Florian Daniel
George A. Papadopoulos
Philippe Thiran (Eds.)

LNCS 8093

Mobile Web Information Systems

10th International Conference, MobiWIS 2013
Paphos, Cyprus, August 2013
Proceedings

Springer

Lecture Notes in Computer Science 8093

Florian Daniel   George A. Papadopoulos
Philippe Thiran (Eds.)

Mobile Web Information Systems

10th International Conference, MobiWIS 2013
Paphos, Cyprus, August 26-29, 2013
Proceedings

Springer

Volume Editors

Florian Daniel
University of Trento
Department of Information Engineering and Computer Science
38123 Povo, Italy
E-mail: daniel@disi.unitn.it

George A. Papadopoulos
University of Cyprus
Department of Computer Science
1 University Avenue, Nicosia, Cyprus
E-mail: george@cs.ucy.ac.cy

Philippe Thiran
University of Namur
Rue grandgagnage, 21, 5000 Namur, Belgium
and
Sirris
Bvd Reyers, 80, 1030 Brussels, Belgium
E-mail: philippe.thiran@fundp.ac.be

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-40275-3 e-ISBN 978-3-642-40276-0
DOI 10.1007/978-3-642-40276-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013944762

CR Subject Classification (1998): H.5, H.3, H.4, C.5.3, K.4, D.2, J.1

LNCS Sublibrary: SL 3 – Information Systems and Application,
incl. Internet/Web and HCI

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 collects the research articles, tool demonstrations, and the keynote speech presented at the 10th International Conference on Mobile Web Information Systems (MobiWIS 2013), held during August 26–28, 2013, in Paphos, Cyprus. The area of information systems is huge and builds on decades of research and development by both academia and industry, yet it is still growing in size and importance. The Internet celebrated its 30th birthday in 2013 and by now the world's largest computer network. The Web was invented by Tim Berners-Lee in 1989, and its public version went online only 20 years ago. Mobile web and information systems (WISs) are a specifically targeted sub-area of Web-based ISs, which particularly focuses on the use of mobile technologies and devices in the development of WISs. Although the trend toward mobile technologies is apparently a recent one (fueled by the spreading of powerful smart devices), MobiWIS is already celebrating 10 years of existence! As such, it has been following the continuous and steady growth that mobile WISs have been experiencing over the last few years and naturally proposes itself as the premier venue to present and discuss all aspects related to issues and solutions of mobile technologies when it comes to the engineering of mobile WISs.

MobiWIS aims to advance the state of the art in scientific and practical research on mobile Web and information systems. It provides a forum for researchers and practitioners from academia, industry, and the public sector to disseminate the latest research results and to share knowledge and experiences regarding tools, techniques, technologies, models, and methodologies that lead to better information and service provisioning in the mobile Web.

The conference features a set of carefully selected tracks for the submission of contributions, which focus on the particular challenges regarding mobile Web information systems. Specifically, MobiWIS 2013 featured the following tracks (next to a "general" track for papers not fitting any of the tracks):

- Mobile Web and Social Applications
- Middleware/SOA for Mobile Web Information Systems
- Mobile Web Searching
- Context- and Location-Aware Services
- Data Management in the Mobile Web
- Mobile Cloud Services
- Mobile Internet of Things
- Mobile Web Security, Trust and Privacy
- Mobile Networks, Protocols and Applications
- Mobile Commerce and Business Services
- HCI in Mobile Applications
- Industry Track

2013 was not only the 10th birthday of MobiWIS, but also the first time MobiWIS published its proceedings with Springer. We are glad Springer recognizes the quality of the conference and is interested to continue this collaboration in the years to come. We are confident the authors and readers of this volume will appreciate this collaboration as well. The volume collects 20 full research papers (42% acceptance rate) and four demonstration papers, which cover a varied set of topics related to the mobile WISs, such as mobile Web services, location-awareness, design and development, social computing and society, development infrastructures and services, SOA and trust, UI migration and human factors, and Web of Things and networks. In addition to these scientific contributions, the volume also contains the abstract of the keynote speech given by Volker Gruhn, Chair for Software Engineering at the University of Duisburg-Essen, Germany.

