LSI algorithm. The objective of using LSI is to enhance search results obtained using available open source enterprise search technologies.

The rest of the paper is organized as follows. Section 2 discusses some of the problems related to enterprise search. Section 3 presents the proposed framework. Section 4 concludes the paper and outlines future research direction.

2 Enterprise Search Problems

2.1 Heterogeneous Documents

Most of the enterprise documents are non-web documents. They are of heterogeneous nature having different types and structure. This type of document heterogeneity causes the application of techniques normally used for ranking webpages to produce less efficient results. For example Power Point files consist of slides and each slide has a title and body; the title part, logically, should have a higher importance than the body part. On the other hand, the Excel sheets have a structure of columns and rows and always consist of numerical values with a limited text description apart from the columns' titles. The challenge is how to apply the same ranking algorithm or technique on a different file types. Text based ranking methods are not effective in this case. [1, 2, 3, 4]

2.2 Non-web Document

Structure analysis shows that the enterprise web is not following the same bow-tie structure as WWW pages, which makes the page rank algorithm less efficient in enterprise search [1, 2, 3, 4, 12], as enterprise documents have no anchor texts. Anchor texts are used by the traditional web search engines as a base to calculate the document importance in their document ranking algorithms.

2.3 Search Context

Search context is useful for disambiguating short or ambiguous queries, since it adds more keywords to the user query which in turn makes it clearer to the search process and can produce a properly ranked search result list [1, 2, 3, 4, 12].

3 The Proposed Framework

We propose an integrated framework in order to improve the quality of enterprise search. By quality we mean the efficiency of the enterprise search engine, accuracy and relevance of results which will satisfy the user. The proposed framework consists of four major components such as Hadoop, Tika, Solr and LSI as shown in Figure 1. These components are briefly described in the subsequent subsections and outlines in table 1.

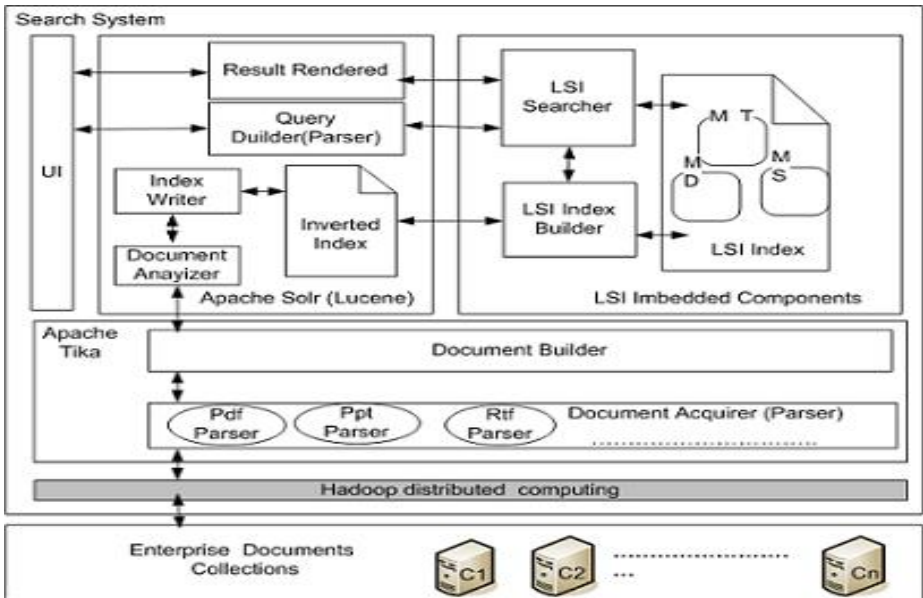


Fig. 1. The proposed Framework

Table 1. Description of the proposed framework components

No	Component	Description
1	Apache Hadoop	An open source framework for distributed computing
2	Apache Tika	An open source toolkit that can parse and acquire different types of documents
3	Apache Solr	An open source enterprise search server
4	Latent Semantic Indexing (LSI)	A vector space based model that is used to retrieve documents based on semantic relationships.

3.1 Apache Hadoop

The Apache Hadoop is an open source component developed for the distributed computing paradigm. It includes several components to provide infrastructure for reliable and scalable distributed computing. Hadoop implements a computational paradigm named map/reduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster [26, 28].

3.2 Apache Tika

Apache Solr and its underlying technology can parse or acquire limited types of documents. In order to address this issue, we use a specialised toolkit, called Apache Tika, to deal with a diversity of document types encountered in enterprise document collections.

Apache Tika is an open source document acquiring and building toolkit that is compatible with Solr. It is able to acquire different types of documents using its standard library. It also offers new plugin services to acquire additional document types [25, 27, 30]. It has two sub components as briefly described below:

Acquiring Component. This component is responsible for extracting metadata from documents as well as the body in the form of text, and then passing this to the document building component.

Document Building Component. This transforms documents from stream text to a field form such as; title, author, and date of creation, the body, the link and others. Solr's analyzing component takes these forms as input and performs analysis on the documents to prepare them for indexing.

3.3 Apache Solr

Apache Solr is a java based open source enterprise search server. Solr can communicate with other components and applications using standard languages (example: XML) and protocols (example: HTTP).

Apache Solr is widely used in public websites such as CNet, Zappos, and Netflix, as well as intranet sites. In addition to its ability to return a list of search results, it has various other features such as: result highlighting, faceted navigation which allows user to select various alternatives for their query, search term spelling suggestions, auto-suggest queries or more option for finding similar documents.

Apache Lucene is the core technology underlying Solr. Lucene is an open source, high-performance text search library [26, 29] whereas the Solr is a search server. Solr uses the following components:

Document Analysis Component. No search engine indexes text directly. Instead, the text is broken into a series of individual atomic elements called tokens. This is carried out in the 'analyse document' stage, and this step determines how the textual fields in the document are divided into a series of tokens preparing to add it to the index.

Index Writer. After the input has been converted into tokens it is added to the index. Index writer stores the input in a data structure known as an inverted index. The index is built incrementally as building blocks called segments, and each segment consists of a number of documents. The Solr (Lucene) index consists of the following main files: Segments, Fields Information, Text Information, Frequency, and Position.

Query Parser. The query may have Boolean operations, phrase queries (double quoted), wildcard terms and other expressions that might need specific processing to transform it into the appropriate syntax required by the search mechanism. At this stage, query parser transforms the user's query into common search syntax, known as query object form. With respect to that, Solr provides a package to build the query called QueryParser.

Result Renderer. This part is responsible for taking the search results as a plain list, and then putting them in the right order before passing them to the user interface. Result Renderer will be tailored to fit the LSI based search result.

3.4 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a vector space based technique used to create associations between conceptually related documents' terms [16, 17, 18]. LSI is a novel information retrieval method used to retrieve relevant documents using a statistical algorithm. The algorithm is capable of finding relevant documents based on the semantic relationship between the terms used in the user query and a collection of documents [13, 19, 20]. LSI has been proven to be more effective in searching and ranking relevant items as compared to other classical key word based methods, such as Google and Yahoo [13, 21, 22, 23, 24].

The LSI uses linear algebra to identify the conceptual correlations among a text collection. In general, the process involves the following steps;

- Building term-document matrix. This step builds the term-document matrix (A) which consists of the terms (rows) and documents (columns).
- Weighting the matrix (A) which applies a mathematical semantic weighting formula to give a numerical value for each cell in the matrix.
- Performing an SVD technique on the matrix (S) to identify patterns in the relationships between the terms and concepts contained in the text [14]. It then builds the term-document vector spaces by transforming the single term-frequency matrix, A , into three other matrices: (T) the term-concept Vector matrix, (S) the singular values matrix, and (D) the concept-document vector matrix as shown below [14]:

$$A \approx T . S . D$$

After the term document vector space is built, the similarity of terms or documents can be represented as a factor of how close they are to each other, which could be simply computed as a function of the angle between the corresponding vectors.

- Querying and Augmenting LSI matrices. The same steps are used to locate the vectors representing the text of queries and new documents within the document space of an existing LSI index [15].

LSI Index Builder. The LSI index builder builds the LSI index. It takes the Solr inverted index as input and then creates the basic Term-Document matrix by applying the LSI semantic weighting function. Following that it applies Singular Value Decomposition to the matrix to extract the three sub matrices (T, D and S) which construct the LSI index.

LSI Searching and Ranking. We modify the Apache Solr searching and ranking component to be used with the LSI index in order to take into account the semantic weights in the ranking of relevant documents.

4 Conclusions and Future Work

In this paper, we presented an integrated framework for enhancing enterprise search. The framework is based on open source technologies which include Apache Hadoop, Tika, Solr and Lucene. Importantly, this framework also benefits from Latent Semantic Indexing (LSI) algorithm to improve the quality of the obtained search results. LSI is a mathematical model used to discover the semantic relationships in a collection of documents. We envisage that the proposed framework will benefit various enterprises, improving their productivity by meeting their information needs more effectively. Our future work will include further development of the proposed framework. We will conduct a series of experiments on test data (TREC 2007 Enterprise Document TEST Collection) to evaluate the effectiveness of this framework. Different evaluation metrics will be used to measure the accuracy of the obtained results, efficiency of the proposed implementation, and user satisfaction.

References

1. Mangold, C., Schwarz, H., Mitschang, B.: u38: A Framework for Database-Supported Enterprise Document-Retrieval. In: 10th International Database Engineering and Applications Symposium (IDEAS 2006), IEEE, Los Alamitos (2006) 0-7695-2577-6/06
2. Hawking, D.: Challenges in Enterprise Search. In: 5th Australasian Database Conference (ADC 2004), Dunedin, NZ, Conferences in Research and Practice in Information Technology, vol. 27 (2004)
3. Feldman, S.: Sherman. C.:The cost of not finding Information. IDC (2003)
4. Dmitriev, P., Serdyukov, P., Chernov, S.: Enterprise and desktop search. In: WWW 2010, pp. 1345–1346 (2010)
5. Owens, L.: The Forrester Wave™: Enterprise Search, Q2 (2008)
6. Dmitriev, P., Eiron, N., Fontoura, M., Shekita, E.: Using Annotations in Enterprise Search. In: WWW 2006. ACM, Edinburgh (2006)
7. Zhu, H., Raghavan, S., Vaithyanathan, S., Löser, N.A.: The intranet with high precision. In: 16th international conference on World Wide Web, pp. 491–500 (2007)

8. Li, H., Cao, Y., Xu, J., Hu, Y., Li, S., Meyerzon, D.: A new approach to intranet search based on information extraction. In: 14th ACM International Conference on Information and Knowledge Management, pp. 460–468 (2005)
9. Xue, G., Zeng, H., Chen, Z., Zhang, H., Lu, C.: Implicit link analysis for small web search. In: 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 56–63 (2003)
10. Fisher, M., Sheth, A.: Semantic Enterprise Content Management. Practical Handbook of Internet Computing (2004)
11. Demartini, G.: Leveraging Semantic echnologies for Enterprise Search. In: PIKM 2007. ACM, Lisboa (2007) 978-1-59593-832-9/07/001
12. Mukherjee, R., Mao, J.: Enterprise search: tough stuff. Queue 2 (2004)
13. Telcordia Technologies, <http://lsi.research.telcordia.com>
14. Berry, W., Dumais, T., Brien, W.: Using Linear Algebra for Intelligent Information Retrieval. SIAM Review 37(4), 573–595 (1994/1995)
15. Brand, M.: Fast Low-Rank Modifications of the Thin Singular Value Decomposition. Linear Algebra and Its Applications 415, 20–30 (2006)
16. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: Indexing by latent semantic analysis. J.of the Society for Information Science 41(6) (1990)
17. Chen, C., Stoffel, N., Post, M., Basu, C., Bassu, D., Behrens, C.: Telcordia LSI Engine: Implementation and Scalability Issues. In: 11th Int. Workshop on Research Issues in Data Engineering (RIDE 2001): Document Management for Data Intensive Business and Scientific Applications, Heidelberg (2001)
18. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: Indexing by latent semantic analysis. J. of the Society for Information Science 41(6) (1990)
19. Landauer, T.: Learning Human-like Knowledge by Singular Value Decomposition: A Progress Report, pp. 45–51. MIT Press, Cambridge (1998)
20. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive Learning Algorithms and Representations For Text Categorization. In: ACM-CIKM 1998, Maryland (1998)
21. Zukas, A., Price, R.J.: Document Categorization Using Latent Semantic Indexing. White Paper, Content Analyst Company, LLC (2003)
22. Homayouni, R., Heinrich, K., Wei, L., Berry, W.: Gene Clustering by Latent Semantic Indexing of MEDLINE Abstracts. Bioinformatics 21, 104–115 (2004)
23. Ding, C.: A Similarity-based Probability Model for Latent Semantic Indexing. In: 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, California, pp. 59–65 (1999)
24. Bartell, B., Cottrell, G., Belew, R.: Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling. In: Proceedings, ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 161–167 (1992)
25. Fagin, R., Kumar, R., McCurley, K., Novak, J., Sivakumar, D., Tomlin, J., Williamson, D.: Searching the workplace web. In: 12th World Wide Web Conference, Budapest (2003) 1581136803/03/0005
26. McCandless, M., Hatcher, E., Mccandless, M.: Lucene in Action. Manning Publications (2009)
27. Smiley, D., Pugh, E.: Solr 1.4 Enterprise Search Server. Packt Publishing (2009)
28. Apache Hadoop, <http://hadoop.apache.org/>
29. Apache Lucene, <http://lucene.apache.org/solr/>
30. Apache Tika, <http://tika.apache.org/>

Trace of Objects to Retrieve Prediction Patterns of Activities in Smart Homes

Farzad Amirjavid, Abdenour Bouzouane, and Bruno Bouchard

Computer Science Department, University of Quebec at Chicoutimi (UQAC)
555, University Boulevard, Chicoutimi, Quebec, Canada
{farzad.amirjavid, abdenour_bouzouane, bruno_bouchard}@uqac.ca

Abstract. An elderly person can forget sometimes to complete the activities that he begins. By the tracing of objects that he may apply in realization of activities in a smart home, it would be possible to predict his intention about what activity he considers to realize. In this way, we would be able to provide hypotheses about the Smart Home resident's goals and his possible goal achievement's defects. To achieve that, spatiotemporal aspects of daily activities are surveyed to mine the patterns of activities realized by the smart homes residents. Based on the inferred daily activities realization patterns we are able to make prediction patterns that can predict uncompleted activities.

Keywords: activity recognition, pattern learning, data mining, objects, concepts.

1 Introduction

Smart homes can be described as augmented residential environments equipped with sensors, actuators and devices. They may be inhabited by the elderly. These residents may be studied to enhance and improve their quality of life and prolong their stay at home with less human needed assistance [3]. In this environment (the Smart Home), embedded sensors can provide data about multiple aspects of activities realized by an Observed Agent (OA) and in this way we would be able to recognize activities.

Most of the introduced non-vision based approaches to conduct activity recognition in smart homes such as [5, 6] are quantitative approaches and their performance depends directly on the quantity of the training tests [1,7]. We infer that quantitative approaches would need a relatively large number of training sets to be trained and the inferred knowledge would depend directly on the quantity of different activities regarding to the total number of training samples [7]. In contrast, we present an idea in that different *styles* of activity realization should be mentioned and quantity of activities realizations in a training set should not determine (at least directly) the chance of accomplishment of an activity by the resident.

We thus propose an approach that handles generated data from multiple embedded sensors in smart homes. Activities are accordingly observed through sensors. In the proposed approach, by use of data mining techniques the primary data is analyzed and realization patterns of activities (induced from trace of objects) are retrieved. The discovered pattern describes complete scenarios by which activities are realized.

By comparing these realization patterns, a prediction pattern itself is retrieved. This prediction pattern explains the possible intentions of ‘OA’ when the resident accomplishes a few actions to achieve a goal or realize an activity.

2 Modeling the Activities Realization and Prediction Patterns

In this paper we introduce a model to learn the Activities Realization Patterns (ARP). This model provides a complete and detailed pattern of activities that in turn as Activities Prediction Patterns (APP) captures the resident’s goal (given that it is his desired activity) from the observation of the accomplishment of only a few initial actions or operations. In this way, we would be able to make hypotheses about these goals, their achievement, defects and anomalies. The mentioned patterns can then be regarded as applicable knowledge for activity and intention recognition.

2.1 Definitions

To explicate our endeavor, we introduce some definitions that are applied in our learning model. These are:

Definition1. *Sensor:* the “world” of the proposed learning problem is observed through set of applied sensors “S”. S_i represents sensor “i” from the set of applied sensors.

Definition2. *Observation:* an observation is considered as value generated from sensors. At each time that we refer to sensors, they generate a special value. These observations represent raw and primary data that are inputted to the learner model. Observation through applied sensors is mentioned as $O_{S,G}$; in which “S” is the set of applied sensors and G is the concerning goal. $O_{S,G}$ represents a set of observations through sensors set “S” and concerning to achievement of goal G (an activity in activity recognition). Observation of sensor “i” in time “t” is referred to as $O_{i,t}$. In fact it is $O_{i,t} \subseteq O_{S,G}$.

Definition3. *Significant difference or change in observation of sensor “i”:* this is referred to as “ e_i ” and indicates a noticeable difference within *consecutive* observations concerning to sensor “i”. In fact,

$$e_i = \{ \langle x_{i,t}, x_{i,t+1} \rangle \mid \langle x_{i,t}, x_{i,t+1} \rangle \in O_i, \mid x_{i,t} - x_{i,t+1} \mid > \varepsilon, \varepsilon > 0 \} \quad \text{and} \quad e_i \subseteq O_i.$$

The interpretation of the term “noticeable difference” (ε) depends to the problem circumstances and the process of its detection that can be designed through the use of the experience of an expert. Whenever the mentioned changed is observed, an “event” is inferred.

Definition4. *Activities Realization Patterns (ARP):* We define Activities Realization Patterns as set of couples constituting from events and their occurrence order. ARP

definition can be demonstrated as $a_j = \{ \langle e_i, t \rangle \}$ and t concerns to the order of occurrence of event “ i ”.

Definition5. *Activities Prediction Patterns (APP):* It expresses the most important parameters and values that best divide the existing inferable activities. It holds ordered information entropy of hypotheses and the most resumed one holds minimum information entropy for realization of all activities. This can be created during a classification process.

3 Pattern Learning

In the proposed model, sensors observe actions concerning each activity and by the tracing of objects and the Observed Agent, we are able to express how the objects change their positions in realization of activities (also the Observed Agent can be traced as an object that is concerned with the realization of activities).

Observation is the first step to learn the patterns of activities realizations. Embedded sensors in the environment conduct the observation. The observation is registered frequently. For example, each six milliseconds an observation is completed by computers that are connected to the sensors and register their generated values.

ARPs are inferred by trace of objects displacements. Objects displacements are inferred if objects get close (or get far) relative to *special points* in the environment.

In this way, activities are defined by the displacement of the objects in space. The position of objects to special geographical points can be described by partial membership of their distances to specific fuzzy classes. For example, “near”, “intermediate distance” and “far”; however, in the implementation two simple concepts; “far” and “near” are applied. The movement of the objects regarding to each geographical point can be made into two ways; “getting closer” or “getting farther”.

All the mentioned definitions are fuzzy terms and can be defined through fuzzy functions [2]. Therefore, at the implementation, instead of directly applying the “*integers* representing the distance of objects”, fuzzy membership measures of the distances are applied to the fuzzy classes. It should be mentioned that we do not pay undue attention to the position of object in x-y page, or their precise distance, but rather their movement and displacement regarding to some specific positions are noticeable.

Implementation of the fuzzy approach lies on the concept of “event” (already discussed in previous sections). In this approach, events are inferred from each meaningful change in position of objects in regard to some already known geographical positions. Observance of special events could mean recognition of a special activity. To infer the occurred event we applied the following membership function to be prepared for the classifier:

$$\mu(far) = \begin{cases} 1 & \text{if } d > b \\ \frac{d-a}{b-a} & \text{if } a < d < b \\ 0 & \text{if } d < a \end{cases}, \mu(close) = \begin{cases} 1 & \text{if } d < a \\ \frac{b-a}{d-a} & \text{if } a < d < b \\ 0 & \text{if } d > b \end{cases}, \begin{matrix} \mu(get_far) = \mu^1_{Object}(far) - \mu^2_{Object}(far) \\ \mu(get_near) = \mu^1_{Object}(near) - \mu^2_{Object}(near) \end{matrix}$$

In the proposed model, the spatiotemporal patterns of activities are learnt through values that are first observed from the world. These are compared to each other and then the displacements of objects are inferred as events. To derive the order of events, we assign a number to the events according to their occurrence order. ARP is in fact the ordered inferred events concerning to the activities. The activities realization pattern expresses the events and the order of events occurrence that should be observed in order to recognize a certain activity. Applying the fuzzy function introduced in the last part, we were able to make sense of fuzzy events and their orders. Here, a tuple combined from couples of “events” and their “occurrence order” indicates the Realization pattern for an activity named ‘activity1’. In this example, O_i represents an object that is applicable in realization of activity1, A_i represents one of the geographical points that are fixed, and displacements of objects are defined relative to them:

Realization Pattern for activity_1 = {< get_close($O1_A1$),1 >, < get_far($O1_A2$),2 >, < get_close($O2_A1$),3 >, < get_close($O3_A1$),4 >, < get_close($O3_A2$),5 >, < get_close($O4_A1$),6 >, < get_far($O4_A2$),7 >}

In summary, through these inferred daily activities realization patterns we are able to infer the needs of the resident.

4 Concluding Remarks

We introduced an approach in which the summarization of the observations tries to provide suitable information for the classifier algorithm. Applying fuzzy spatiotemporal reasoning, we are able to make Activities *Realization* Patterns, which provides us with a complete explanation about actions and their accomplishment order. Based on these inferred daily activities realization patterns we are able to produce prediction patterns, namely Activities *Prediction* Patterns (APP) that can usefully predict the uncompleted activities of these residents.

References

1. Amirjavid, F., Bouzouane, A., Bouchard, B.: Action recognition under uncertainty in smart homes. In: MAICS (2011)
2. Zadeh, L.A.: Probability measures of fuzzy events. *Journal Math. Anal. Appl.* 23, 421–427 (1968)
3. Bouchard, B., Bouzouane, A., Giroux, S.: A Keyhole Plan Recognition Model for Alzheimer’s Patients: First Results. *Journal of Applied Artificial Intelligence (AAI)* 22(7), 623–658 (2007)
4. Jakkula, V., Cook, J.: Temporal pattern discovery for anomaly detection in a smart home. In: 3rd IET International Conference on Intelligent Environments, IE 2007 (2007)
5. Galushka, M., Patterson, D., Rooney, N.: Temporal data mining for smart homes (2007)
6. Mitsa, T.: Temporal data mining. Chapman and Hall/CRC (2010)
7. Ross Quinlan, J., Ghosh, J.: Top 10 Algorithms in Data Mining. In: IEEE international Conference on Data Mining, ICDM (December 2006)

Distributed Context Aware Collaborative Filtering Approach for Service Selection in Wireless Mesh Networks

Neeraj Kumar¹ and Kashif Iqbal²

¹ School of Computer Science & Engineering, SMVD University, Katra (J&K), India

² Department of Computing and Digital environment, Coventry University, UK
neeraj.nehra@smvdu.ac.in, k.iqbal@coventry.ac.uk

Abstract. In last decade, there is a paradigm shift in technology in the sense that large numbers of users over the internet share the valuable information with others. Users working in this field work at different levels for information sharing. As these users share the information with each other, there is a need of efficient collaborative mechanism among them to achieve efficiency and accuracy at each level. So to achieve high level of efficiency and accuracy, a distributed context aware collaborative filtering (CF) approach for service selection is proposed in this paper. Users profiles are created as a database repository from the previous data of different users and their respective interests. For the new user who wants to avail a particular service, system matches the request with the existing users profiles and if the match is found then a suitable service is recommended to him based upon his profile. To select the relevant contents of user choice that match his profile with the existing users, a Distributed Filtering Metric (DFM) is included which is based upon user input. Moreover, the intersection of existing users profiles and their interests is also included in this metric to have high level of accuracy. Specifically, we have taken an example of movie selection as a service offered to the users by some network. The underlying network chosen is Wireless Mesh Networks (WMNs) which are emerged as a new powerful technology in recent years due to the unique features such as low deployment cost and easy maintenance. A novel Context Aware Service Selection (CASS) algorithm is proposed. The performance of the proposed algorithm is evaluated with respect to efficiency and accuracy. The results obtained show that the proposed approach has high level of efficiency and accuracy.

Keywords: Collaborative filtering, service selection, wireless mesh networks.

1 Introduction

In last few years, with an increase in the use of large number of social networking sites, large amount of data flows between users working in this domain. The users share valuable information with each other on the fly in this new computing environment. This type of environment in which users collaborate with each other for sharing

valuable information is known as collaborative computing [1]. The resources in collaborative computing are provided to the end users by different types of networks which may be wired or wireless. It has been found in literature that wireless mesh networks (WMNs) is a new emerging powerful technology for providing resources to the end users so that users can collaborate with each others [2, 3, 4]. These networks are special type of Ad hoc networks having multiple hops, and are self configured, self healing and cost effective and have many advantages over traditional wireless networks, such as robustness, greater coverage, low up-front costs with ease of maintenance and deployment [2].

The availability of context aware data in this network is a challenging task and it has been found that Collaborative filtering (CF) [1, 5] technique can be an efficient solution to filter the relevant context aware contents and provide the desired services to the users. There are number of proposals exist in the literature for CF based context aware service selection. But the existing CF methods have limitations that they have considered user or item-based ratings which reduce the chances of exact filtering in any collaborative computing environment [6, 7]. To overcome these limitations, in this paper a novel context aware service selection (CASS) algorithm is proposed. A new Distributed Filtering Metric (DFM) is included which selects the contents based upon user rating given by the users in an interactive manner.

Rest of the paper is organized as follows. Section 2 describes the related work. Section 3 describes the system model. Section 4 describes the proposed algorithm. Section 5 discusses the simulation and results. Section 6 concludes the article.

2 Related Work

There are various proposals for collaborative filtering in literature. Liu et al. [1] proposed a hybrid collaborative filtering recommendation for P2P networks. In this proposal, authors proposed both user and item based rating mechanism to improve the predictive accuracy. Hu et al. [8] proposed a hybrid user and item based collaborative filtering with smoothing on sparse data. Marc et al. [9] present eSciGrid for efficient data sharing in which the physical distance between elements and the amount of traffic carried by each node is considered. Wang et al. present a unified probabilistic model for collaborative filtering using Parzen-window density estimation for acquiring the probabilities of the proposed unified relevance model [10, 11]. Alexandrin et al. [12] proposed probabilistic model for unified and collaborative contents based recommender system. Wang et al. [13] present a unified relevance models for rating prediction in collaborative filtering. Goldbergh et al. [14] present a constraint time collaboration filtering algorithm. Hofmann et al. [15] proposed a collaborative filtering using Gaussian latent semantic analysis.

It is difficult to make the accurate decisions and to decide the similarities in user behaviors which interact with each others. So the main objective of CF is to make the predictions about the user preferences and behaviors to minimize the prediction errors by accurate decisions [16-19]. These accurate decisions are helpful in various social networking sites that provide dynamic and collaborative communication, interaction and knowledge sharing [20-22].

3 System Model

The proposed system model is shown in figure 1. There are two domains in the proposed system and a centralized data repository which is shared by the users working in these domains. Users working in different domains may be located at different locations such as govt. buildings, houses, city locations etc. as shown in figure 1. To provide the users a service of their choices, user rating is taken about that service on the maximum and minimum scale and a rating matrix is constructed for the same. Once the rating is taken then the desired service is located. The service may be located at local site or at global sites in distributed manner.

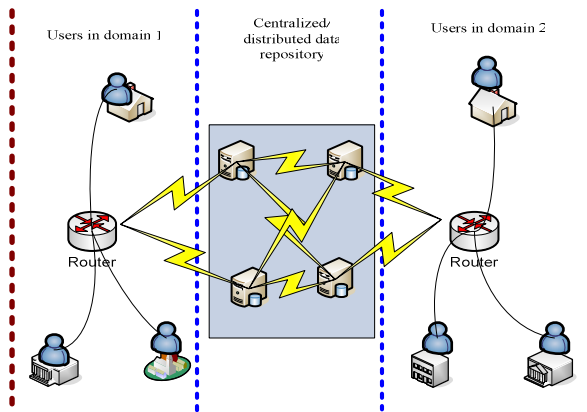


Fig. 1. System model in the proposed approach

3.1 Problem Definition

Over the years, number of applications have been developed and evaluated by taking into consideration the view of the users in which users take decisions in evaluating collaborative filtering approach [23]. Most recently authors in [24] propose a new metric to improve the behavior of the recommender system. The proposed metric combines the numerical information of the votes with independent information from those values, based on the proportions of the common and uncommon votes between each pair of users. The most common methods to correlate the data for the users based upon the similarity behaviors are: Pearson correlation, cosine, constrained Pearson's correlation and spearman correlation [24]. Among these methods the most common Pearson correlation is the most commonly used method. But it has disadvantages such as there must exist linear relationship between the variables to be correlated and both the variables should be normally distributed [24]. Although authors in [24] enhance the existing Pearson metric by considering the mean square difference (MSD) in to account. But we propose a new Distributed Filtering Metric (DFM) which takes into account the network aspect also. The formulization of DFM is as follows:

Let there are N users and S services to be recommended by these users. Each user can recommend a service based upon the rating in the range $[1 - 5]$ with 1 as the minimum and 5 as the maximum rating. For a non rated service, the aggregate rating of s most similar items by the users is counted. Based upon the rating values a User Rating Matrix (URM) is constructed as follows:

$$URM = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1N} \\ R_{21} & R_{22} & \dots & R_{2N} \\ \dots & \dots & \dots & \dots \\ R_{S1} & R_{S2} & \dots & R_{SN} \end{bmatrix}, \text{ where } 1 \leq R_{ij} \leq 5 \text{ for } 1 \leq i \leq N, 1 \leq j \leq S$$

Each service S_i is defined by the three parameters as $S_i = (loc_i, R_{ij}, avail_i)$, where loc_i specifies the location of a particular service, i.e., local or global and $avail_i$ specifies the availability of the service. The variable loc_i has values with L as local and G as global while $avail_i$ is a probabilistic variable whose value changes with selection and execution of users recommendations. The value of the variable R_{ij} is as defined in URM matrix.

The DFM metric is defined as follows:

$$DFM(S_i) = loc_i * R_{ij} * avail_i \dots\dots\dots(1)$$

$$\text{Where } avail_i = f(P_i * N) \dots\dots\dots(2)$$

Where $P_i = \frac{\text{number_of_search_in_local_and_global}}{\text{total_number_of_search}}$, f is a flag to be set

according to the value of variable P_i . Its value is in the interval $[0, 1]$ depending upon successful or unsuccessful search in local or global scope.

There may be various service classes for different applications. Each service offered to user belongs to one of the service class. Each service has duration time either locally or globally which is exponentially distributed with mean values as $1/\alpha$ and $1/\beta$. So the average time for service availability is $\frac{\beta}{\alpha + \beta}$.

The combined probability of service availability is

$$P(S_i) = P_i * \left(\frac{\beta}{\alpha + \beta}\right)^L * \left(1 - \frac{\beta}{\alpha + \beta}\right)^G \dots\dots\dots(3)$$

So the maximum number of services available in a particular time interval would be $P(S_i)^{\max} = \max\{(P(S_i) * f) > 1 - \lambda\} \dots\dots\dots(4)$

Where $0 < \lambda < 1$ is the service time of the service S_i .

Equation (4) gives the maximum probability of service availability in a particular time interval with a given service time and locality of that service.

Let the time required for service failure and recover is described by exponential function as

4 Proposed Approach

4.1 Rating by Users for a Particular Service

In the proposed approach, users can provide the rating in an interactive manner to a particular service and also get feedback about this service. User can rate a particular service based upon the rating as defined above and URM is constructed. In the proposed scheme, services are assumed to be located at different locations in distributed manner, hence each service is characterized by three parameters namely as location, availability and rating as provided by the user in URM. Once the rating is provided by the users, the next task is to select and provide the desired service to the users. The detailed algorithm is as follows:

4.2 Context Aware Service Selection (CASS) Algorithm

Input: number of services and users, rating and location of services

Output: Service selection and provision to user

Initialize the states of the services as follows:

Procedure Available (S_i , type of services, number of users)

Construct the network of services available and number of users demanding the service

 Locate the service at local and global site

If (service == *avail_i*)

 Call Procedure Processing (S_i , location)

 Allocate the service to the users

End if

End Procedure

Procedure Processing (S_i , location)

 Take the initial rating from the users about the available services as

$R = R^{init}$ and construct the URM

 Call Procedure Filtering (S_i , DFM)

while(($S_i \neq \phi$) & & ($N \neq \phi$))

if ($S_i(avail_i) == L$) || $S_i(avail_i) == G$)

Calculate the average service availability time $\frac{\beta}{\alpha + \beta}$ and combined service availability probability as above from equation (3)

End Procedure

Procedure Filtering (S_j , DFM)

Use DFM metric to filter the contents of the service and values of rating as given by the users in URM and take the intersection of existing users profiles
 Select the service having highest DFM value

End Procedure

5 Results and Discussion

The performance of the proposed system is evaluated with respect to the metrics such as efficiency and accuracy. The data is selected from well known and widely used Movie Lens (<http://www.MovieLens.umn.edu>) and Jester Data sets. More than 500 users are selected from the database and these are divided into different groups of users. These users provide the rating to different services provided in WMNs based upon standard scale of 1-5 as defined above. Based upon the inputs and recommendations from the users the proposed scheme is evaluated with respect to the above metrics.

Efficiency and accuracy

Figures 2 and 3 describe the performance of the proposed scheme with respect to scalability and accuracy with an increase in number of items/services. The results obtained show that the proposed scheme has high level of accuracy and efficiency with an increase in number of data items. This is due to the fact that the proposed scheme uses the DFM metric for content selection which in return the most suitable service for the users. As soon as the relevant contents are matched corresponding to the user profile, a particular service is recommended to him. This shows the effectiveness of the proposed scheme.

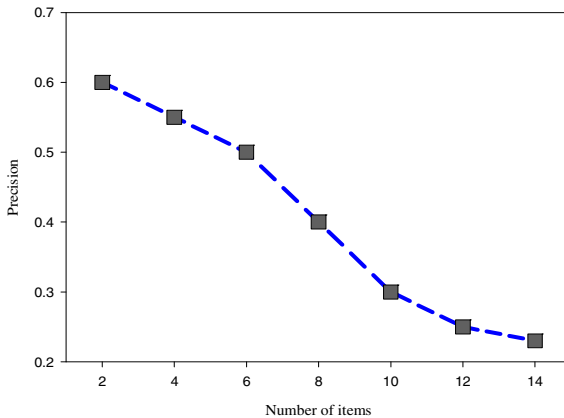


Fig. 2. Precision with number of items

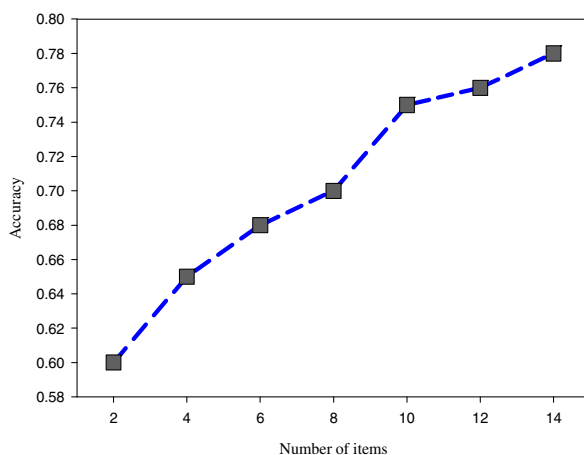


Fig. 3. Accuracy with number of items

6 Conclusions

In recent times, collaborative computing has emerged as a new computing paradigm in which various users working in different domains share their resources in an efficient manner. It has been found that in most of the cases, users are interested in viewing only contents of a particular service of their choices. So collaborative filtering plays a crucial role in selection of the contents of user choice. In this paper, we propose a novel distributed collaborative filtering approach for service selection and provision in wireless mesh networks. A novel Context Aware Service Selection (CASS) algorithm is proposed. The proposed algorithm filter the information using distributed collaborative filtering and a new Distributed Filtering Metric (DFM) which selects the contents based upon the user input keeping in view of the users needs. The performance of the proposed algorithm is evaluated with respect to scalability and accuracy. The results obtained show that the proposed algorithm has high level efficiency and accuracy.

References

- [1] Liu, Z., Qu, W., Li, H., Xie, C.: A hybrid collaborative filtering recommendation mechanism for P2P networks. *Future Generation Computer Systems* 26, 1409–1417 (2010)
- [2] Akyildiz, F., Wang, X., Wang, W.: Wireless Mesh Networks: a survey. *Computer Networks* 47(4), 445–487 (2005)
- [3] Rodríguez-Covili, J., Ochoa, S.F., Pino, J.A., Messeguer, R., Medina, E., Royo, D.: A communication infrastructure to ease the development of mobile collaborative applications. *Journal of Network and Computer Applications* (2010)
- [4] Lee, W.-H., Tseng, S.-S., Shieh, W.-Y.: Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system. *Information Sciences* 180, 62–70 (2010)

- [5] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of ACM Conference on Computer Supported Cooperative Work (1994)
- [6] Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–46 (2007)
- [7] Sullivan, D.O., Wilson, D., Smyth, B.: Preserving recommender accuracy and diversity in sparse datasets. In: FLAIRS Conference 2003, pp. 139–143 (2003)
- [8] Hu, R., Lu, Y.: A hybrid user and item-based collaborative filtering with smoothing on sparse data. In: Pan, Z., Cheok, D.A.D., Haller, M., Lau, R., Saito, H., Liang, R. (eds.) ICAT 2006. LNCS, vol. 4282, pp. 184–189. Springer, Heidelberg (2006), doi:10.1109/ICAT.2006.12
- [9] Sánchez-Artigas, M., García-López, P.: eSciGrid: A P2P-based e-science Grid for scalable and efficient data sharing. *Future Generation Computer Systems* 26(5), 704–719 (2010)
- [10] Wang, J., De Vries, A.P., Reinders, M.J.T.: A user_item relevance model for logbased collaborative filtering. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavilinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 37–48. Springer, Heidelberg (2006)
- [11] Wang, J., De Vries, A.P., Reinders, M.J.T.: Unifying user-based and item based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 501–508. ACM Press, New York (2006)
- [12] Popescul, A., Ungar, L.H., Pennock, D.M., Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, pp. 437–444 (2001)
- [13] Wang, J., de Vries, A.P., Reinders, M.J.T.: Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems* 26(3), 1–42 (2008)
- [14] Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: a constant time collaborative filtering algorithm. *Information Retrieval* 4(2), 133–151 (2001)
- [15] Hofmann, T.: Collaborative filtering via Gaussian probabilistic latent semantic analysis. In: Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2003)
- [16] Adomavicius, E., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6), 734–749 (2005)
- [17] Herlocker, J.L., Konstan, J.A., Riedl, J.T., Terveen, L.G.: Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22(1), 5–53 (2004)
- [18] Ryan, P.B., Bridge, D.: Collaborative recommending using formal concept analysis. *Knowledge-Based Systems* 19(5), 309–315 (2006)
- [19] Pu, P., Chen, L.: Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems* 20(6), 542–556 (2007)
- [20] Giaglis, G.M., Lekakos, G.: Improving the prediction accuracy of recommendation algorithms: approaches anchored on human factors. *Interacting with Computers* 18(3), 410–431 (2006)
- [21] Fuyuki, I., Quan, T.K., Shinichi, H.: Improving accuracy of recommender systems by clustering items based on stability of user similarity. In: Proceedings of the IEEE International Conference on Intelligent Agents, Web Technologies and Internet Commerce (2006), doi:10.1109/CIMCA.2006.123

- [22] Manolopoulos, Y., Nanopoulos, A., Papadopoulos, A.N., Symeonidis, P.: Collaborative recommender systems: combining effectiveness and efficiency. *Expert Systems with Applications* 34, 2995–3013 (2007)
- [23] Herlocker, J.L., Konstan, J.A., Riedl, J.T., Terveen, L.G.: Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22(1), 5–53 (2004)
- [24] Bobadilla, J., Serradilla, F., Bernal, J.: A new collaborative filtering metric that improves the behaviour of recommender systems. *Knowledge-Based Systems* 23, 520–528 (2010)

A Framework for the Evaluation of Adaptive IR Systems through Implicit Recommendation

Catherine Mulwa, Seamus Lawless, M. Rami Ghorab,
Eileen O'Donnell, Mary Sharp, and Vincent Wade

Knowledge and Data Engineering Research Group,
School of Statistics and Computer Science,
Trinity College Dublin
{mulwac, ghorabm, odonee, seamus.lawless,
mary.sharp, vincent.wade}@scss.tcd.ie

Abstract. Personalised Information Retrieval (PIR) has gained considerable attention in recent literature. In PIR different stages of the retrieval process are adapted to the user, such as adapting the user's query or the results. Personalised recommender frameworks are endowed with intelligent mechanisms to search for products, goods and services that users are interested in. The objective of such tools is to evaluate and filter the huge amount of information available within a specific scope to assist users in their information access processes. This paper presents a web-based adaptive framework for evaluating personalised information retrieval systems. The framework uses implicit recommendation to guide users in deciding which evaluation techniques, metrics and criteria to use. A task-based experiment was conducted to test the functionality and performance of the framework. A Review of evaluation techniques for personalised IR systems was conducted and the results of the analysed survey are presented.

Keywords: Personalisation, Personalised Information Retrieval Systems, Implicit Recommendations, User-based Evaluation, Task-based Evaluation.

1 Introduction

Evaluation has been an integral part of Information Retrieval (IR) research from its early days with the Cranfield experiments (Cleverdon et al. 1966) that used pre-defined queries that were run against a test collection in batch mode. One major problem with traditional IR systems is that they provide uniform access and retrieval results to all users, solely based on the query terms the user issued to the system. Evaluation frameworks for personalised information retrieval systems (PIRS) and adaptive information retrieval systems (AIRS) are necessary to "better interpret and give more exact hints and false inferences than a simple global vision, thus facilitating the improvement of applications and services, when required, as well as the generalisation and reuse of results"(Tobar 2003). In this paper we call systems which combine Adaptive Hypermedia (AH) and IR approaches, AIRS (Lawless et al. 2010).

The main aim of the framework described in this paper is to provide comprehensive support for users through implicit recommendations on which evaluation methods, metrics and criteria should be used to evaluate these systems and how to best combine these approaches. Access to this repository of evaluation approaches is supported for geographically distributed users of any nationality by facilitating dynamic translation of content. The authors acknowledge that personalisation in IR is aimed at improving the user's experience by incorporating user subjectivity in the retrieval process.

The rest of this paper is structured as follows: Section 2 presents a summary of a comparison of personalisation approaches and evaluation techniques in PIR systems. Section 3 introduces the proposed framework to evaluate AIRS systems. Also the methodology, architecture design, functionality and evaluation of the framework are also introduced. Finally section 4 concludes the paper and proposes future work.

2 A Review of Personalisation Approaches and Evaluation Techniques for AIRS

Personalised Information Retrieval (PIR) is a research area which has gained attention in recent literature and is motivated by the success of both areas, IR and AH (Gauch et al., 2007, Micarelli et al., 2007). IR systems have the advantage of scalability when dealing with large document collections and performing a vast amount of information processing. AH systems have the advantage of including the user in the process and thus the ability to satisfy individual user needs by modeling different aspects of the user. In PIR, different stages of the retrieval process are adapted to the user such as adapting the user's query and/or the results. This review focuses on personalisation approaches and existing evaluation techniques for PIR systems.

2.1 Overview of Personalisation Approaches

Personalisation can be performed on an individualised, collaborative, or aggregate scope. Individualised personalisation is when the system's adaptive decisions are taken according to the interests of each individual user as inferred from their user model (Speretta and Gauch, 2005, Teevan et al., 2005). Collaborative personalisation is when information from several user models is used to determine or alter the weights of interests in other user models (Sugiyama et al., 2004). This is usually used when a system groups the users into a number of stereotypes according to certain similarity criteria between their user models; at which point the system can judge the relevance of a certain item or document to a user based on information from other user models that belong to the same group. Stereotypes can be manually pre-defined or automatically learnt using machine learning techniques (e.g. clustering techniques). Personalisation can be implemented on an aggregate scope when the system does not make use of user models; in which case personalisation is guided by aggregate usage data as exhibited in search logs (i.e. implicitly inferred general users' interests from aggregate history information) (Smyth and Balfe, 2006, Agichtein et al., 2006).

The authors acknowledge that user-based evaluation of personalised IR systems is challenging because of the user effect in terms of the inconsistency in ranking and in

relevance criteria usage. End-users are seen as the ultimate assessors of the quality of the information and of the systems as well as services that provide information (Barry and Schamber, 1998). User satisfaction is a composite term; amalgamating a cluster of “felt experience”. Table 1 provides a comparison of the surveyed systems in the literature. The comparison focuses on the personalisation implementation stage of the surveyed systems, guided by the three classification criteria (i.e. individualised, collaborative and aggregate usage data).

Table 1. Comparison of Personalisation Approaches (Ghorab et al., 2011)

Application Area	Personalisation Scope	Personalisation Approach	Published Study
Monolingual IR	Individualised	Result Adaptation (result re-ranking)	(Speretta and Gauch 2005), (Stamou and Ntoulas 2009), (Teevan et al.2005), (Pretschner and Gauch 1999)
Monolingual IR & Information Filtering	Individualised	Result Adaptation (result re-ranking)	(Micarelli and Sciarone 2004)
Monolingual IR	(1)Individualised & (2) Collaborative	Result Adaptation (result re-ranking)	(Sugiyama et al. 2004)
Monolingual IR	Aggregate usage data	Result Adaptation (result re-ranking)	(Smyth and Balfe 2006)
Monolingual IR	Aggregate usage data	Result Adaptation ((1)result scoring & (2)result re-ranking)	(Agichtein et al. 2006)
Information Filtering	Individualised	Result Adaptation (result scoring)	(Stefani and Strapparava 1999)
Monolingual IR	Individualised	Query Adaptation (query expansion using keywords from user model)	(Chirita et al. 2007)
Structured Search on a Database	Individualised	Query Adaptation (query rewriting)	(Koutrika and Ioannidis 2004)
Cross-lingual IR	Aggregate usage data	Query Adaptation (query suggestions using similar queries from multiple languages)	(Gao et al. 2007)
Monolingual IR	Individualised	Query & Result Adaptation (query expansion using keywords from user model, and result re-ranking)	(Pitkow et al. 2002)

2.2 Evaluation Approaches for PIR Systems

The evaluation of PIR systems is challenged by user effect, which is manifested in terms of users’ inconsistency in relevance judgment ranking and relevance criteria usage. Personalisation in PIR systems is generally performed by adapting the query and/or the results to the user’s interests. Adaptation can either target specific individualized user needs, or target common needs of groups of users. Personalised systems involve information about users in the process and therefore adapt the retrieval process to the users’ needs. In other words, a PIR system does not retrieve documents that are just relevant to the query but ones that are also relevant to the user’s interests.

Table 1. Comparison of Evaluation Techniques (Ghorab et al., 2011)

Scope of Evaluation	Evaluation Metric & Instrument	Experimental Setting	Example Publications
System Performance (retrieval process)	Quantitative (Precision at K, Recall at K, F-measure, Break-even point)	Controlled setting (47 users, 25 information needs per user, open web corpora via meta search engine)	(Smyth and Balfre 2006)
System Performance (retrieval process)	Quantitative (R-precision)	Controlled setting (20 users, 50 information needs per user, open web corpora via Google wrapper)	(Sugiyama et al. 2004)
System Performance (retrieval process)	Quantitative (Normalised Discounted Cumulative Gain (NDCG))	Controlled setting (15 users, 10 information needs per user, open web corpora via MSN Search)	(Teevan et al. 2005)
System Performance (retrieval process)	Quantitative (Normalised Discounted Cumulative Gain (NDCG))	Controlled setting (18 users, 4 information needs per user, open web corpora via Google wrapper)	(Chirita et al. 2007)
System Performance (retrieval process)	Quantitative (rank scoring based on explicit relevance judgments by users)	Controlled setting (11 users, 68 information needs per user on average, open web corpora via Google wrapper)	(Stamou and Ntoulas 2009)
System Performance (retrieval process)	Quantitative (rank scoring based on implicit relevance judgments from clickthrough)	Controlled setting (6 users, 2 information needs per user, open web corpora via Google wrapper)	(Speretta and Gauch 2005)
System Performance (retrieval process)	Quantitative (Precision at K(P@K), Normalised Discounted Cumulative Gain (NDCG), and Mean Average Precision (MAP))	Large-scale setting (12 million interactions by users, 3000 randomly selected queries out of 1.2 million unique queries, open web corpora using a major search engine)	(Agichtein et al. 2006)
System Performance (retrieval process)	Quantitative (11-point precision)	Large-scale setting (7 million unique English queries from MSN Search logs, 5000 randomly selected French queries out of 3 million queries from a French query log, 25 French-English query pairs, TREC-6 collection)	(Gao et al. 2007)
System Performance (user model & retrieval process)	Qualitative & Quantitative (questionnaires for users about how well the model depicted their interests & 11-point precision)	Controlled setting (16 users, 3 information needs per user, open web corpora via ProFusion)	(Pretschner and Gauch 1999)
System Usability & Performance (usability & retrieval process)	Qualitative & Quantitative (usability questionnaire & 11-point precision, rank scoring based on explicit relevance judgments by users)	Controlled setting (24 users, 15 information needs per user, open web corpora via Alta-Vista wrapper)	(Micarelli and Sciarone 2004)
User Performance (task-based)	Quantitative (time and number of actions needed to complete search tasks)	Controlled setting (48 users, 12 information needs per user, open web corpora via Google wrapper)	(Pitkow et al. 2002)

3 The Proposed Personalised Framework

3.1 Methodology and Architectural Approach

The rational unified process (RUP) Methodology was used in the design and implementation of the framework described by this paper. The RUP methodology is significant with respect to: i) conducting iterative development, ii) requirements management, iii) designing a component-based architecture iv) visual modeling of the system, v) quality management and vi) change control management. The user-centred

evaluation approach is used in order to verify the quality of an AIRS, detecting problems in the system functionality or interface, and supporting adaptivity decisions.

The framework is designed as a web-based 3-tier architecture, as can be seen in Figure 1, which consists of: *i) the presentation layer*, *ii) The business logic layer* which is pulled out from the presentation tier, it controls the frameworks functionality by performing detailed processing and *iii) the data persistence layer* which keeps data neutral and independent from application servers or business logic. The framework is divided into 4 major sections (i.e. the recommender, repository for current studies and search interface, and a user-centred evaluation methodology).

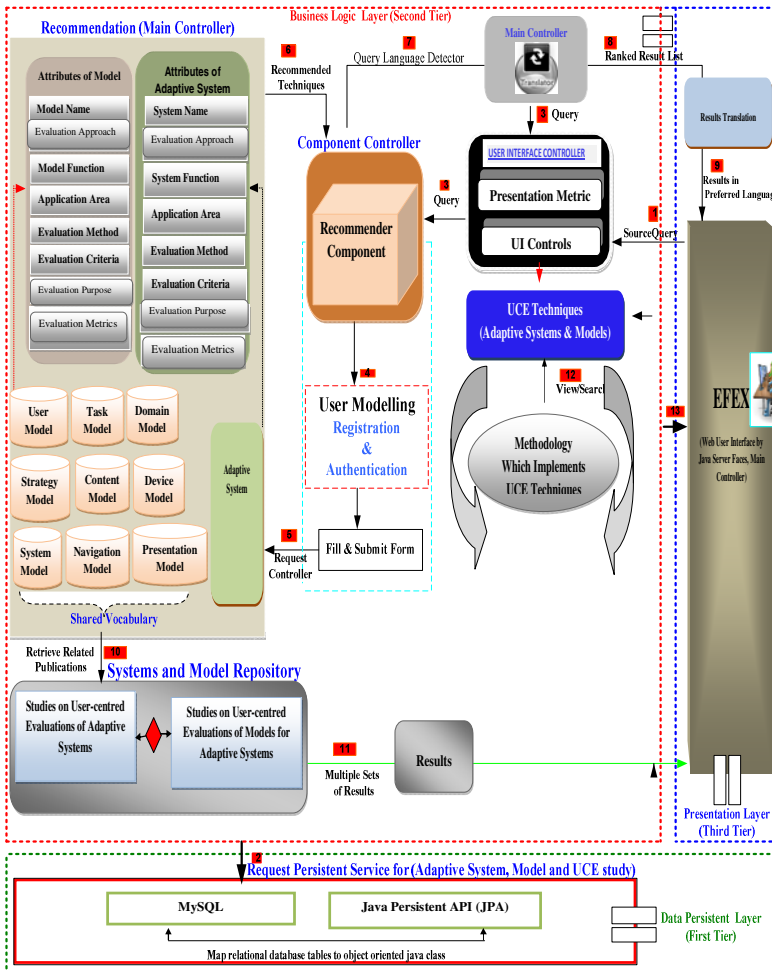


Fig. 1. Architectural Design of Proposed Personalisation Framework

3.2 Implementation and Technologies Used

A combination of several technologies was used to implement the framework: **Net-Beans 6.9** - platform, **Apache Lucene** - Search engine, **Apache OpenJPA** - To store and Retrieve data from the database, **Apache Tomcat** - server, **Myfaces-core** - Java Server Faces (JSF) used to display data on the Web, **MySql-win32** - MySql database server, **MySql-connector-java** - Connector for java to communicate to mySql, **Google Translate** – to translate the presented information into users choice of language, **Json** - To parse translations.

3.3 Proposed Implicit Recommendation Algorithm

The algorithm implemented in this framework applies implicit recommendation techniques to personalise and recommend evaluation methods, metrics and criteria. Suppose two types of users want to use the framework: i) **User A** wants to get recommendations on how to evaluate an AIRS system. The user does not know which methods, criteria or metrics to use; ii) **User B** wants to get recommendations on how to evaluate an AIR system he/she has developed. The user knows which methods, criteria or metrics to use, but is not sure whether they are the most appropriate ones. He/she wants recommendations on how to evaluate his system. Using the algorithm provided in Figure 2, the framework provides implicit recommendations to the users.

Start:

Step1: The user selects the system categories and approach in the initial steps.

Step 2. Using the categories selected the recommender does the following

- a. Select all the systems belonging to these categories
- b. Select all the evaluations that have been carried out on these systems
- c. Using the approach of these evaluations all the methods, metrics and criteria are retrieved from database together with their evaluation results.
- d. All the evaluation results for each method, metric and criteria are stored in a list.
- e. Each result has a success score and a flag as to whether this evaluation was carried out specifically for this system or not. If it was it is given extra weight in the scoring process.
- f. When all the results for each method, metric and criteria are collated they are added up and the list is sorted by score.
- g. The results are presented as a percentage of the highest score in the list which will always have 100%
- h. If the methods, metrics and criteria in the list match the methods, metrics and criteria being used in the current evaluation then they may be highlighted in the list.

End

Fig. 2. Functionality of the Recommender Algorithm

3.4 Benefits and Functions of the Framework

Users of the framework are provided with personalised information to suit the user's requirements. In this case the framework considers the users interests and preferences in order to provide personalised services. Users are able to:

- Search for literature published from 2000 to date, such as user-centred evaluation (UCE) studies or evaluations of adaptive systems (i.e. adaptive hypermedia, adaptive educational hypermedia, adaptive e-learning, adaptive recommender, PIR and AIRS systems). The query results presented to the user are based on the following characteristics of the evaluated system: system name, developer, evaluation approach, evaluation purpose, system description, application area, evaluation methods, evaluation criteria, evaluation metrics, year of evaluation and finally what was improved by the adaptation.
- Get implicit recommendations on how to combine different evaluation methods, metrics and measurement criteria in order to evaluate a specific system.
- Translate the user interface into 49 different languages to suit the user.

3.5 Task-Based Experiments and User Evaluations

To evaluate the framework, three phases of evaluation were defined (requirements specification, preliminary evaluation and final evaluation phase). For each phase, the appropriate evaluation methods, metrics and criteria were identified. Currently, only the requirement specifications and preliminary evaluations have been conducted. This involved interviewing 12 domain experts and conducting a task-based experiment. The use of interviews provided qualitative feedback on user experience after using the framework. The experiment was designed based on a task-based problem scenario.

The task based experiment was significant in evaluating the overall performance and usefulness of the developed framework. In this case, 10 test users were presented with a list of tasks. The techniques adopted was based on internal quality estimation consisting of six characteristics: i) functionality, concerned with what the framework does to fulfil user needs; ii) reliability, evaluating the frameworks capability to maintain a specified level of performance; iii) usability, assessing how understandable and usable the framework is; iv) efficiency, evaluating the capability of the framework to exhibit the required performance with regards to the amount of resources needed; and v) maintainability, concerned with the framework's capability to be modified and finally portability, which will involve measuring the frameworks capability to be used in a distributed environment.

The results from the requirements specification and preliminary evaluation phase were used to improve the functionality of the developed framework. A major evaluation will be conducted for the final phase. This will involve a large number of users performing several tasks.

4 Conclusion and Future Work

This paper described a review and classification of personalised IR approaches and evaluation techniques for PIR systems in the literature. Future personalised IR systems could build on harnessing the benefits of both implicit and explicit approaches to gathering user information and feedback about the user's searches. There are currently no standard evaluation frameworks for AIRS systems. The framework presented in this paper will be a significant contribution to both the AH and IR scientific communities. Evaluators of AIRS systems should ensure that the correct evaluation methods, metrics and criteria are used while evaluating these systems. Two major evaluations

of the framework will be conducted in future to test the: i) usability and performance of the overall framework and ii) end-user experience of using the framework.

Acknowledgements. This research is based upon works supported by Science Foundation Ireland (Grant Number: 07/CE/I1142) as part of the Centre for Next Generation Localization (www.cngl.ie). The authors are grateful for the suggestions of the reviewers for this paper.

References

1. Agichtein, E., Brill, E., Dumais, S.: Improving Web Search Ranking by Incorporating User Behavior Information. In: 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006). ACM, Seattle (2006)
2. Barry, C.L., Schamber, L.: Users' criteria for relevance evaluation: a cross-situational comparison. *Information processing & management* 34, 219–236 (1998)
3. Chirita, P.-A., Firan, C., Nejdl, W.: Personalised Query Expansion for the Web. In: 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007). ACM, Amsterdam (2007)
4. Cleverdon, C.W., Mills, J., Keen, E.M.: An inquiry in testing of information retrieval systems (vols. 2) (Cranfield, UK: Aslib Cranfield Research Project, College of Aeronautics) (1966)
5. Gao, W., Niu, C., Nie, J.-Y., Zhou, D., Hu, J., Wong, K.-F., Hon, H.-W.: Cross-Lingual Query Suggestion Using Query Logs of Different Languages. In: 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007). ACM, Amsterdam (2007)
6. Ghorab, M.R., Zhou, D., O'Connor, A., And Wade, V.: Users' Search History Approach for Personalised Information Retrieval: survey and Classification. Submitted to the Journal of User Modelling and User Adapted Interaction (2011) (Under Review)
7. Lawless, S., Mulwa, C., O'Connor, A.: A Proposal for the Evaluation of Adaptive Personalised Information Retrieval. In: Proceedings of the 2nd International Workshop on Contextual Information Access, Seeking and Retrieval Evaluation, Milton Keynes, UK. CEUR-WS.org 4, March 28 (2010)
8. Koutrika, G., Ioannidis, Y.: Rule-based Query Personalised in Digital Libraries. *International Journal on Digital Libraries* 4, 60–63 (2004)
9. Micarelli, A., Sciarrone, F.: Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. *User Modeling and User-Adapted Interaction* 14, 159–200 (2004)
10. Pretschner, A., Gauch, S.: Ontology Based Personalised Search. In: 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 1999). IEEE, Chicago (1999)
11. Pitkow, J., Schutze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., Breuel, T.: Personalised Search. *Communications of the ACM* 45, 50–55 (2002)
12. Smyth, B., Balfe, E.: Anonymous Personalised in Collaborative Web Search. *Information Retrieval* 9, 165–190 (2006)
13. Speretta, M., Gauch, S.: Misearch. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005), Compiegne University of Technology. IEEE Computer Society, France (2005a)
14. Stamou, S., Ntoulas, A.: Search Personalised Through Query and Page Topical Analysis. *User Modeling and User-Adapted Interaction* 19, 5–33 (2009)

15. Stefani, A., Strapparava, C.: Exploiting NLP Techniques to Build User Model for Web Sites: the Use of WordNet in SiteIF Project. In: 2nd Workshop on Adaptive Systems and User Modeling on the World Wide Web, Toronto, Canada (1999)
16. Sugiyama, K., Hatano, K., Yoshikawa, M.: Adaptive Web Search Based on User Profile Constructed without Any Effort from Users. In: 13th International Conference on World Wide Web. ACM, New York (2004)
17. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing Search via Automated Analysis of Interests and Activities. In: 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005). ACM, Salvador (2005)
18. Tobar, C.M.: Yet another evaluation framework. In: Second Workshop on Empirical Evaluation of Adaptive Systems is part of the 9th International Conference on User Modeling (2003)

MedMatch – Towards Domain Specific Semantic Matching

Jetendr Shamdasani, Peter Bloodsworth, Kamran Munir,
Hanene Boussi Rahmouni, and Richard McClatchey

CCS Research Centre, FET Faculty,
University of the West of England Coldharbour Lane,
Frenchay, Bristol BS16 1QY, UK

Abstract. Ontologies are increasingly being used to address the problems of heterogeneous data sources. This has in turn often led to the challenge of heterogeneity between ontologies themselves. Semantic Matching has been seen as a potential solution to resolving ambiguities between ontologies. Whilst generic algorithms have proved successful in fields with little domain specific terminology, they have often struggled to be accurate in areas such as medicine which have their own highly specialised terminology. The MedMatch algorithm was initially created to apply semantic matching in the medical domain through the use of a domain specific background resource. This paper compares a domain specific algorithm (MedMatch) against a generic (S-Match) matching technique, before considering if MedMatch can be tailored to work with other background resources. It is concluded that this is possible, raising the prospect of domain specific semantic matching in the future.

1 Introduction

Heterogeneity between data sources has been seen as a significant barrier to the exchange and use of information. Whilst ontologies have shown the potential to address some of these issues, they have often moved the problem up a level thus leading to heterogeneity between ontologies themselves. Semantic Matching has shown some promising results in terms of resolving ambiguities between ontologies [1]. Generic algorithms whilst appearing to have been applied successfully in fields with little domain specific terminology, often perform less accurately in areas such as medicine which have their own specialised terminology and use of language. MedMatch was initially created to apply semantic matching in the medical domain through the use of a domain specific background resource. This paper briefly demonstrates the potential benefits of domain specific semantic matching and seeks to understand whether the MedMatch approach can be extended to other domains which have similar attributes. A number of domains also require a specialised terminology, examples of these include Law, Physics and the Biological Sciences. The medical domain is therefore not unique in its requirement for semantics.

Currently the MedMatch algorithm has been implemented to use the UMLS as a domain specific background resource. In order to understand if it is indeed possible and if so what is required to change from one background resource to another it is necessary to select a second medical source. The Foundational Model of Anatomy ontology (FMA) was chosen for use as a test case. It was chosen because it appeared to have adequate coverage, semantics, granularity and metainformation and its model is very different from the UMLS itself. To replace the UMLS with the FMA is clearly a non-trivial task. This is primarily due to the fact that the structure of the FMA is different to the UMLS since they were designed for different purposes. The UMLS has been designed to function as a thesaurus for the domain, whereas the FMA is designed to model human anatomy. By considering the issues associated with changing background resources in this case, we aim to reach some initial conclusions regarding the expansion of MedMatch into a domain specific semantic matching framework in which new background resources can be harnessed.

The next section considers a number of terminology specific fields and draws conclusions regarding the emerging need for domain specific semantic matching in the future. Following this we briefly consider the related work in this area and then carry out a comparison between generic and domain specific semantic matching methods in order to determine the benefits that such approaches can deliver. The concluding sections describe a criteria by which background knowledge sources can be assessed and it is then used to evaluate the FMA. An analysis of the changes that would be needed to the MedMatch algorithm is carried out and final conclusions are reached.

2 Domain Specific Terminology and Related Work

Many fields require a specialised terminology, examples include Law, Physics and the Biological Sciences. Creating an ontology that covers the entire legal domain is a very challenging task. Some legal ontologies appear to be well developed. These however, are often “core” ontologies which cover the most important aspects of law such as the Legal Knowledge Interchange Format (LKIF) [2]. Such resources are mainly used as a basis for creating more specific ontologies such as ones related to criminal law or other sub-domains. A number of projects are currently working on legal ontologies, at present however none of them cover the legal domain sufficiently to provide a background resource for MedMatch.

The integration of semantics within Physics experiments is developing but is still in its infancy. Once again this leads to the conclusion that an appropriate background resource for Physics is not currently available. It appears likely that this situation will improve in the medium term as existing projects mature and make their results available. In the Biological Science domain there have been attempts to classify species into categories since the early days of the field. These have been done by using what are known as biological taxonomies, or taxa for short. There are many systems that these taxa follow, the most popular of

which is the Linnaean system [\[1\]](#). After some investigation it was discovered that some ontologies do exist for this domain, however, they are top-level ontologies such as BioTop [\[3\]](#) which are not yet sufficiently granular for the semantic matching process.

Domain specific ontologies are becoming more widely adopted and comprehensive. It is likely therefore that in the medium term a number of candidates for use as background knowledge with the algorithm will appear. The medical domain was one of the first to embrace the use of ontologies and as such, may perhaps be seen as an early indicator of how resources will develop in other domains over the next few years. It would appear likely that UMLS-like resources may well be developed. This would suggest that the need for domain specific semantic matching techniques will grow in the near future.

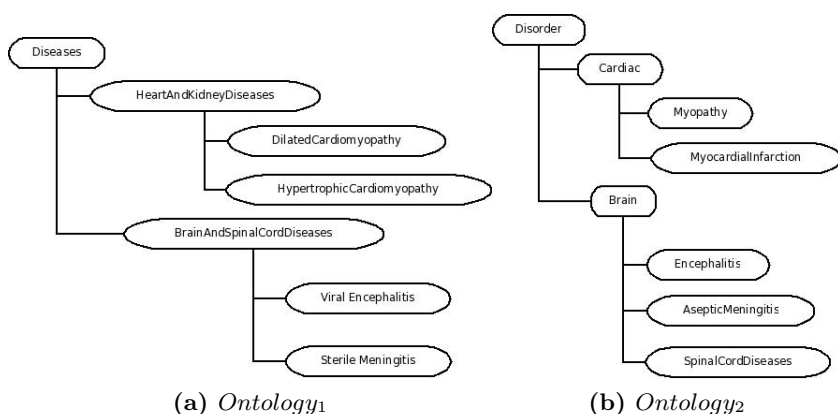


Fig. 1. Example Ontologies

There are many ontology alignment techniques available today [\[4\]](#) and research has been active in this area for quite some time mostly due to the emergent semantic web [\[5\]](#). These techniques range from simple string comparisons [\[6\]](#) to complex machine learning approaches [\[7\]](#). There have been some approaches that have used background knowledge as a dictionary [\[8\]](#) to more expressive techniques, for example text mining [\[9\]](#) or search engine distances [\[10\]](#). The majority of these systems output an equivalence relationship between feature pairs with a confidence value. The focus of this work is on a subset of the ontology alignment problem known as *Semantic Matching* [\[11\]](#) where relationships besides equivalence are discovered between concepts in two ontologies. For example matching the concepts “BrainAndSpinalCordDiseases” from *Ontology₁* in figure [1a](#) to “Brain” from *Ontology₂* from figure [1b](#) using a standard string matching approach the result of this match would be an equivalence relationship with a confidence value. In the case of semantic matching, however, the result would be that “BrainAndSpinalCordDiseases” is more general than “Brain”. The purpose of the MedMatch system was to show a method of conducting semantic

¹ http://en.wikipedia.org/wiki/Linnaean_taxonomy

matching in the medical domain. The following section compares the domain specific MedMatch algorithm against the generic SMatch system in order for us to better understand the potential benefit that using domain specific background resources can deliver.

3 SMatch Comparison

A freely available version of SMatch² was compared with MedMatch using a set of trees which were chosen to examine some common types of alignment contexts. An automated method was used to create reference alignments. This was validated by a clinician in order to confirm that the references were accurate. It should be noted that whilst this version of SMatch was able to discover disjoint correspondences between concepts. Such mappings have been omitted from the final matching results as MedMatch presently does not discover such relationships. Table 1 shows the results of the original SMatch algorithm in comparison to the results from each of the matching tasks that were produced by MedMatch. The new algorithm performed *better* overall than the original SMatch algorithm. This was demonstrated by the higher values for f-measure in all of the tasks. Its precision was also higher than the SMatch algorithm in all of the matching tasks.

Table 1. SMatch Results Compared To MedMatch

Algorithm	Expected	Overlap	Total Found	Precision	Recall	F-Measure
Test 1 – Similar Subdomains						
MedMatch	757	703	779	0.9	0.93	0.92
SMatch	757	562	763	0.736	0.742	0.739
Test 2 – Structure						
MedMatch	406	406	433	0.94	1	0.97
SMatch	406	404	532	0.759	0.995	0.861
Test 3 – Granularity						
MedMatch	1069	1003	1075	0.933	0.938	0.936
SMatch	1069	858	1149	0.746	0.802	0.773

In relation to the first test, which was the Similar Subdomains test, the SMatch algorithm performed significantly worse than the new algorithm. The values for precision were 0.736, 0.742 recall and 0.739 f-measure. The modified algorithm produced a 0.9 precision, 0.93 recall and 0.92 f-measure. In the second experiment which was the Structure test, the new algorithm obtained a much higher value for f-measure (0.97) than the original SMatch (0.733). The reason for this is that the current algorithm scored higher values for precision 0.94, in comparison to 0.759 and recall was also slightly higher at 1.0, when compared to 0.995. The third test, looked at the effect that two trees with different levels of granularity had on the algorithm. SMatch performed worse in terms of f-measure than the modified algorithm once again. The f-measure value for SMatch was 0.773 and this work achieved a f-measure of 0.936. This difference in f-measure is

² <http://semanticmatching.org/s-match.html>

mainly because the SMatch algorithm achieved lower precision (0.746 vs 0.933) and lower recall (0.802 vs 0.938).

This experiment has indicated that MedMatch can outperform the original SMatch algorithm when matching trees which contain medical terminology. This is because MedMatch had consistently higher f-measure values than SMatch. This metric is widely seen as a good indicator of the relative performance of alignment techniques, since it considers both the metrics of precision and recall. The significance of these results is that it shows that when matching medical trees containing complex structures and terms, this version of the algorithm performs better than the SMatch algorithm. It is also of interest to note that this version of the algorithm only has a single means of matching atomic formulae, whereas the SMatch algorithm has more than just a WordNet matcher. This confirms the need for domain specific semantic matching. The following section considers what is required in terms of a background resource by the MedMatch algorithm.

4 Background Knowledge Requirements

In order for a background resource to be useable as a source of knowledge by the MedMatch algorithm, it should address the criteria of *coverage*, *semantics*, *granularity* and *metainformation* to a satisfactory degree. This means that the background resource should contain a class hierarchy from which semantics can be extracted and synonym terms to feed the semantic matching process. It is also difficult for the algorithm to adapt to sources which have a sparse number of classes since they are often incomplete and do not always contain sufficient *coverage* for the matching of ontologies in a particular domain. This means that the class hierarchy needs to cover as much of the domain of the input ontologies as possible. The class hierarchy that is present must contain *semantics* between the concepts in the class hierarchy. These need to be at the very least “broader than” and “narrower than” relationships between concepts in the chosen resource. This is so that subsumption relationships can be present between anchor points for the algorithm. Synonym relationships also need to be present so that equivalence can be determined between concepts from the input ontologies.

Granularity relates to the level of detail that the chosen background resource contains. This requirement is again tied to the number of concepts present in the background resource. There have to be anchor points present for at least some of the concepts from the input ontologies so that a relationship can exist for the final reasoning process. The more information there is present from a trusted source of background knowledge the more relationships can be induced by the final reasoning process. For the criteria of *metainformation* there must be a string-to-concept mapping for the anchoring process to be successful. This means that labels of concepts in the input ontologies must exist in the source of background knowledge, or alternatively there must be a mapping between labels which describe the input concepts and the representation of the background knowledge source, such as is the case with the UMLS.

5 FMA Evaluation

The purpose of this section is to understand what is necessary in order for other domain specific ontologies to be “plugged in” to the algorithm as background resources. This is seen as an initial step in the future generalisation of the algorithm so that it can perhaps perform semantic matching in domains other than medicine. The FMA will be assessed using the criteria that was previously described.

- **Coverage.** The purpose of the FMA is to represent the anatomy of human beings. If only anatomical ontologies were to be matched, the coverage of the FMA is reasonably good. This is because the inputs to the algorithm are trees which cover the domain of anatomy. The FMA is considered to be one of the best resources that deals with the domain of anatomy. It is a very large and complete ontology describing the domain of anatomy.
- **Semantics.** In the area of semantics the FMA is very expressive. It is constructed in F-Logic (Frames) and therefore it supports classes, properties and instances. The design of the FMA is based on the lateral position of parts in the human body. For example, “left arm” and “right arm” are different concepts which are both subclasses of the concept arm. The FMA also contains some synonyms for concepts with the “synonym” attribute containing alternate concept names. The FMA only contains a single source of hierarchical information for the semantic matching process.
- **Granularity.** The FMA is highly granular with a high depth level for the anatomical concepts it describes. The purpose of the FMA is to describe human anatomy in great detail, therefore, it has many concepts to a high depth level.
- **Meta-information.** The FMA is created in Frames using the Protege tool. Protege provides an API which provides access to the different features of the FMA such as synonyms within concepts and the FMA hierarchy.

6 Algorithm Changes Required for Use with the FMA

The MedMatch algorithm has four major steps. These are 1) String to formula conversion, 2) Context creation and Filtering 3) Atomic formula matching and 4) Reasoning. The model of the FMA is different from the UMLS, the FMA contains more specific relationships between concepts where as within the UMLS relationships taken from a source vocabulary are abstracted into higher level relationships such as “PAR” for a general parent relationship. Changes need to be made to the algorithm to accommodate the new information that the FMA provides. This includes how concepts are organised within the FMA model such as how to extract synonyms and how to extract hierarchical information from the FMA itself. In this section the changes that are necessitated by the use of a different resource will be discussed as well as how these can be addressed. The fourth and final reasoning step requires no change since the axiom creation and reasoning scheme is identical across resources and therefore it shall not be discussed further.

6.1 Step 1 - String to Formula Conversion

The sub-steps for this part of the algorithm remain unchanged. There are multi-word concepts present in the UMLS as well, therefore the rule used in the first step for preferring multi-word concepts applies in this case as well. One important note on the anchoring scheme is that when searching through the FMA terms the synonym fields of concepts as well as the “Non English-Equivalent” field should be taken into account. This field contains the Latin equivalent of common anatomical terms such as Encephalon, which is a common synonym for Brain.

6.2 Step 2 - Context Creation and Filtering

Context is given to a node by using the background resource. This context was achieved by taking the logical formulae from the previous step then taking a conjunction from the formula of the current node to all the formulae leading to the root node. When using the FMA the context creation process is nearly identical i.e. the conjunction from the current node is still taken to its parent nodes. Where there are only single string to concept relationships present, a new filtering algorithm is required which can take into account the predicates present in the FMA to provide concepts with a more precise context.

6.3 Step 3 - Atomic Formula Matching

The UMLS hierarchies were used to match concepts attached to atomic formulae. The basic principle remains the same for using the FMA as a background resource. Semantics present in the background resource are used for the semantic matching process since these are mapped onto their propositional equivalents for the final reasoning process. These initial relationships form the background theory to seed the reasoning process from which other relationships not present in a background resource can be extracted. The rules for matching concepts attached to atomic formulae for both the UMLS hierarchies and the FMA are similar. The semantics present in both these resources are used for the creation of these rules. This would suggest that MedMatch could be modified to use the FMA as its background resource will relatively little changes being necessary to the algorithm itself. We can therefore conclude that MedMatch may well be generalisable and could be used in the future to create domain specific semantic matching systems.

7 Conclusion

MedMatch was initially created to apply semantic matching in the medical domain through the use of a domain specific background resource. This paper has shown that domain specific semantic matching can be beneficial in fields which have a specialised terminology. It has also been demonstrated that the MedMatch approach is capable of being extended to other domains which have similar attributes. These include Law, Physics and the Biological Sciences. At

present the MedMatch algorithm has been implemented to use the UMLS as a domain specific background resource. In order to understand if it was possible and what was required to change from one background resource to another the Foundational Model of Anatomy ontology (FMA) was chosen for use as a test case. It was selected because it appeared to have adequate coverage, semantics, coverage and meta-information and its model is very different from the UMLS itself. The rules for matching concepts attached to atomic formulae for both the UMLS hierarchies and the FMA were found to be very similar. The semantics present in both of these resources was used to creation of these rules. This suggests that MedMatch can be modified to use the FMA as its background resource with relatively little changes being necessary to the algorithm itself. We can therefore conclude that MedMatch may well be generalisable and could be used in the future to create domain specific semantic matching systems.

References

1. Giunchiglia, F., Shvaiko, P.: Semantic Matching. *The Knowledge Engineering Review* 18(3), 265–280 (2003)
2. Sartor, G., Casanovas, P., Casellas, N., Rubino, R.: Computable models of the law and ICT: State of the art and trends in european research. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) *Computable Models of the Law*. LNCS (LNAI), vol. 4884, pp. 1–20. Springer, Heidelberg (2008)
3. Beisswanger, E., et al.: BioTop: An upper domain ontology for the life sciences: A description of its current structure, contents and interfaces to OBO ontologies. *Appl. Ontol.* 3, 205–212 (2008)
4. Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer, Heidelberg (2007)
5. Berners-Lee, T., et al.: The Semantic Web. *Scientific American* 284(5), 35–43 (2001)
6. Stoilos, G., Stamou, G., Kollias, S.D.: A String Metric for Ontology Alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) *ISWC 2005*. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005)
7. Doan, A., et al.: *Ontology Matching: A Machine Learning Approach*. In: *Handbook on Ontologies in Information Systems*, pp. 385–403 (2004)
8. Madhavan, J., et al.: Generic schema matching using cupid. In: *27th International Conference on VLDB*, pp. 49–58 (2001)
9. Tan, H., Jakonienė, V., Lambrix, P., Aberg, J., Shahmehri, N.: Alignment of Biomedical Ontologies Using Life Science Literature. In: Bremer, E.G., Hakenberg, J., Han, E.-H(S.), Berrar, D., Dubitzky, W. (eds.) *KDLL 2006*. LNCS (LNBI), vol. 3886, pp. 1–17. Springer, Heidelberg (2006)
10. Risto, G., et al.: Using Google Distance to Weight Approximate Ontology Matches. In: *BNACI 2007* (2007)
11. Giunchiglia, F., et al.: Semantic Matching: Algorithms and Implementation. *Journal on Data Semantics* 9, 1–38 (2007)

Application Identification of Semantic Web Techniques in KM Systems

Mohammad Reza Shahmoradi* and Babak Akhgar

Shiraz University, Shiraz, Iran
Re_shahmoradi@yahoo.com, B.Akhgar@shu.ac.uk

Abstract. Knowledge management (KM) has been recognized as one of the most critical factors for obtaining organizational invaluable competitive advantage (Antoniou & Harmelen, 2004). New advances in IT provide novel methods to manage organizational knowledge efficiently. In this way, organizations have developed various systems to create, collect, store, share and retrieve organizational knowledge in order to increase their efficiency and competitiveness. Currently, regard to KM systems must maintain mass amount of data from various systems all over the organization as well as organizational extended value chain. KM systems must be able to integrate structured and unstructured data coming from heterogeneous systems to have a precise management on knowledge. This integration will help to apply useful operations on data such as analyze, taxonomy, retrieve and apply logical inference in order to obtain new knowledge. Nevertheless, there is some limitation in achieving the objectives of KM due to limited ability for semantic integration. Thus, the traditional methods are not responsible for KM systems users needs anymore. So There is a growing need for new methods to be used. To overcome these limitations we need to find a way to express meaning of concepts, the area that semantic techniques can help. This techniques offer novel methods to represent meaning of concepts and applying logical operation to get new knowledge of that concepts. These techniques organize information in a machine-processable manner, which allow machines to communicate directly without human intervention. Therefore, using these emerging techniques in KM systems show us a promising future in managing knowledge. The first step on implementing such systems is application identification of semantic techniques in KM systems and recognizing that in which areas this techniques can help to solve all limitations of current KM systems. In this paper, we aim to identify limitation of KM systems and present a comprehensive view of how adding semantics to data can help to overcome.

Keywords: Knowledge management, semantic web, semantic knowledge management, KMS limitations.

1 Introduction

Semantic technology, for example the semantic web, as a significant advance in IT, has attracted much interest and has been applied in many areas (Peer, 2002; Berendt,

* Corresponding author.

Hotho, & Stumme, 2002). One very promising application area of the Semantic Web is KM. The main idea of Semantic Web as an extension of the current Web, is to provide information in a well-defined manner in order to enabling computers and people to work in cooperation (Bernes-Lee, Hendler, & Lassila, 2001). In this way, Semantic Web handles machine-processable information, which enables communication between machines without human intervention. Current technologies, for example the Internet and the World Wide Web (WWW) support a dynamic and unprecedented global information infrastructure for organizations to exchange needed information to work in cooperation. However, there is some problems, one of the major problems is the huge amount of information available and our limited capacity to process it (Cui et al., 2001), which has been mentioned as “information overload”. Another is the fact that most of information on the Web is designed for human consumption, so we cannot design a robot to browse the web, whereas the structure of the data would not be evident to it (Berners-Lee, 1998). Moreover, there is not a unique way to handle the syntax and semantics of data/information, leading to problems of interoperability. However, the main problems that arise of diversity of system, syntax or structure, but an important aspect of information/knowledge – meaning (semantic) – has not been properly addressed (J. Joo, S.M. Lee, 2009).

In the recent years, in the field of KM many tools have been developed, the traditional KM tools assumed a centralized knowledge repository and therefore are not suitable for today’s knowledge space that is highly distributed and ever changing. Particularly when we consider the KM as part of Enterprises extended value systems (i.e. Customer, Supplier, business environment etc).

2 Areas on the Semantic Web for KM

Akhgar (2001, 2008) define KM as “a process of creating and exploiting value added Learning Processes (i.e. knowledge) so that the knowledge becomes a key strategic resource of an organization based upon particular infrastructure and Enterprise

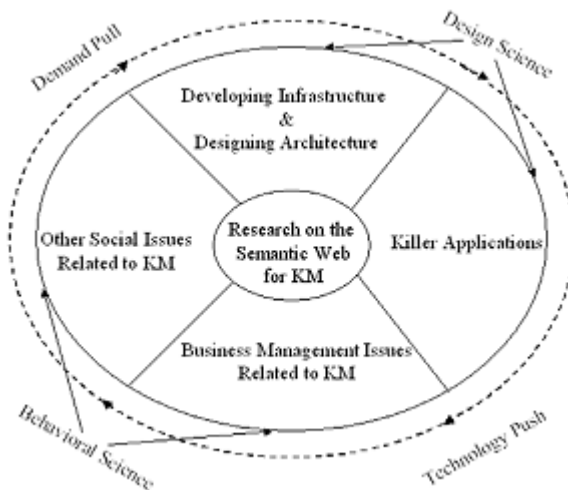


Fig. 1. Research area of the semantic web in KM- adapted of (J. Joo, S.M. Lee, 2009)

architecture, with measurable and quantifiable value". In line with this definition Joo and Lee 2009, classified KM research into four areas: architecture and developing infrastructure, business management issues, killer applications and other social issues. KM systems usually design and use, as specialized Information systems where various technologies – each of them is working with its own data format - are integrated. KM systems help organizations increase their effectiveness and competitiveness. However, due to limited ability for semantic integration, there are some limitations in achieving the objectives of KM. In this study, we will focus on necessity of applying semantic web techniques on KM systems and the areas of KM that the semantic web can help. In this way, we have identified the limitations of current KM systems so in the rest of paper we will discuss how to overcome these limitations.

3 Semantic Web

The principal aim is to enable software agents to exploit the contents of textual resources present on the Web so that users can ultimately be relieved of certain information searching and combination tasks (Fensel, Wahlster, Lieberman, & Hendler, 2003). The developed technologies apply as much to the Web as a whole as to organizational semantic web in particular. Current organizational semantic webs architectures rely on the coupling of a collection of textual resources with formal resources, the latter also being qualified as "semantic" resources. Of these, one can distinguish annotations of textual resources or "metadata" (which express knowledge about textual resources) (Handschuh & Staab, 2003) on one hand, and ontologies (which stipulate the meaning of the terms used to express the textual resources and the metadata) (Davies, Fensel, & van Harmelen, 2003; Abecker & van Elst, 2004) on the other hand. Again, one finds a distinction between knowledge and meaning. In terms of the contribution of these semantic resources, various approaches are being explored by academia and industrial foundations such as W3C.

4 The Semantic Web as Innovation Tool for KM

In first step, we must understand the complexity of technology, which is a critical factor to gain business value.

In order to have comprehensive languages, W3C developed RDF as standard for metadata and adopted OWL as a recommended standard for knowledge representation. Recently semantic web technologies and its abilities in representing knowledge have taken many attentions in scientific societies. On-to-knowledge is a project of information society technologies (IST) where various tools including OntoBroker, OntoEdit, Sesame, and OntoShare were developed. The findings of the empirical analysis by J. Joo, S.M. Lee (2009) indicate that the limitation factors of a KM system are related to system quality and knowledge quality. Where the limitation factors of system quality are mainly related to the technology itself while the limitation factors of knowledge quality are related to people and culture (Benbya et al., 2004). In this section, we discuss how the Semantic Web support KM processes and how a KM system based on the Semantic Web offers an opportunity to overcome technical limitations of the current KM systems.

The Semantic Web represents novel methods to overcome the barriers to knowledge retrieval in the current KM system. All resources in the Semantic Web are represented in RDF (Resource Description Framework). Using this methods makes it possible for users to query and get answers as if they are using DBMS, also some services has been developed to converting other knowledge representation methods to RDF without human intervention such services makes using semantic techniques more easier.

The Semantic Web also supports RDFS and ontology, which provides formal description of entities and enables semantic analysis on vocabularies contained in query and domains as well as syntactic analysis. Thus, the Semantic Web can provide accurate knowledge suitable to users. The Semantic Web also can offer context-aware knowledge to users because ontology languages, such as OWL, support reasoning functions and domain knowledge. The inference function and context-aware capability can help KM system to enhance the ability to search knowledge suitable for users. We can represent all resources in RDF, Internal or external documents of organizations or even a concept within a web page or a web resource can be expressed as a RDF statement. A resource of RDF, a knowledge object, can be searched with an independent knowledge unit as a user searches a document in document management systems. Moreover, a specific part or sentence of a Web page or a part of a document may be represented as a knowledge object, so we can break the knowledge object into small pieces, this capability is very important to find accurate and correlate results as we can search for a knowledge unit rather than document unit.

5 Integration Limitation

Only a few studies have focused on the interoperability problem of knowledge sharing. Heterogeneous knowledge sources – for example, heterogeneous product catalog/order/knowledge repositories – may have various data formats, including those of reports extracted from relational or object databases, HTML pages, XML files or possibly (RDF) files. There are three types of integration: data, application and process (Giachetti, 2004). Due to different levels of applications heterogeneity such as using different syntax or semantic, organizations have problems in achieving interoperability between various systems. In this way, they want to integrate different systems and applications as if they work with a single system. The goal of data integration is data sharing where different systems exchange data with each other. The goal of application integration is to achieve interoperability between systems (J. Joo, S.M. Lee, 2009). Until recently, the traditional approaches for providing interoperability include standardization and middleware or mediators as well as enterprise application integration (EAI). The traditional integration approaches easily integrate structured data, which extracted from heterogeneous databases, but they have problems when integrating unstructured data or knowledge from sources such as HTML, spreadsheet or diverse format of documents. For instance, the traditional approaches for integration, do not play the role of a content integrator, which automatically extracts related knowledge from different sources and aggregates this knowledge. There are some sort of applications such as EKP, that just integrates different applications and offers one access point for users.

Fig.2 shows a comparison of traditional and semantic integration.

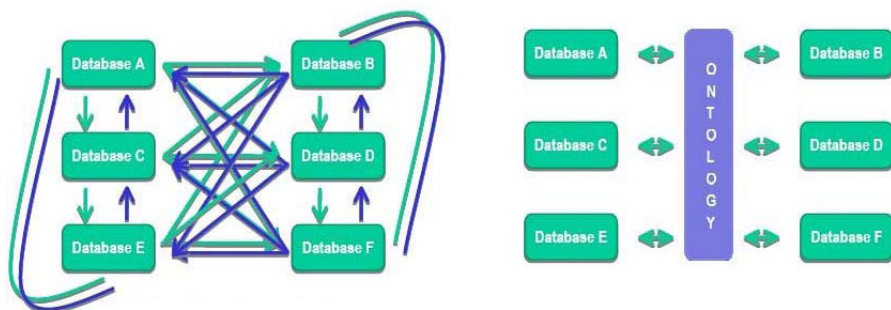


Fig. 2. Comparison of semantic and traditional integration

Although using middleware's as traditional approach provide syntactic and structural integration at the data and application level, it due to lack of semantic, cannot enable semantic integration. In other words, the traditional approach allows users to share data between different systems and provides interoperability between systems by exploiting the mediator. The traditional approach in the worst case, as shown in Fig.2, needs $n*(n - 1)$ mediators for mapping and translating between systems, these mediators can provide syntactic integration, so applications can talk to each other but most of the problems are due to lack of semantic integration, which mediators are not be able to solve them. Software agents which is applied as a mediator cannot understand terms represented in different systems or process them without human intervention.

In the Semantic Web approach, the software agents understand the meanings of the terms and automatically process them by exploiting the RDF and ontologies. In this way, we can enables semantic agents to recognize the meaning of the words and to take proper action encountering synonymous or a word with different meanings in various systems.

Since W3C adopted XML as the Web document standard, XML is widely used as a baseline for semantic web's technologies. The Semantic Web, RDF, RDFS, and OWL follow XML-based syntax (Antoniou & Harmelen, 2004). In the Semantic Web, a software agent can access heterogeneous systems and provide knowledge and information suitable for users (Beydoun, Kultchitsky, & Manasseh, 2007).

6 Improving the Knowledge Creation Cycle

Knowledge creation cycle according to the SECI model suggested by Nonaka and Konno (1998) has four steps, socialization, externalization, combination, and internalization. Let us discuss how the semantic web improve the process of the SECI model. Nonaka & Konno, (1998) state that, socialization involves the sharing of tacit knowledge among individuals; socialization is activated by exchanging of tacit knowledge through joint activities. The KM systems that uses Semantic Web technologies, facilitates exchange of tacit knowledge for an online or virtual community or knowledge network by providing capabilities of semantics and integration.

According to SECI model, externalization is said to conversing tacit into explicit knowledge. Obviously, externalization requires the expression of tacit knowledge. Semantic Web technologies including RDF, RDFS, and ontology provide a novel approach for knowledge representation as well as reasoning for the KM system, that allow us to extract new knowledge from existing knowledge. This techniques for knowledge representation and reasoning, facilitates externalization of knowledge creation.

Combination of the SECI model indicates combining explicit knowledge in order to gain new and more complex sets of explicit knowledge. In this step, we must be able to capture, edit, process or disseminate, and integrate explicit knowledge. To illustrate how the semantic web can help combination of knowledge, we can consider blogs as an example of explicit knowledge on the Web. Blogging is an activity for creating, capturing, editing, and integrating explicit knowledge from various sources. Bloggers easily and conveniently bring other blogs in their blog site and reply or add new knowledge to them. Cayzer (2004) discussed the blog as a killer application for the Semantic Web. Finally, internalization of the SECI model means the conversion of explicit knowledge into tacit knowledge; internalization requires identification or search of relevant knowledge as well as a learning and training program. When we work with large amount of information, finding relevant information become a problem and soon the information would be unmanageable. As we know the information overload is one of the problems using traditional KM systems. The KM system that uses Semantic Web techniques has the ability to resolve the information overload problem resulting from the traditional keyword search methods. Users of the Semantic Web-driven KM system can find more relevant and accurate knowledge in Web resources and improve their learning effect and knowledge quality.

7 Sharing Knowledge

In order to achieve distributed Knowledge management, one of the critical factors is sharing knowledge among organizations. In this way, Interoperability among organizations with heterogeneous knowledge sources becomes a research focus in the field of knowledge management. Specifically, sharing knowledge among stakeholders in a supply chain is crucial. However, only a few studies have addressed the problem of interoperability and knowledge sharing in supply chains. Current technologies, such as EDI, RosettaNet or the current Web, have been designed to share data/information, rather than knowledge (Huang and Hua Lin, 2010).

Nowadays, to prevent the bullwhip effect in a supply chain, an efficient information/knowledge sharing approach is more important than ever. (Devanna & Tichy, 1990; Dove, 1994, 1995). Woodworth and Kojima (2002), Archer and Wang (2002), Malone (2002), Zhuge (2002) and Nabuco et al. (2001), have discussed information/knowledge sharing and its solution extensively. When knowledge workers interact with unfamiliar knowledge sources that have been independently created and maintained, they must make a non-trivial cognitive effort to understand the information contained in the sources. Moreover, the information overload is another problem, which makes achieving semantic interoperability between a knowledge source and a knowledge receiver more critical than ever. (Chun-Che Huang, Shian-Hua Lin, 2010).

All of entities that play a role in a supply chain can improve their productive coordination, if they share information about their organizational situation. In a typical industry that has been built by several actors (firms, associations, workers, etc.) or in a supply chain it is extremely important to have a knowledge platform on which to share business strategies, technological trends, opportunities, forecasts, customer care, etc. This information sharing can be crucial to whole network as an important performance factor.

Numerous schema-level specifications (such as DTD or XSD) have recently been proposed as standards for application. Although such schema-level specifications can be used successfully to specify an agreed set of labels with which information can be exchanged, but due to lack of semantics, the current Web technology cannot be assumed to solve all problems of semantic heterogeneity (Cui et al., 2001). Huang and Hua Lin (2010) state, there is four interoperability problems in knowledge sharing: (i) there is too many schema-level specifications but they do not use the same terminology. (ii) Bussler (2002) showed that because of the high complexity of the interactions of B2B protocols, B2B integration through programming does not scale,. (iii) The current technologies, such as, EDI, do not explicitly link the semantic requirements to formal process models. In such situation, integration of SCM implementations will be infeasible (Huhns & Stephens, 2001). (iv) The problem of semantic heterogeneity still applies while all data are exchanged using XML, structured according to standard schema-level specifications, which means using schema level specification is not enough for semantic integration. Therefore, we need a solution that involves semantic technologies in a distributed knowledge management; such technologies have the potential to make a big revolution in the IT world. In the current heterogeneous information world, the semantic web enables a flexible and complete integration of applications and data sources, so it offers a holistic solution in order to have a integrated platform to share data and information in a scalable manner. Furthermore, it provides a well-defined structure, ontology, within which meta-knowledge can be applied (Fensel et al., 2000).

8 Semantic Annotation

Annotating is one of semantic web`s techniques to describe documents or entities within documents. An annotation (also referred to as “metadata”) is a document containing knowledge about another document. Using formal annotation (intended to be interpreted by machines) can be very useful to achieve semantic web`s goals.

Concerning the content of these annotations, there can be two levels of annotation, knowledge can relate either to the contents of an annotated document (e.g., by means of a collection of concepts or entities expressed in the document) or to a document itself (e.g., author, publication date, language used, etc).

The annotation process relates to the unstructured formats of distributed knowledge documents and allows knowledge workers to be accessed efficiently, such that these hetero-formatted or un-structured knowledge documents can be retrieved. (A. Kiryakov et al, 2009).

Usually huge amount of organizational knowledge has stored in unstructured texts. In order to extract knowledge from unstructured texts we need applications to be able to automatically analyze text, recognize entities and make relationships among them. Nevertheless, without semantic techniques we would not achieve such capability.

In this way, some applications have been developed; these applications can be very useful in extracting knowledge from unstructured documents. KIM (Knowledge and information management) is one of these applications.

The main focus of KIM was on providing the necessary infrastructure for automatic extraction of named entity references and descriptions, including attributes and relations, from text. We call both the process and the result “semantic annotation”. We hold Ontologies and background knowledge in a semantic repository and use it for analyzing text, to extract information.

After analyzing a text we will have three important result:

- Extension of the semantic repository with structured data, extracted from the text, which can be used to analyzing texts, more accurate.
- Generation of annotations (metadata) that link the text with the repository, which enables us to search and extract correlate results from diverse documents.
- Finding new attributes of an entity and relations.

For example, Fig.3. presents the interlinking between the text and the data; each annotation contains an identifier of entity, described in the semantic repository, as the dot dash lines show, named entities are recognized and relationships between ontologies (in KB) and the entities has been built. In addition to the Named entities, KIM is also be able to annotate and extract Key-phrases to describe document`s overall properties.

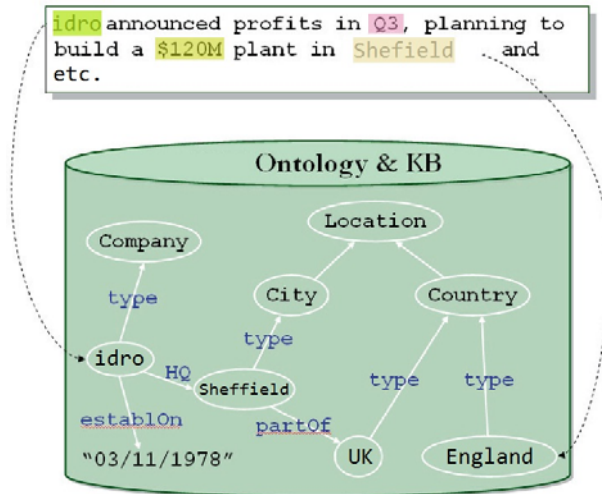


Fig. 3. Semantic annotation in KIM

KIM is designed to be able to take advantage of pre-existing structured data and inference in order to extract new knowledge and find new relations with other concepts— also the text-mining algorithms are initialized from a semantic repository, which contains huge amounts of instance data. An important part of the annotation process, is called identity resolution: to find whether a particular named entity reference denotes an already known named entity in the repository and which one.

In case the reference has not any match in the repository, a new entity description will be created and added in the ontology.

9 Input Data Validation

Electronic commerce sites such as E-banking and e-Purchasing have concluded that more than 92% of web applications have some weaknesses and are vulnerable to some form of attack. The Gartner study found that 75% of Internet assaults are targeted at the web application level.

Several reports have shown a sharp increase in the number of web-based attacks. For example, The Computer Emergency Response Team (CERT) has reported that, there has been a dramatic increase in the number of security vulnerabilities (weaknesses in a computing system) which threaten web sites and applications.

Technically, we can control some of Web browsers through registry, so these browsers can be attacked by simple viruses, on the other hand, access to databases through Web browsers is now common place with some form of web server being provided by all major vendors of database software.

Thus, our data can be in danger and hackers can access it simply. In addition, one of the most common methods to access a database is code injection, such as SQL injection and cross-site scripting that can cause unauthorized access to database. Most of these can be prevented by an accurate input data validation technique.

Most of security issues in applications are caused by inadequate input validation. If we use an exact validation rules, we will be able to check for correctness, meaningfulness, and security of data that are input to the web-based application or database.

An Input validation can be a critical for application security. Typically, a little attention is paid to it in a web development project. Now, it is estimated the web application vulnerabilities (such as XSS or SQL injection) for more than two thirds of the reported web security vulnerabilities, moreover, a hacker can use these vulnerability to attack to many web applications and web sites using methods like “Backdoor”, that it can be very dangerous not only to organizations web application but also to all applications that use the same host . Number of common data validation techniques has been proposed, but none of them represents a final solution to prevent all problems resulting from security vulnerabilities. Semantic web technologies also can be applied in this area of application development. S.Aljawarneh et al (2010) have proposed a new data validation service, which is based on semantic web Technologies, the proposed semantic architecture consists of five components: RDFa annotation for elements of web pages, interceptor, RDF extractor, RDF parser, and data validator. The experimental results of the pilot study of the service indicate that the proposed data validation service might provide a detection, and prevention of some web application attacks. Although the proposed solution is not complete, and still have some

issues, but it shows that, semantic technologies have the potential to adopt as a holistic solution in the near future.

10 Conclusion

In this paper, we have reviewed the traditional KM systems and its limitations, and described how semantic technologies can help these barriers to solve. Some of these limitations cause user dissatisfaction and inefficient use of knowledge. In this way, we have discussed, semantic approach can be make significant improvement in knowledge retrieval, sharing, taxonomy, content management, integration as well as improving in knowledge creation cycle.

The main conclusion of this paper is, Because of i) the nature of heterogeneous source of knowledge in organizations; and ii) Today`s knowledge space that is highly distributed and ever changing; current KM systems is not be able to manage organizational knowledge properly and we need to apply semantic approach in developing next generation of KM systems.

References

1. Akhgar, B., Siddiqi, J., Hafeez, K., Stevenson, J.: A conceptual architecture for e-knowledge Management. In: ICMLA 2002 Conference, CSREA Press, USA (2002)
2. Joo, J., Lee, S.M.: Adoption of the Semantic Web for overcoming technical limitations of knowledge management systems. *Expert Systems with Applications* 36, 7318–7327 (2008)
3. Huang, C.-C., Lin, S.-H.: Sharing knowledge in a supply chain using the semantic web. *Expert Systems with Applications* 37, 3145–3161 (2010)
4. Chen, M.-Y., Chu, H.-C., Chen, Y.-M.: Developing a semantic-enable information retrieval mechanism. *Expert Systems with Applications* 37, 322–340 (2010)
5. Cayzer, S.: Semantic Blogging and Decentralized Knowledge Management system. *Communication of the ACM* 47(12) (2004)
6. Aljawarneh, S., Alkhateeb, F., Maghayreh, E.A.: A Semantic Data Validation Service for Web Applications. *Journal of Theoretical and Applied Electronic Commerce Research*, Electronic version 5, 39–55 (2010)
7. Ma, Z., Wang, H., (eds.) The semantic web for knowledge and data management. *International Journal of Information Management* 29, 420–422 (2009)
8. Qu, Z., Ren, Z.: The Frame of Enterprise Knowledge Management Model Based on Semantic Web. In: ICSP 2008 Proceedings (2008)
9. Sereno, B., Uren, V., et al.: Semantic Annotation Support in the Absence of Consensus. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 357–371. Springer, Heidelberg (2004)
10. Bernes-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* 284(5), 34–43 (2001)
11. Pollock, J.T.: *Semantic web for dummies*. Wiley Publishing, Inc., Indianapolis (2009)
12. Beydoun, G., Kultchitsky, R.R., Manasseh, G.: Evolving semantic web with social navigation. *Expert Systems with Applications* 32, 265–276 (2007)

13. Damodaran, L., Olphert, W.: Barriers and facilitators to the use of knowledge management systems. *Behaviour and Information Technology* 19, 405–413 (2000)
14. Kiryakov, A., Popov, B.: Semantic annotation, indexing, and retrieval. *Science, Services and Agents on the World Wide Web* 2, 49–79 (2004)
15. Kiryakov, A., Popov, B., Kitchukov, I., Angelov, K.: *Shared Ontology for Knowledge Management*. Semantic Knowledge Management (2009)

Aligning the Teaching of FCA with Existing Module Learning Outcomes

Simon Andrews

Conceptual Structures Research Group,
Communication and Computing Research Centre,
Sheffield Hallam University, Sheffield, UK
s.andrews@shu.ac.uk

Abstract. Careful design of teaching and assessment activities is required to properly align a topic to the intended learning outcomes of a module. This paper describes and evaluates a four year project to align the teaching of FCA with the learning outcomes of a final-year undergraduate *Smart Applications* module at Sheffield Hallam University. Biggs' constructive alignment, incorporating an adapted version of Yin's case study research method, was used in an iterative process to analyse and modify teaching and assessment activities.

1 Introduction

Formal Concept Analysis (FCA) [5] is a valuable subject to study as part of many Degree courses; it has applications in biological sciences, music, linguistics, data mining, semantic searching and in many other area. Its mathematical basis, visualisation and wide scope for software development make it a suitable problem domain in a variety of disciplines.

At Sheffield Hallam University, a project was undertaken to introduce FCA as a topic to an existing undergraduate computing module called *Smart Applications*. It was felt that that the applications of FCA, particularly in semantic search and knowledge organisation, made it an interesting subject for the module. To monitor the success of its introduction and make modifications where it was found to be not properly aligned to the existing learning outcomes, an iterative approach was taken, applying the *constructive alignment* model of Biggs [2].

2 Biggs Constructive Alignment

The purpose of Biggs' constructive alignment is to design learning activities and assessment tasks so that they are aligned with the learning outcomes that are intended (Figure 1). The method includes modification of learning activities based on the outcomes of assessment. It is essential that the learning outcomes are assessed and a proven way of doing this is by criteria-based assessment where grades are awarded according to how well students meet the intended learning outcomes [3]. The problem with the introduction of a new topic into an existing

curriculum is that the intended learning outcomes of the module will probably have been designed without the new topic in mind. A means is required of testing the alignment of the new topic with the existing scheme. Central to Biggs' is the notion that students construct their own meaning from their learning activities. A means of accessing these constructed meanings could, therefore, be used to ascertain the extent to which particular learning outcomes have been met.

The author's familiarity with research methods led to the idea of using a research method to identify the extent to which existing Smart Applications learning outcomes were being met by FCA. The common practice of using case studies as assignments for the Smart Applications module, combined with the idea that modifications would play a key role in the alignment process, suggested Yin's case study research method [6].

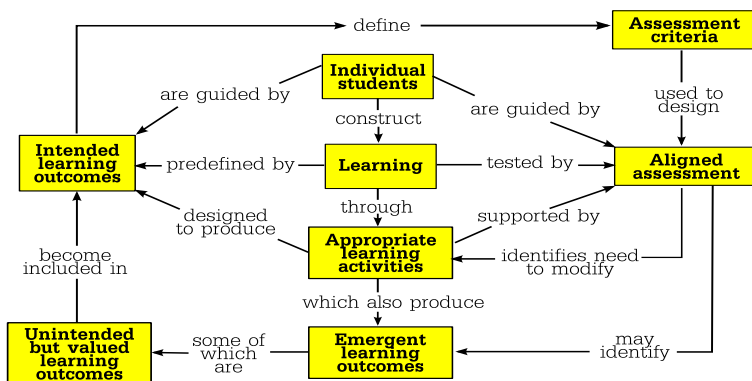


Fig. 1. Biggs constructive alignment: source HEA academy [4]

3 Yin's Case Study Research Method

Yin describes a method whereby theory is modified by conclusions drawn from a series of case studies (Figure 2). A theory is investigated by carrying out several distinct case studies. Conclusions regarding the veracity of the theory are made stronger by the fact that a single experiment is not relied on; by putting the theory to the test in different ways, corroborating cross-case conclusions can be made. In the approach used for aligning FCA with the Smart Applications module, the link between Yin and Biggs is made by considering the *learning activities* in Biggs as the *theory* in Yin. The case studies used to test and modify the theory are thus the assessment activities of the module. Yin's method adapted for use in Biggs becomes that in Figure 3.

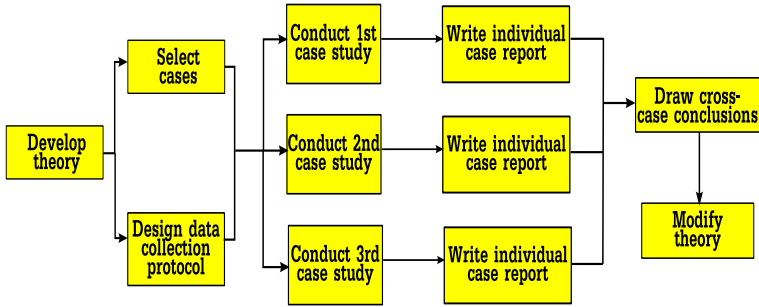


Fig. 2. Yin's case study method

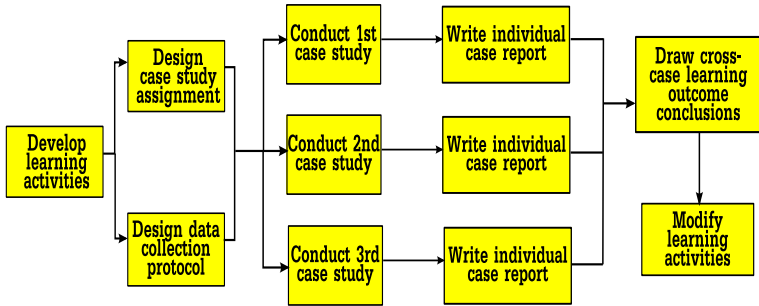


Fig. 3. Yin's case study method adapted for Biggs

3.1 Data Collection

To conduct Yin, a data collection protocol is required that allows the testing of the theory. For the purpose of using Biggs, a data collection protocol was required that would allow cross-case conclusions to be made regarding the outcomes of learning activities; how well have the students achieved the intended learning outcomes of the module? Because students' construction of their own meaning is so central to Biggs, it seemed sensible to use students' own conclusions from their case study assignments as the data source. The content of coursework clearly provides qualitative evidence of learning and, by taking only the concluding sections of coursework, appropriate data in a manageable quantity could be procured. Analysis of the students' conclusions also provided the mechanism in Biggs whereby emergent learning outcomes could be identified.

A quantitative measure was also required to allow comparison with other modules. Assessment marks seemed a sensible choice as this would also provide an indication as to the depth to which learning outcomes had been achieved. In conjunction with a measure of what has been learned, marks can tell us how well something has been learned. The incorporation of the method into Biggs can be seen in Figure 4.

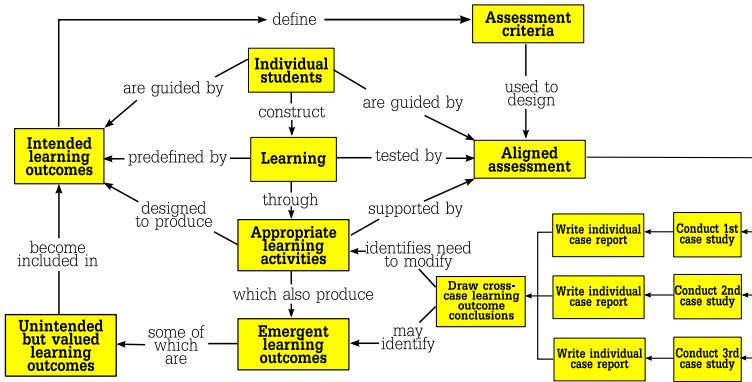


Fig. 4. Yin's adapted case study method incorporated into Biggs constructive alignment

4 Applying the Method: The Four Cycle Smart Applications Project

Smart Applications is a final year undergraduate computing module taught at Sheffield Hallam University. One of the aims of the module is to introduce students to frameworks and techniques for representing and reasoning with knowledge for smart applications. It was felt that FCA was appropriate to study to achieve this aim. However, it would have to be aligned with the existing intended learning outcomes (ILOs). It was decided to use this as an opportunity to develop and test the pedagogical method, described above. The process was carried out in four cycles, over the academic years 2007/8, 2008/9, 2009/10 and 2010/11. The existing ILOs were:

- ILO1.** Describe the notion of representing and reasoning with knowledge for smart applications.
- ILO2.** Draw on one or more frameworks and techniques for representing and reasoning with knowledge for smart applications.
- ILO3.** Critically evaluate the key issues in knowledge representation and knowledge sharing for smart applications.
- ILO4.** Identify the practical use of software tools for developing smart applications.

4.1 Cycle 1: 2007/8

The learning activities associated with FCA in the 2007/8 delivery of the Smart Applications module were primarily concerned with the mathematical underpinnings of FCA. After an introduction to FCA, lectures and tutorials were based on the themes of *Knowledge Architectures for Smart Applications through Conceptual Structures* and *Specifying Smart Applications using FCA*. The assignment was a case study for managing user profiles in a business information system. Variation was designed into the assignment by phrasing it openly; the

students were asked to investigate possible solutions and make their own recommendations. The deliverable was a report that considered “How might the implementation of a ‘user profile concept lattice’ be accomplished, that supports the capture and reconfiguration of user profiles?” The following are quotes from the conclusions of the students’ case study reports that gave an indication of the learning outcomes achieved:

1. “FCA lends itself well to mapping user profile details. This is best achieved using a series of different lattices covering the different aspects of user profiles. Trying to capture all of the relevant user information in one lattice would probably be a little ambitious and also result in a complex and impractical lattice.”
2. “Through implementation of web mining techniques of Association rules or Conceptual Clustering Mining you can capture User Profiles.”
3. “By utilising BI techniques and FCA modeling, user profiles would be smart as they would show relevant information to users across the company.”
4. “I have found that using FCA for capturing User Profiles makes seeing the relationship between objects and attribute a lot easier, which we can then use to see relationships for that particular profile which helps gathering data for use in trends.”
5. “Through user profiling with FCA I found that content and information only relevant to that particular individual within the company can be displayed, saving time and improving productivity.”
6. “Although FCA seems to be a good way to analyse and create user profiles it can become slightly difficult if you have a model that constantly changes as it can be difficult to adapt a new role into the model.”
7. “Once a Smart Google system can comprehend the context of a sentence, through the identification of relations between words in a search phrase, it will then deliver the user with answers rather than results. This I would consider as smart.”
8. “Combining FCA and BI with the functions and facilities available in Web DynPro has the capacity to store a lot of information in a very well organised and formal structure, which can be added to or reduced very easily, with little adaptation of the data structure.”

The marks for the assignment were slightly disappointing with a mean of 54%. It was clear that some of the ILOs were being met, to some extent. Much of the learning seems to have been centered around the visualisation aspect of FCA, and thus relevant to ILO1, but only quote 1 suggests any depth to that learning. ILO2 appears to have been achieved in quotes 1 and 7 and perhaps in quote 2, but again without depth. In terms of evaluating issues (ILO3), displaying relevant information through FCA appears to be the main message, but with some contradictory understanding of how changes in information can be managed in FCA; quote 6 suggests this is a problem but quote 8 advocates FCA as being advantageous in this regard. Only quote 1 suggests that lattice complexity is a problem. For ILO4, only quote 8 mentions a software tool for developing smart applications. The learning appears to be about FCA itself rather than the practical application of FCA as a framework or technique. Little was learned about the issues involved in knowledge representation for smart applications or of the use of software tools for the development of smart applications. It was decided, therefore, to modify the learning activities of the module to focus less on the

mathematical theory of FCA and more on its practical application; perhaps by implementing FCA-based software, students' learning outcomes would be better aligned with the intended ones.

4.2 Cycle 2: 2008/9

Modifications were made to the learning activities of the previous cycle to focus less on the theoretical aspects of FCA and more on the engineering of FCA-based software. The themes were *Semantic Search: 'Sleuthing' with FCA*, *Data Structures for FCA* and *FCA-based Smart User Interfaces*. The assignment was a modified version of the 2007/8 assignment, with an element of application prototyping replacing the investigation of theory. The following are quotes from the case study reports:

1. "Integrating FCA user profiles into an SUI such as Dynpro comes with issues. Dynpro is not very flexible in terms of changing things by the program. It requires the user profiles to already be configured to support an FCA ontology."
2. "Concept lattices were used to structure ... an FCA-based user profile. This data was ... integrated into a Smart User Interface (SUI) by being focused on the organising, sorting and searching of data, and finding matching concepts."
3. "Using the unique way that FCA stores and represents knowledge the Smart User Interface can give the user multiple ways of finding information."
4. "We defined user sessions as only the URLs that have been requested and visited by the user. This considerably lowered the amount of attributes and data that had to be analysed."
5. "We have presented how Web Dynpro implements the MVC framework which allows FCA of the data used within it, integrated with BI to support user profiles, customising a user's interface, manipulating it to suit the aims of enterprise and user. Using FCA to discover patterns and correlations [was a] way of determining relationships and discovering the implicit ones that other methods struggle to uncover."
6. "It is possible to integrate the FCA user profiles successfully; however, there are several difficulties when implementing them: firstly there's the issue with accuracy, as the larger the company or department the more complicated it is going to be to implement. There's also a concern with timing, as it's a long process computing the profiles."
7. "The use of FCA based user Profiles as the basis for e-commerce recommender systems does bring benefits, but the same level of functionality can be achieved using [off-the-shelf] alternatives."

The marks for the assignment were poor, with a mean of 47%; down significantly on the previous year. However, it was clear that most of the ILOs were being met and in more depth. ILO1 was evidenced in many of the comments, as was ILO2. Key issues were highlighted more (ILO3) with some of the quotes indicating a good awareness of complexity and performance issues. Some of the advantages of FCA have been better explored and understood. Some quotes indicate that a broader understanding of the context has taken place. Less in evidence is ILO4; very few of the students reflected on the use of software tools for developing smart applications and the low mean mark shows that the overall

achievement was unsatisfactory. Although the quotes give an encouraging picture of the learning that has taken place, the technical aspects were a struggle. The programming skills of the students were not sufficient for developing a useful prototype application. It was therefore decided to modify the learning activities to require less of these technical skills and focus more on the investigation, use and development of existing FCA tools and applications.

4.3 Cycles 3 & 4: 2009/10 & 2010/11

For cycle 3 the learning activities were based on the themes of *FCA Tools*, *FCA ‘Sleuth’ Applications* and *Data Mining with FCA*. Practical sessions were designed to explore the capabilities and limitations of existing software. The assignment was changed by replacing the prototyping element with one that used existing tools and techniques to carry out FCA on real sets of user profile data. The following are quotes from the case study reports:

1. “FCA raises new questions regarding the actual warranty of ‘hidden’ information. How can we trust that a smart application is actually giving the right results?”
2. “To make this investigation even more interesting, we could use similar FCA techniques on the categorised attributes and incorporate the boolean attributes to discover if the top 4 factors are actually the factors of people from all age groups, for example.”
3. “Filtering the formal concepts generated allowed for visualisation of the large data set and allowed for affective analysis.”
4. “Although there may be some current issues with the interoperability of FCA in existing technologies ... FCA could be integral to the development of semantic knowledge architectures.”
5. “While formal concept analysis and the enabling technologies described are now in a usable state, the applications to use it now have to catch up with them.”
6. “Visualisation techniques are key to enabling Smart Applications. Information that would be hard to find otherwise was made clear to understand.”
7. “There seems to be a lack of ability to be able to communicate and exchange data between FCA systems and tools with non-FCA applications. Although the data mining is ‘smart’, problems arise when changes are needed to be made to lattices.”

The marks for the assignment had a mean of 67%. The quotes indicate that the ILOs have been met, and to a good extent. Notions of representing and reasoning with knowledge for smart applications (ILO1) and the drawing on FCA as a framework/technique for smart applications (ILO2) are apparent in many of the comments. There is good evidence that an ability to critically evaluate key issues has been demonstrated (ILO3), particularly in quotes 1, 3, 4 and 7. And there is a strong sense that the students can identify the practical use of software tools for developing smart applications (ILO4).

In cycle 4, therefore, only minor modifications to the activities were made. The assignment this time centered on carrying out FCA on a number of public data sets. The results were encouraging with a mean mark of 64% and many of the coursework conclusions indicated that the ILOs had been met to a good degree.

Table 1. Summary of *Smart Applications* Results

cycle	mean mark %	course mean %
1	54	63
2	47	59
3	67	64
4	64	62

5 Conclusion

This four-year Smart Applications project shows that FCA can be aligned with the intended learning outcomes of a suitable existing module using sound pedagogical practice. In Yin's adapted case-study method, students drew conclusions on the work they had undertaken, then, from these, teachers drew cross-case conclusions regarding the learning outcomes achieved and how far they were aligned with the intended ones. The students' concluding remarks gave a qualitative measure of their learning whilst their marks gave a quantitative measure of the depth of their learning. The results of all four cycles are summarised in Table 1. The mean mark of for the course as a whole is given to corroborate the alignment.

An improvement to the method may be found by designing the assessment activities in such a way that the marks can be assigned to ILOs, so that alignment with individual ILOs can be quantified.

Acknowledgment. This paper is a revised and updated version of one presented at CS-LTA 2010 [1].

References

1. Andrews, S.: Aligning the Topic of FCA with Existing Module Learning Outcomes. In: Polovina, S., et al. (eds.) Artificial Intelligence Workshops: 1st Conceptual Structures - Learning, Teaching and Assessment Workshop at ICCS 2010, Kuching, Malaysia (2010)
2. Biggs, J.: Teaching for Quality Learning at University. SRHE and Open University Press, Buckingham (1999)
3. Biggs, J.: Aligning Teaching and Assessment to Curriculum Objectives (maginative Curriculum Project, LTSN Generic Centre (2003)
4. HEA Academy, Learning and Teaching Theory Guide, <http://www.engsc.ac.uk/learning-and-teaching-theory-guide/constructive-alignment>
5. Wille, R.: Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis: Foundations and Applications, pp. 1–33. Springer, Germany (2005)
6. Yin, R.: Case Study Research: Design and Methods, 4th edn. Applied Social Research Methods Series, vol. 5. SAGE, Thousand Oaks (2009)

A Proposal for Developing a Primer for Constructing and Analyzing Conceptual Structures

Nagarjuna G. and Meena Kharatmal

Homi Bhabha Centre for Science Education (TIFR),
V.N. Purav Marg, Mankhurd, Mumbai 400 088, India
nagarjun@gnnowledge.org, meena@hbcse.tifr.res.in

Abstract. A rationale and proposal for developing a primer for teaching-learning of conceptual structures is presented. The skills required and developed by an engagement of constructing and analyzing conceptual structures are richer and easier to be dealt with in school education. The teaching-learning context of CS is fundamental and important enough to introduce the topics from logic, philosophy, computer science and linguistics. A proposal is made for the formation of a special interest group for the primer.

Keywords: primer, conceptual structures, education, knowledge representation, propositions, logic, philosophy, teaching-learning.

1 Introduction

This is a proposal for developing a primer on conceptual structures addressing: (a) reasons to learn or teach conceptual structures (CS), (b) background knowledge required for representing and analyzing CS, (c) activities and exercises that will help develop the skills required and (d) to provide a glimpse of the various applications of conceptual structures. After going through the primer the learner can take an undergraduate level (101) beginner course on CS. This paper provides a justification for why such a primer is required, why CS-LTA community should spend time in developing this primer, and suggests an outline of the primer for discussion at the workshop.

2 Rationale

Knowledge representation skills are almost mandatory for those who decided to work in Artificial Intelligence, Logic, Databases, Semantic Web, Linguistics and Philosophy. The community of experts who work in these areas are primarily responsible for developing as well as popularizing this discipline. A repertoire of skills required during the development of these disciplines however have a wider use. One such use is school education. As researchers working in science

education, we think teaching the basics of learning conceptual structures will play an important role in developing rigor, critical reasoning, clarity of thought and expression [1]. Since these skills are essential for mastering any discipline, and if our diagnosis that CS will help the students develop these skills is correct, we can argue that the teaching and learning of CS is essential for higher education of any discipline. Therefore, we would like to make a strong case by suggesting that just as basic arithmetic, algebra and geometry are considered essential for all those who graduate from school, a basic course on conceptual structures (which includes logic) is essential. We need modules that impart skills in constructing and analyzing conceptual structures through basic exercises. Availability of a primer in the form of independent units/modules would facilitate easy integration in the school curricula.

3 Networking with Other Initiatives

Concept maps are already being widely used in school education. These are considered to be imparting meaningful understanding, eliciting knowledge, evaluating students' understanding, for pedagogical designing, lesson planning, etc. [2]. For the purpose of general education, concept maps bear significance, however, when it comes to its use in representing scientific knowledge, these are considered to be informal [3]. The freedom to choose the linking words makes it vulnerable to creating ambiguous representations, hence are not considered to generate the precision and rigor that is required in scientific knowledge [4].

Not surprisingly, there exist groups who do believe that logic, philosophy etc. should be taught as a subject matter even at the school level. Some studies have presented course material of logical reasoning in the high school. It is also very important to introduce logical reasoning along with mathematics to school students as young as 10-12 years old [5]. To help cope up with the information explosion, courses were designed to impart skills to organize, critically read, analyze and evaluate the content [6]. Moreover, these courses were offered for the students enrolled in both the pure sciences as well as social sciences.

There have been tools designed to teach Aristotelian syllogism for elementary logic course using PrologPlusCG [7]. The objective of the idea is to improve argumentation skills and an in-depth understanding of logic, logical reasoning, and conceptual structures. Conceptual graphs were used in business computing wherein students were engaged in inquiry based learning enterprise [8]. In another study Prolog environment was introduced in a course in computer science for developing an understanding of the first order logic. The course trained them in problem solving and knowledge representation skills. Such students were reported to be successful in developing knowledge-based projects in pure and applied sciences such as, biology, medicine, chemistry, archeology, mathematics and geometry [9].

Teaching-learning of philosophy at the school level is already tried in the UK under the program of Philosophy in Children to develop critical thinking skills [10,11]. Another very active group, International History, Philosophy, and

Science Teaching Group [12] is promoting school and university science education as informed by the historical, philosophical and sociological issues of sciences. Apart from the high school students, recent developments in science such as bioinformatics demand knowledge of creating ontologies, semantic web, logic and semantics [13]. Since subject experts of these domains do not have a sound knowledge of logic or semantics, it is felt that an introductory course on conceptual structures would be useful.

Though the efforts of introducing logic, philosophy, computer science etc. are laudable in themselves, we think that the teaching-learning context of CS is richer and easier to introduce the topics from logic, philosophy, computer science and linguistics. When these subject are done independently, the subject matter becomes more abstract, highly formal and therefore difficult. Since CS is less formal than logic and computer science it can help bridge the gap while learning the more formal subjects. One possible way is to introduce the teaching learning of CS as modules within the existing courses [14]. The reasoning that learners apply during constructing CS gives ample opportunities to introduce logic, philosophy and linguistics. The proposed primer is an attempt to meet the objectives of developing rigor, critical reasoning, clarity of thought and expression, as well as an entry to the existing books on conceptual structures [15,16,17] which are targeted at the undergraduate level.

4 Special Interest Group

The development of proposed primer could be carried by a collaborative effort with groups of experts from the disciplines mentioned above. Conducting field trials may be necessary to check the effectiveness of the modules. Inorder to achieve these tasks, we suggest the formation of a Special Interest Group (SIG) preferably as an initiative of CS-LTA. This group may also invite people from the other groups such as [18,19] who attempted to bring in logic, philosophy and computer science into school education.

5 A Draft Framework of the Primer

As a framework for the primer, we have sketched a draft plan of the units which comprises of learning objectives. We propose the units be framed according to the level of difficulty and an appropriate teaching-learning sequence. The framework can be viewed from <http://gnowledge.org/~meena/primer-framework.pdf> to get an idea of the units proposed. The items were mostly chosen from the teaching-learning sequence suggested by the dependency mapping for conceptual structures [20].

References

1. Kharatmal, M., Nagarjuna, G.: Introducing Rigor in Concept Maps. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 199–202. Springer, Heidelberg (2010)

2. Mintzes, J.J., Wandersee, J., Novak, J.D. (eds.): Teaching Science for Understanding— A Human Constructivist View. Academic Press, USA (1998)
3. Sowa, J.: Concept Mapping, Concept mapping. Talk presented at the AERA Conference, San Francisco (2006), <http://www.jfsowa.com/talks/cmapping.pdf>
4. Kharatmal, M., Nagarjuna, G.: A Proposal to Refine Concept Mapping for Effective Science Learning. In: Canas, A.J., Novak, J.D. (eds.) Concept Maps: Theory, Methodology, Technology. Proceedings of the Second International Conference on Concept Mapping, San Jose, Costa Rica (2006)
5. Huertas, A.: Teaching Logical Reasoning in High School. Presented at the First International Congress on Tools for Teaching Logic (2000)
6. Bouhnik, D., Giat, Y.: Teaching High School Students Applied Logical Reasoning. Journal of Information Technology Education. Innovations in Practice. 8 (2009)
7. Øhrstrøm, P., Sandborg-Petersen, U., Ploug, T.: Syllogistics with PrologPlusCG: Syllog - A Tool for Logical Teaching. In: Polovina, S., et al. (eds.) Artificial Intelligence Workshops: 1st Conceptual Structures - Learning, Teaching and Assessment Workshop at ICCS 2010, Kuching, Malaysia (2010)
8. Lauenders, I., Polovina, S., Khazaei, B.: Learning Perspectives of Enterprise Architecture through TrAM. In: Polovina, S., et al. (eds.) Artificial Intelligence Workshops: 1st Conceptual Structures - Learning, Teaching and Assessment Workshop at ICCS 2010, Kuching, Malaysia (2010)
9. Zahava, S., Bruria, H.: Logic Programming Based Curriculum for High School Students: The Use of Abstract Data Types. Technical Symposium on Computer Science Education: Proceedings of the Twenty-Sixth SIGCSE Technical Symposium on Computer Science Education (1995)
10. Fisher, R.: Philosophy in Primary Schools: Fostering thinking skills and literacy, Reading (2001)
11. Hill, R.: Culture, Critical Thinking and Computing. In: Polovina, S., et al. (eds.) Artificial Intelligence Workshops: 1st Conceptual Structures - Learning, Teaching and Assessment Workshop at ICCS 2010, Kuching, Malaysia (2010)
12. International History, Philosophy, and Science Teaching Group, <http://ihpst.net>
13. The Open Biological and Biomedical Ontologies, <http://www.obofoundry.org>
14. Andrews, S.: Aligning the Topic of FCA with Existing Module - Learning Outcomes. In: Polovina, S., et al. (eds.) Artificial Intelligence Workshops: 1st Conceptual Structures - Learning, Teaching and Assessment Workshop at ICCS 2010, Kuching, Malaysia (2010)
15. Sowa, J.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley Publishing Company, USA (1984)
16. Sowa, J.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Brooks/Cole, USA (2003)
17. Chein, M., Mugnier, M.: Graph-based Knowledge Representation. Computational Foundations of Conceptual Graphs. Springer, Heidelberg (2009)
18. International Congress on Tools for Teaching Logic, <http://logicae.usal.es/TICTTL/>
19. Philosophy Learning and Teaching Organization, <http://plato-apa.org/>
20. Nagarjuna, G., Kharatmal, M., Nair, R.: Building Dependency Network for Teaching-Learning of Conceptual Structures. In: Polovina, S., et al. (eds.) Artificial Intelligence Workshops: 1st Conceptual Structures - Learning, Teaching and Assessment Workshop at ICCS 2010, Kuching, Malaysia (2010)

Internationalising the Computing Curricula: A Peircian Approach

Richard Hill¹ and Dharmendra Shadija²

¹ School of Computing and Mathematics,
University of Derby,
Derby, DE22 1GB, UK
r.hill@derby.ac.uk

² Department of Computing,
Sheffield Hallam University,
Sheffield, S1 1WB, UK
d.shadija@shu.ac.uk

Abstract. The internationalisation of Higher Education is a major thrust for EU Universities. This article examines how the curriculum can be designed to not only accommodate different and disparate cultures, but also enhance the cultural experience for all concerned. Using Peirce's writings on critical thinking, we describe how a research-informed curriculum can deliver an improved experience for postgraduate learners.

1 Introduction

Internationalisation of the UK Higher Education experience is an issue for many institutions [5], [10]. The decline in UK government funded undergraduate places and (apparently) unrestricted numbers of non-European Union students, makes a compelling argument to meet the needs of a different marketplace. However, this is not a trivial undertaking [4] and as some HE institutions have discovered already, a change in marketing approach cannot fully support the deep-seated differences of disparate learning cultures [5].

It is apparent that the 'culture' word is used somewhat perjoratively in that it becomes a general container for all of the unexplained behaviours that students of a different culture can exhibit. Apparent plagiarism, or perceived academic dishonesty, is often described as one of the most visible behaviours from students from South East Asia and the Indian Sub Continent.

This state of affairs has been described as a lack of understanding of what is meant by 'critical thinking'. In this article, the authors hope to proffer a more sophisticated response, based upon Peirce's philosophical writings. We start by examining what is meant by critical thinking.

2 Critical Thinking

Many academic teaching staff pride themselves on being part of the educational environment that is purported to 'teach' critical thinking. But what does it

mean to think critically? Wells argues that critical thinking requires an ability to objectively reason [18]. The behaviours of critical thinking include an ability to scrutinise scenarios through the lens of another perspective. A critical thinker will not be unduly influenced by a hastily, or poorly constructed argument. Another dimension is that of the moral obligation of thinking critically [18];

The critical thinker should bring a voice of reason to this discussion.

The moral position is important in that is usually the key justification for the teaching of critical thinking. Such a premise suggests that an individual will be ‘improved’ by undertaking the study of critical thinking, and is therefore, worthwhile study.

2.1 Cultural Perspectives

Critical thinking embodies the beliefs and approach to thinking of western societies. In cases of alleged plagiarism amongst international students, it is common to hear that the cause is that of ‘cultural background’. If we adopt this premise, then it must be recognised that this view is being made about a particular culture, from the perspective of another, disparate culture. What might be judged a more intolerant society must therefore restrict any ability to develop critical thinking behaviours. After teaching hundreds of students from South East Asia and the Indian Sub Continent we have witnessed behaviours that demonstrate the presence of critical thinking. However, we have also witnessed the challenges faced by students when faced with an intrinsically western approach to thinking for the first time. Peirce’s writings offer many philosophical arguments that can be utilised to understand critical thinking. ‘The Fixation of Belief’ [11] indicates an approach to understanding how critical thinking might come about. We now proceed to examine these ideas in order to inform our approach to delivering a more culturally-aware computing curriculum.

3 The Peircian Approach

Peirce offers a model by which we arrive at a belief in such a way that we are inclined to maintain that belief [11]. In other words, the conditions required for a belief to be acquired and held. Peirce summarises the four key scenarios as follows:

1. *Tenacity* - The holding of a belief without any consideration of arguments to the contrary. Any idea that counters what the individual already believes is immediately discounted. Many students exhibit this behaviour, irrespective of their cultural upbringing.
2. *Authority* - Here, Peirce refers to the influence exerted by external agents to enforce the holding of a belief. If multiple agents are collectively *tenacious* then a considerable influence can be exerted over an individual. Peer pressure is an example of *authority*.

3. *A priority* - In some cases there may be a tendency to reason between alternatives, but those alternatives are actually based upon pre-ordained outcomes, thus there is no basis for critical thinking. This state can be misleading in that there may be a perception that critical thinking is taking place. One challenging aspect is that this potentially dishonest practice could be unwittingly fostered amongst students through the delivery of curricula if the conditions for critical thinking are ignored.
4. *The scientific* - This represents a condition whereby a conclusion is reached through a managed process of systematic, objective enquiry. There is a clear difference in the behaviour of an individual who operates in this state and students unanimously appear to aspire to be able to operate in such a way. The development of the personal characteristics required however, is challenging to facilitate.

Thus we have some conditions that serve to identify the characteristics of thinking critically, but perhaps more importantly serve to identify scenarios where pseudo-critical thinking is occurring.

3.1 Managing Doubt

One aspect that is common to all of the conditions above is the feeling of doubt. Peirce explains belief as something that opposes doubt; a belief is much stronger than an opinion [18]. A doubt is a state whereby an individual feels uncomfortable until a belief is achieved. This may provide some impetus to reach a premature belief, if only to achieve a degree of comfort. To think critically, it would seem, requires effort and tolerance. The logical conclusion to be drawn from a feeling of doubt is to conduct enquiry, with the ultimate aim of resolving the doubt. Peirce describes this stage as ‘managing doubt’. Essentially we need to create the conditions where students have doubts that they can manage towards a more enlightened, reasoned understanding. The engagement with this process may demonstrate the behaviours of critical thinking. Unfortunately the realities and practicalities of the academic environment are such that the situation is far more convoluted. Firstly, a cohort of new students, or established students faced with a new topic, is riddled with doubts. There will be doubts that are readily predicted by the tutor, as well as doubts that are personal to an individual. Secondly, individuals will succumb to different conditions based upon their existing behaviours. So an individual may be *tenacious* and therefore have a reluctance to experience any doubtful feelings. Or the *authority* of the collective may come into play and create a compelling reaction that is difficult to resist. Thus it is the management of doubt that is important for the tutor to recognise, and ultimately the challenge for the educational setting to replicate.

4 A Case Study: Teaching Introductory Programming

Sheffield Hallam University has a range of postgraduate computing programmes that recruit significant numbers of students from South East Asia and the Indian

Sub Continent. Whilst these programmes foster the development of advanced computing skills, the programmes are designed to accommodate individuals who are moving from one technical discipline to another also. As such there are some individual modules common to all of the programmes:

- Advanced Learning and Study Skills;
- Research Principles and Practice;
- Industrial Expertise;
- Dissertation

These modules provide the context for the higher order thinking that is demanded by Masters level study. Whilst these modules may appear to be quite generalist, there is one other computing module, *Web Application Design and Modelling* (WADM), that attempts to normalise students' prior experience of computing across the cohort. WADM was originally intended to convey the essential software design and principles of application programming for web-based architectures, and is heavily biased towards the acquisition of foundation computing skills. For some time there was a noticeable benefit derived from the module, but there was an immediate set of issues presented once the demographics of the recruited cohort became mostly from South East Asia and the Indian Sub Continent.

An immediate effect with the increased numbers of international students was a sharp increase in the number of plagiarism cases. It appeared that international students could not differentiate between making distinct, individual contributions to a group activity, and then presenting their individual work for the purposes of assessment.

Secondly there was a pre-occupation with the desire to learn a particular programming language syntax, without wanting to learn the conceptual basis of programming.

Thirdly it was clear that the ambiguity experienced during requirements gathering and design stages was presenting difficulties for students. For instance, students were unwilling to publicly disagree with peers when discussing functional requirements; erroneous ideas were being left unchallenged, creating subsequent design hurdles later on.

Lastly, there was a great deal of anxiety surrounding any assessment that was not a time-constrained examination. Interestingly examination performance was generally good, but coursework and practical demonstrations were poor. This last point created the greatest academic concern, since potential employee's skills are appraised by their application of knowledge to practical scenarios. Good examination performance is only a small part of a holistic assessment package, and clearly a practical demonstration can be a much more effective indicator of how the student may perform in a work-based scenario. Since the practical demonstrations were disappointing, a re-think in terms of learning and teaching approach was required.

4.1 Learning Outcomes

Learning Outcomes (LO) [15] are an established means of communicating the intentions of a course of study to students. Biggs has popularised an approach referred to as ‘constructive alignment’ [1], [3] which is based upon the premise that the students will take more responsibility for their own learning if they understand what the intended learning outcomes (ILO) are. To quote Shuell:

If students are to learn desired outcomes in a reasonably effective manner, then the teacher’s fundamental task is to get students to engage in learning activities that are likely to result in their achieving those outcomes. [16], p429.

Often it is evident that the assessment vehicle becomes the focus of the learning. Speculation as to how an individual will be assessed can adversely affect the behaviour of an individual in terms of their learning [2]. Gibbs argues that the assessment itself can be used to increase engagement [7]. An alternative is the traditional examination, but it is widely recognised that this particular mode is biased towards testing recall [6].

In many cases, at UK level 6 (undergraduate final year) and level 7 (post-graduate) the LO will make specific reference to being critical:

- *Critically appraise and select the most appropriate design pattern*

Biggs’ argument is that the LO should align with both the learning and assessment activities to assist the student to construct their own learning. There can be a tendency to include the use of ‘critical’ since it is generally considered to be good practice. However, if critical thinking is to be fostered it is necessary to make this clear at the outset. As such there is a need to establish doubt, but also manage the maintenance of the doubt until the learner is sufficiently confident to construct their own learning.

4.2 Module Organisation

Our approach has been to design a curriculum that takes the students through Peirce’s four conditions to help students experience and manage their own doubt. This is achieved by placing less emphasis upon a scheduled delivery of material in favour of a curriculum informed by the following principles:

- *Opportunities for self assessment* - when the module commences each student is required to assess their own strengths, firstly with reference to a professional development framework (SFIA [17]) and then using their own criteria.
- *Reflection as a professional activity* - all in-class activity is recorded within a weekly discussion forum (in the University’s Virtual Learning Environment), as well as a private online learning journal in the form of a blog.
- *Open debates* - questions that raise doubts are given to the class to openly debate. Since there is a significant effect from the collective authority, debates are managed in pairs, groups of four, then the whole class.

- *Frequent, prompted reflection in action* [14] - students are actively prompted to regularly reflect *in action*, and then post session *on* reflection.
- *Impromptu presentations* - opportunities to discuss, explore and present ideas are taken frequently to provide as much practice as possible.
- *Regular micro-teaching* - students are encouraged to teach each other challenging, but focused topics.
- *Large group tutorials to replace lecture presentations* - this was a controversial decision, but it was deemed necessary to maximise the amount of time spent facilitating critical behaviours and so all module materials were provided online as an alternative.
- *Feedback for learning* - all of the above principles are directly influenced by the notion that feedback be offered, exchanged, recorded and collated for the purposes of learning. In particular such feedback should be generated by student-to-student interaction rather than being limited to more formal, summative assessment feedback.

What this means in practice is that each of the principles above requires the tutor's focus upon the *process* of learning rather than the delivery of *content*. There are many opportunities for tutors to raise doubts, by posing questions. However, there is often less emphasis upon managing the process between the initial doubt and its eradication. We have attempted to address this by placing the emphasis upon guiding the learner to their own conclusion by following a process of systematic enquiry.

5 Discussion

In many cases students are taught 'how to reference' as a means of demonstrating 'academic integrity'. In the absence of any additional contextual information, tentative links are made between referencing and critical thinking (erroneously), which severely hampers progress. Once the conceptual basis of information gathering and attribution to the correct sources is understood, academic 'offences' seem to diminish. We firmly believe that a return to a more abstract curriculum, both in terms of the content and the way it is delivered, directly supports international students' successful achievement.


Additionally we have extended this understanding by providing a range of activities where critical thinking can be applied, that might not necessarily lead to a referenced piece of writing. The opportunity to engage with, and understand the processes of thought, discussion, evaluation and group consensus serves to communicate the true value of research activity as a means of developing new cognitive skills. Our approach to curriculum design is embodied by the mapping of the principles identified in section 4.2 to behaviours that are conducive to the management of doubt towards Peirce's *the scientific*. Thus each learning session should be seen as an opportunity to draw from a range of activities, rather than being overly prescriptive from the outset. Whilst the more experienced tutors may regard this aspect as appearing 'obvious', it is useful nonetheless to have

a set of principles that can be used as a prompt during the design of curricula. Additionally, as tutors gain more experience they often rely less upon prescriptive preparation and move towards more creative, dynamic, process-driven approaches. We feel that the identification of Peirce's thinking in relation to the internationalisation of UK Higher Education is a useful move towards accelerating this agenda.

Furthermore the recognition that to think critically is inextricably entwined with an individual's cultural background is something that needs to be aired at the earliest opportunity. Using Peirce's classification we have witnessed more ideas based upon *authority* with students from South East Asia and the Indian Sub Continent. This is most prevalent when ideas are being explored in public, where one might expect the authority of the masses to be greatest. However, through the use of journalling and activities designed to promote the development of reflection we have also identified that Peirce's *scientific* also exists, albeit privately. The challenge therefore is to develop activities and the necessary fora to support the more public sharing of these ideas. This approach has enabled learners to contribute more in terms of the emergent design of a course as they are studying it. The objective is still to teach programming; the actual manifestation of this is more learner directed and researched content.

6 Conclusion

Peirce's conditions have served to frame some of the scenarios and conditions that are exhibited by pseudo-critical thinking behaviour. Unfortunately there are many cases where both tutors and learners genuinely believe that critical thinking is occurring. Peirce's insight into the process of managing doubt, serves to identify how we can become more aware of the processes of learning, and presents opportunities to improve the underlying rigour of our teaching philosophies. This improves the tutor's understanding of the complexities of this teaching context, and also serves to improve the students' comprehension, by providing a more informed set of learning experiences. Essentially, the students are learning *what they need to know* in the context of *their own learning needs* by following the processes of research, yet the actual technical content delivery has been mostly relegated to online documentation. The actual delivery concentrates upon the teaching of *process*, and is characterised by a set of principles that create the setting for critical thinking to flourish.

Acknowledgement. This article is a revised and updated version of that presented at CS-LTA 2010 .

References

1. Biggs, J.: Teaching for Quality Learning at University. The Society for Research into Higher Education. Open University Press, McGraw-Hill Education (2003)
2. Biggs, J.: Aligning the curriculum to promote good learning. In: Constructive Alignment in Action: Imaginative Curriculum Symposium, LTSN Generic Centre (2002), <http://www.palatine.ac.uk/files/1023.pdf>

3. Biggs, J.: Enhancing teaching through constructive alignment. *Higher Education* 32, 347–364 (1996)
4. Carroll, J., Ryan, J. (eds.): *Teaching International Students. Improving learning for all.* Routledge, Oxon (2005)
5. Caruana, V., Spurling, N.: *The internationalisation of UK Higher Education: a review of selected material: project report*, 147 pages. Higher Education Academy, York England (2007),
<http://www.heacademy.ac.uk/ourwork/learning/international>
(accessed June 2010)
6. Elton, L., Johnston, B.: *Assessment in universities: a critical review of research*, LTSN Generic Centre (2002),
<http://eprints.soton.ac.uk/59244/01/59244.pdf>
7. Gibbs, G.: Using assessment strategically to change the way students learn. In: Brown, S., Glasner, A. (eds.) *Assessment Matters in Higher Education.* The Society for Research into Higher Education. Open University Press, Buckingham (1999)
8. Hill, R.: *Culture, Critical Thinking and Computing.* In: Polovina, S., et al. (eds.) *Artificial Intelligence Workshops: 1st Conceptual Structures - Learning, Teaching and Assessment Workshop at ICCS 2010*, Kuching, Malaysia (2010)
9. Hill, R.: *Why should I do this? Making the information systems curriculum relevant to strategic learners.* *ITALICS Journal*, Higher Education Academy Information and Computer Science Subject Centre (June 2009)
10. Knight, J.: *Internationalization: A decade of changes and challenges.* *International Higher Education* 50, 6–7 (2008),
http://www.bc.edu/bc_org/avp/soe/cihe/newsletter/Number50/p6_Knight.htm
(accessed June 2010)
11. Peirce, C.S.: *The Fixation of Belief*, *Popular Science Monthly*, 12, pp. 1-15 (1877); Reprinted in *Collected Papers of Charles Sanders Peirce*, Vol. V, pp. 358-87
12. Peirce, C.S.: *How to Make or Ideas Clear*, *Popular Science Monthly*, 12, pp. 286-302 (1878); Reprinted in *Collected Papers of Charles Sanders Peirce*, Vol. V, pp. 338-410
13. Peirce, C.S.: *Collected Papers of Charles Sanders Peirce.* In: Hartshorne, C., Weiss, P. (eds.) *Pragmatism and Pragmaticism*, vol. V, pp. 334–335. Belknap Press, Harvard University, Cambridge (1934)
14. Schon, D.: *The Reflective Practitioner.* Temple Smith, London (1983)
15. Scroggins, W.: *Student learning outcomes - a focus on results (Modesto Junior College)* (2004), <http://www.crconsortium.com/images/SLOfocuson.pdf>
16. Shuell, T.J.: *Cognitive conceptions of learning.* *Review of Educational Research* 56, 411–436 (1986)
17. *Skills Framework for the Information Age (SFIA)*, <http://www.sfia.org.uk/> (last accessed June 28, 2010)
18. Wells, K.: *Learning and Teaching Critical Thinking: From a Peircean Perspective.* *Educational Philosophy and Theory* 41(2), 201–218, Philosophy of Education Society of Australasia (2007),
<http://dx.doi.org/10.1111/j.1469-5812.2007.00376.x>

Broadening the Ontological Perspectives in Science Learning: Implications for Research and Practice in Science Teaching*

Nancy R. Romance¹ and Michael R. Vitale²

¹ Florida Atlantic University, Boca Raton, FL 33431 USA

² East Carolina University, Greenville, NC 27858 USA
romance@fau.edu, vitalem@ecu.edu

Abstract. The argument presented in this paper is that efforts designed to engender systemic advancements in science education for fostering the scientific literacy of learners are directly related to the ontological perspectives held by members of the discipline. In elaborating this argument, illustrative disciplinary perspectives representing three complementary aspects of science education are addressed. These three perspectives represent the disciplinary knowledge and associated dynamics of: (a) science students, (b) science teachers, and (c) science education researchers. In addressing the ontological perspectives of each, the paper emphasizes how interdisciplinary perspectives can accelerate progress in science education.

1 The Function of Ontology in Science Education

The focus of this paper is ontological functions rather than general philosophical issues. This section emphasizes the interrelationship of ontology with knowledge representation as considered in computer-oriented cognitive science. An ontology is the product of the study of categories of things that exist or may exist within a domain [1]. More specifically, the categories of an ontology consist of the predicates, concepts, or relationships used to represent, provide focus on, and allow discussion of topics in the domain. An ontology and its categories impose an intellectual structure on what the substantive aspects of a domain are and how they are characterized.

The issue of ontology is highly relevant to science students, science teachers, and science education researchers. From the standpoint of science students, ontology reflects the core concepts and principles within science that constitute the major learning goal. To achieve the learning goal, students must understand how the hierarchical structure of a domain translates directly into building a schematic framework for core concepts and core concept relationships which serves as the basis for knowledge applications and as prior knowledge for further learning.

* This research was supported by the National Science Foundation, USA REC 0228353.

The ontological framework for teachers and researchers in science education is broader than that of students because, in addition to core science content, their framework must include the additional pedagogical and/or research knowledge that represents their professional roles. For teachers, their expanded ontological function has to do with the conceptual understanding of both the science to be taught and the means for planning, conducting, and communicating all aspects of science teaching. For researchers, their expanded ontological function encompasses that of teachers along with the additional knowledge (e.g., theories, research findings, research methodology) that forms the intellectual basis for being a member of the science education research community.

An important issue relevant to this paper is the fundamental distinction between ontology and logic [1]. In comparison to an ontology which consists of a substantive categorization of a domain, logic is neutral. That is, logic itself imposes no constraints on subject matter or the way the domain may be characterized. As Sowa [1] noted, the combination of logic with an ontology provides a language that has the means to express extensible inferential relationships about the entities in a domain of interest.

2 Linkage between Ontology and Knowledge Representation

Within cognitive science, the area of knowledge representation is closely related to that of ontology. As noted by Davis, Schrobe, and Szolovita [2], ontology determines the categories of things that exist or may exist and, in turn, these categories represent a form of ontological commitment as to what may be represented about a domain. At the same time, Sowa [1] pointed out that everyday knowledge is far too complex, fluid, and inconsistent to be represented comprehensively in any explicit system. Rather, because of the complexity of the world, knowledge is better considered a form of soup about which explicit systems of knowledge representation can only address selected structural aspects. Among the most important factors affecting consistency in the ontological representation of the same phenomena are multiple uses of the same words, vagueness in scientific language, and/or the interaction of multiple perspectives (i.e., different views).

As applied to science in general, every branch uses models that enhance certain features and ignore others, even within the hard (vs. behavioral) sciences. Areas of science can be considered as a collection of subfields, each focusing on a narrow range of phenomena for which the relevance of possible features is determined by a perspective for which details outside the primary focus of attention are ignored, simplified, or approximated. In the present paper, this suggests that the integration of interdisciplinary views, all relevant in different ways to the three aspects of science education (student learning, teaching, research), has a substantial potential to accelerate the advancement of disciplinary knowledge, even in the face of paradigmatic resistance from within the individual sub disciplines themselves (see [3]).

3 Knowledge-Based Instruction as a Framework for Science Education

An informal review of science education research trends in scholarly journals, handbooks, and textbooks revealed a surprising finding. In fact, relatively few of the studies in science education involve experimental (or field experimental) research that demonstrates the effect of approaches to or characteristics of science instruction on meaningful conceptual understanding by students in school settings [4]. Rather, the majority of science education studies (a) describe teacher experiences in science instructional settings, (b) evaluate student misconceptions (including reporting teacher frustration on the resistance of student misconceptions to conceptual change), or (c) use science content as an incidental research context (vs. focusing on in-depth science content) as a setting for the exploration of other concerns (e.g., equity/gender issues, use of professional development strategies, explorations focusing primarily on the processes of teaching using constructivist, cooperative learning, or inquiry/questioning strategies).

In comparison to science education, research from related disciplines (e.g., cognitive science, instructional psychology) provide rich perspectives and findings that bear upon the improvement of science teaching and learning. This section emphasizes research findings whose foundations which are grounded in interdisciplinary research fields having implications for improving student meaningful learning of science.

The idea of knowledge-based models comes from expert systems applications in computer science developed in late 1970. All such models met the requirement that the knowledge representing expertise was encoded in a fashion that was separate and distinct from other parts of the software that operated on the knowledge-base (e.g., to diagnose problems and offer advice). Building on the original expert systems, a new form of knowledge-based instructional architectures called intelligent tutoring systems (ITS) were developed in the 1980s [5]. In these systems, an explicit representation of knowledge to be learned provided an organizational framework for all elements of instruction, including the determination of learning sequences, the selection of teaching methods, the specific activities required of learners, and the evaluative assessment of student learning progress.

Figure 1 shows a propositional concept map that illustrates how concepts within a domain can be organized in a way to insure instructional coherence [6]. Using Figure 1 as a curricular framework, teachers are able to locate and then sequence reading/language arts and hands-on activities by linking them as elements to concepts on the map [7]. As a result, teachers are able insure that instruction is highly coherent in a manner that expands student in-depth science knowledge in a cumulative fashion. Referencing the curricular framework as a guide, teachers also are able to apply a coherent inquiry-oriented approach that (a) emphasizes what additional knowledge is learned over a sequence of related instructional activities that results in additional knowledge and understanding and (b) guides students to relate what they have learned as representations or elaborations of the core concepts. Overall, the foundational ideas underlying

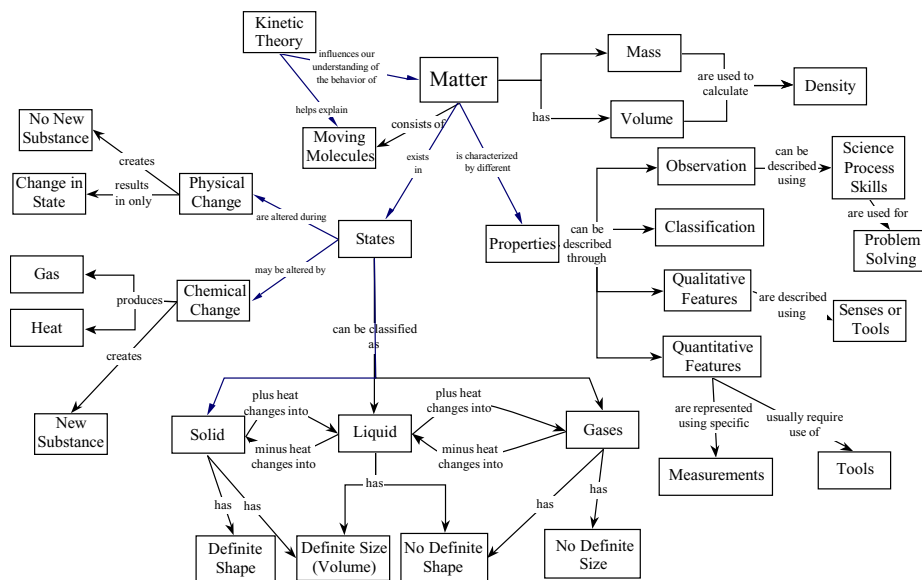


Fig. 1. Simplified illustration of a propositional curriculum concept map used by grade 3-4 Science IDEAS teachers to plan a sequences of science instructional activities [7]

knowledge-based instruction models are that (a) curricular mastery can be considered to be and approached as a form of expertise, and (b) the development of prior knowledge is the most critical determinant of success in meaningful learning.

The National Research Panel publication, *How People Learn* [8], serves as a guide for the interdisciplinary interpretation of the relevance of knowledge-based perspectives to science education. Focusing on meaningful learning, Bransford et al emphasized that to teach effectively, the knowledge being taught must be linked to the key organizing principles of that discipline. Such organized and accessible prior knowledge is the major determinant in developing the forms of cumulative learning consistent with the expertise characteristic of scientists. All forms of science pedagogy should explicitly focus upon the core concepts representing the ontological structure of the discipline.

In relating prior knowledge to meaningful learning, Bransford et al. [8] focused on the cognitive differences between experts and novices and showed that expert knowledge is organized in a conceptual fashion that differs from novices and that the use of knowledge by experts in application tasks is primarily a matter of accessing and applying prior knowledge under conditions of automaticity. Related is work by Anderson and others [9,10,11] who distinguished the strong problem solving process of experts as highly knowledge-based and automatic from the weak strategies that novices with minimal knowledge are forced to adopt in a trial-and-error fashion. Andersons cognitive theory suggests that all learning tasks should (a) consider all cognitive skills as forms of proficiency

that are knowledge-based, (b) distinguish between declarative and procedural knowledge (i.e., knowing about vs. applying knowledge), and (c) identify the conditions in learning environments (e.g., extensive practice) that determine the transformation of declarative to procedural knowledge (i.e., apply knowledge in various ways).

In characterizing the learning processes, this interdisciplinary research perspective emphasizes that extensive amounts of varied experiences (i.e., initially massed followed by diverse distributed practice) involving the core concept relationships to be learned are critical to the development of expert mastery in any discipline [12]. Others [13] explored the conditions under which extensive practice to automaticity focusing on one subset of relationships results in the learning of additional subsets of relationships.

For science education, a knowledge-based approach suggests that the cumulative experiences of students in developing conceptual understanding (i.e., expertise) implies the development of a framework of general ontological (knowledge) categories in the form of core concepts/concept relationships. Thus, additional knowledge is first assimilated and then used as a form of expertise by students as prior knowledge for new learning. Such expertise facilitates students cumulatively acquiring, organizing, accessing, and thinking about new information that is embedded in comprehension and other meaningful tasks to which such new knowledge is relevant [14].

4 Ontological Implications of Knowledge-Based Instruction for Research and Practice in Science Education

Each of the perspectives illustrated below are grounded in disciplines other than science education. As a result, consideration of their potential application to science education has major ontological implications for the discipline and poses paradigmatic implications as well.

The major ontological implication for science learning is the importance of focusing all aspects of instruction on student mastery of core concepts and relationships. This implies a very different curricular approach at both the elementary and secondary levels (which typically emphasize a variety of rote hands-on activities), one that would emphasize the cumulative development of conceptual understanding that is consistent with that of scientists and has implications for significant curricular reform [7].

Within a knowledge-based context, the first consideration consists of a curricular distinction regarding the observational basis for science concepts taught [15]. They distinguished among three types of science concepts: (a) concepts which students could observe directly, (b) concepts which could be observed but for which observation was not feasible (e.g., observing the earth and moon from space), and (c) concepts which are artificial or technical symbolic constructs created by the discipline for which the notion of exhaustive direct observation

within a learning setting does not apply (i.e., they represent labels for complex relationships that are tied to observation in an abstract fashion). Certainly the three types of concepts require substantially different curricular strategies for teaching [16] with types two and three being more difficult. However, in practice, virtually no distinction is made between them (e.g., young students are taught graphic representations of atoms and molecules with no operational association to observable phenomena), a curricular consideration that impacts teaching and learning in science.

Presented next are interdisciplinary research exemplars that have ontological implications for science education, considered from a knowledge-based approach to student learning. Each serves two major functions. The first is to illustrate one or more major points within applied science learning contexts or experimental settings. The second is to point out that despite the fact that the exemplars provide specific implications for improving the quality of school science instruction, they cannot be represented within the current ontological framework of science education at the appropriate level of detail.

The curricular findings of the highly-respected TIMSS study [17] provide a strong knowledge-based framework for considering the exemplars presented. In comparing the science curricula of high achieving and low achieving countries, the TIMSS study found that the curricula of high achieving countries were conceptually focused (on core concepts), coherent, and carefully articulated across grade levels while that in low-achieving countries emphasized superficial coverage of numerous topics with little conceptual emphasis or depth and that were addressed in a highly fragmented fashion.

The first exemplar is the work of Novak and Gowin [18] who studied the developmental understanding of science concepts by elementary students over a 12 year period. In their longitudinal study, concept maps were used to represent the cumulative development of student understanding of science topics based on interviews and initiated the use of concept maps by students to enhance their understanding of science [19]. Overall, these studies demonstrated the importance of insuring students have the means to understand the development of their own views of core concept relationships.

The second exemplar is a videodisk-based instructional program by Hofmeister et al. [20] that focuses on the development of core science concepts in physical science (e.g., heating, cooling, force, density, pressure) that are necessary to understand phenomena in earth science (e.g., understanding how the concept of convection causes crustal, oceanic, and atmospheric movement). Two complementary studies are relevant here. Muthukrishna [21] demonstrated experimentally that use of the videodisk-based materials to directly teach core concepts was an effective way to eliminate common misconceptions (e.g., cause of seasons) of elementary students while Vitale and Romance [22] showed in a controlled study that the use of the same instructional program resulted in mastery of the core concepts by elementary teachers (vs. control teachers who demonstrated virtually no conceptual understanding of the same content). These studies suggest

that focusing instruction on core concepts is important for meaningful learning in science.

The third exemplar is a series of studies at the elementary and postsecondary levels. In an analyses of learning by elementary students, Vosniadou [23] showed that concepts have a relational nature that influences their order of acquisition in order for students to gain meaningful understanding. Dufresne et al. [24] found that postsecondary students who engaged in analyses of physics problems based upon a conceptual hierarchy of relevant principles and procedures were more effective in solving problems. Complementing these two studies, Chi et al. [25] showed that success in application of science concepts was facilitated by amplifying student understanding of the hierarchical organization of science concepts, findings aligned with TIMSS.

The fourth exemplar is a series of field-experimental studies with upper elementary students by Romance and Vitale [7] in which they implemented an integrated instructional model, Science IDEAS, that combined science concepts, hands-on activities, reading comprehension, and writing for 2 hours daily (as a replacement for reading instruction). Teachers used core science concepts as curricular guidelines (see Figure 1) for identifying and organizing all instructional activities while also emphasizing students learning more about what had been learned.

In applying an ontological perspective to the preceding issues, it is important to keep in mind that many interdisciplinary controversies reflect semantic rather than substantive concerns. However, overall, the issue of how to address science education ontologically is of paradigmatic importance in that an interdisciplinary approach would imply a substantial advancement in knowledge and understanding of the science teaching-learning process.

References

1. Sowa, J.: Knowledge representation: Logical, philosophical, computational foundations. Brooks, NY (2000)
2. Davis, R., Schrobe, H.: Szolovita What is knowledge representation? *AI Magazine* 14(1), 17–33 (1993)
3. Kuhn, T.: The structure of scientific revolution. University of Chicago Press, Chicago (1996)
4. Vitale, M.R., Romance, N.R., Crawley, F.: Trends in science education research published in the *Journal of Research in Science Teaching: A longitudinal policy perspective*. Paper presented at the Annual Meeting of the National Association for Research in Science Teaching, Philadelphia, PA (2010)
5. Luger, G.F.: Artificial intelligence: Structures and strategies for complex problem solving. Addison-Wesley, NY
6. Vitale, M.R., Romance, N.R.: Concept mapping as a means for binding knowledge to effective content area instruction: An interdisciplinary perspective. In: Canas, A.J., Novak, J.D. (eds.) *Concept Maps: Theory, Methodology, and Technology*, pp. 112–119. San Jose, Costa Rica (2006)
7. Romance, N.R., Vitale, M.R.: Implementing an in-depth expanded science model in elementary schools: Multi-year findings, research issues, and policy implications. *International Journal of Science Education* 23, 373–404 (2001)

8. Bransford, J.D., Brown, A.L., Cocking, R.R.: How people learn. NAP, Washington (2000)
9. Anderson, J.R.: Automaticity and the ACT theory. *American Journal of Psychology* 105(2), 165–180 (1992)
10. Anderson, J.R.: Problem solving and learning. *American Psychologist* 48(1), 35–44 (1993)
11. Anderson, J.R.: ACT: A simple theory of complex cognition. *American Psychologist* 51(4), 335–365 (1996)
12. Cepeda, N.J., Coburn, N., Rohrer, D., Wixted, J.T., Mozer, M.C., Pashler, H.: Optimizing distributed practice: Theoretical analysis and practical applications. *Experimental Psychology* 56, 236–246 (2009)
13. Sidman, M.: Equivalence relations and the reinforcement contingency. *Journal of the Experimental Analysis of Behavior* 74, 127–146 (2000)
14. Vitale, M.R., Romance, N.R.: A knowledge-based framework for unifying content-area reading comprehension and reading comprehension strategies. In: McNamara, D. (ed.) *Reading Comprehension Strategies: Theory, Interventions, and Technologies*, pp. 75–103. Erlbaum, NY (2007)
15. Romance, N.R., Vitale, M.R.: How should children's alternative conceptions be considered in teaching and learning science concepts: Research-based perspectives. National Association for Research in Science Teaching, San Diego, CA (1998)
16. Duncan, R.G., Hmelo-Silver, C.E.: Learning progressions: Aligning curriculum, instruction, and assessment. *Journal of Research in Science Teaching* 46, 606–609 (2009)
17. Schmidt, W.H., et al.: A splintered vision: An investigation of U.S. science and mathematics education, vol. III. Kluwer Academic Publishers, Dordrecht (1997)
18. Novak, J., Gowin, D.B.: Learning how to learn. Cambridge University Press, Cambridge (1984)
19. Mintzes, J.J., Wandersee, J.H., Novak, J.D.: Teaching science for understanding: A human constructivist view. Academic Press, NY (1998)
20. Hofmeister, A.M., Engelmann, S., Carnine, D.: Developing and validating science education videodisks. *Journal of Research in Science Teaching* 26(8), 665–667 (1989)
21. Muthukrishna, N., Carnine, D., Grossen, B., et al.: Children's alternative frameworks: Should they be directly addressed in science. *Journal of Research in Science Teaching* 30(3), 233–248 (1993)
22. Vitale, M.R., Romance, N.R.: Using videodisk technology in an elementary science methods course to remediate science knowledge deficiencies and facilitate science teaching attitudes. *Journal of Research in Science Teaching* 29(9), 915–928 (1992)
23. Vosniadou, S.: Learning environments for representational growth and cognitive science. In: Vosniadou, S., DeCorte, E., Glaser, R., Mandl, H. (eds.) *International Perspectives on the Design of Technology-Supported Learning Environments*, Mahwah, pp. 13–24. Erlbaum, NJ (1996)
24. Dufresne, R.J., Gerance, W.J., et al.: Constraining novices to perform expert like problem analyses: Effects of schema acquisition. *The Journal of Learning Sciences* 2(3), 307–331 (1992)
25. Chi, M., Feltovich, L., Glaser, R.: Categorization and representation of physics problems by experts and novices. *Cognitive Science* 5, 121–152 (1981)

Author Index

- Agache, Alexandru 215
Akhgar, Babak 383
Alhabashneh, Obada 346
Amin, Saad 346
Amirjavid, Farzad 353
Andrews, Simon 50, 63, 394
Angelova, Galia 173, 298
Arabnia, Hamid R. 323
Aufaure, Marie-Aude 91
- Bahrami, Azita 315
Beckley, Russell 339
Bloodsworth, Peter 375
Bouchard, Bruno 353
Bouzouane, Abdenour 353
Bouzoubaa, Karim 159
Boytcheva, Svetla 298
Bridges, Shaun 309
- Cellier, Peggy 77
Charnois, Thierry 77
Coombs, Jeffrey 339
Cuvelier, Etienne 91
- Dau, Frithjof 1
Dedene, Guido 201
de la Rosa, Josep Lluís 104
Del Vescovo, Chiara 187
Dobre, Ciprian 215
Dobrocsi, Gábor 104
Ducassé, Mireille 77
- Elzinga, Paul 201
- Ferré, Sébastien 77
- G., Nagarjuna 402
Galitsky, Boris A. 104
Ghorab, M. Rami 366
Giumale, Cristian 215
- Haemmerlé, Ollivier 229
Hashemi, Ray 315
Hernandez, Nathalie 229
Hill, Richard 406
- Iqbal, Kashif 357
Iqbal, Rahat 346
- Jakobsen, David 118
James, Anne 346
- Kabbaj, Adil 159
Kazanci, Caner 323
Keeler, Mary 131
Kharatmal, Meena 402
Kumar, Neeraj 357
Kuznetsov, Sergei O. 104, 201
- Launders, Ivan 145
Lawless, Seamus 366
Love, David 304
Luper, David 323
- McClatchey, Richard 375
Mühlhäuser, Max 270
Mulwa, Catherine 366
Munir, Kamran 375
Muraru, Mihnea 215
- Nasri, Mohammed 159
Negreanu, Lorina 215
Nikolova, Ivelina 173
Nkambou, Roger 257
- O'Donnell, Eileen 366
Øhrstrøm, Peter 118
Ogaard, Kirk 331
- Parsia, Bijan 187
Paulheim, Heiko 270
Poelmans, Jonas 201
Polovina, Simon 63
Popovici, Matei 215
Pradel, Camille 229
Priss, Uta 243
- Rahmouni, Hanene Boussi 375
Reza, Hassan 331
Romance, Nancy R. 414
Rouane-Hacene, Mohamed 257
Rudolph, Sebastian 19

- Sattler, Ulrike 187
Schärfe, Henrik 118
Schiffel, Jeffrey A. 309
Schmidt, Benedikt 270
Schneider, Michael 19
Schramski, John 323
Sears, Les 315
Shadija, Dharmendra 406
Shah, Nazaraf 346
Shahmoradi, Mohammad Reza 383
Shamdasani, Jetendr 375
Sharp, Mary 366
Sowa, John F. 35
Stoitsev, Todor 270
Sugiyama, Mahito 284
Taghva, Kazem 339
Tcharaktchiev, Dimitar 298
Valtchev, Petko 257
Viaene, Stijn 201
Vitale, Michael R. 414
Wade, Vincent 366
Yamamoto, Akihiro 284