We would like to thank all authors for submitting their works to MobiWIS 2013 and for contributing to this volume and for presenting and discussing their papers during the conference. It is their work that is the very reason for conferences like MobiWIS to exist. Then, we would like to thank all the Program Committee members and external reviewers, who provided valuable and constructive feedback to the authors and important assessments to the Program Vice-Chairs and the Program Chairs for the selection of papers. We would like to thank Maristella Matera and Gustavo Rossi (Workshop Chairs), Joyce El Haddad and Tor-Morten Gronli (Publicity Chairs), Quang Z. (Michael) Sheng and Debmalya Biswas (International Liaison Chairs), and George Samaras (Awards Chairs) for their professional work. We would also like to express our gratitude to the keynote speaker Volker Gruhn for his availability and insights, and Muhammad Younas and Elhadi Shakshuki (General Vice-Chairs) for the assistance and help with the organization of the overall event.

August 2013

Florian Daniel
George A. Papadopoulos
Philippe Thiran

Organization

General Chair

George A. Papadopoulos — University of Cyprus, Cyprus

General Vice-Chairs

Muhammad Younas — Oxford Brookes University, UK
Elhadi Shakshuki — Acadia University, Canada

Program Chairs

Florian Daniel — University of Trento, Italy
Philippe Thiran — Sirris and University of Namur, Belgium

Program Vice-Chairs

Leandro Wives, Universidade Federal Do Rio Grande Do Sul, Brazil
Track Mobile Web and Social Applications
Zaki Malik, Wayne State University, USA
Track Middleware/SOA for Mobile Web Information Systems
Daqiang Zhang, Institute Telecom, France
Track Mobile Web Searching
Hong-Linh Truong, Vienna University of Technology, Austria
Track Context- and Location-Aware Services
Roberto De Virgilio, Universitá Roma Tre, Italy
Track Data Management in the Mobile Web
Claudio Geyer, Universidade Federal Do Rio Grande Do Sul, Brazil
Track Mobile Cloud Services
Federico M. Facca, CREATE-NET, Italy
Track Mobile Internet of Things
Sabrina De Capitani di Vimercati, Università degli Studi di Milano, Italy
Track Mobile Web Security, Trust and Privacy
Carlos A. Iglesias, Universidad Politecnica de Madrid, Spain
Track Mobile Networks, Protocols and Applications
Michael Weiss, Carleton University, Canada
Track Mobile Commerce and Business Services
Nikolay Mehandjiev, University of Manchester, UK
Track HCI in Mobile Applications
Ghita Kouadri, Diamond Light Source, UK

Industry Track
Muhammad Younas, Oxford Brookes University, UK
General Track

Workshop Chairs

Maristella Matera	Politecnico di Milano, Italy
Gustavo Rossi	University of La Plata, Argentina

Publicity Chairs

Joyce El Haddad	University of Paris Dauphine, France
Tor-Morten Gronli	Norwegian School of IT, Norway

International Liaison Chairs

Michael Sheng	University of Adelaide, Australia
Debmalya Biswas	Nokia Research, Lausanne, Switzerland

Awards Chair

George Samaras	University of Cyprus, Cyprus

Program Committee

Track Mobile Web and Social Applications

José Palazzo M. De Oliveira	Universidade Federal do Rio Grande do Sul, Brazil
Serge Garlatti	Telecom Bretagne, France
Karin Becker	Universidade Federal do Rio Grande do Sul, Brazil
Alejandro Fernandez	LIFIA - Universidad Nacional de La Plata, Argentine
Jérome Gensel	Laboratoire d'Informatique de Grenoble, France
Luiz Augusto Pizzato	University of Sydney, Australia
Rongxing Lu	University of Waterloo, Canada

Track Middleware/SOA for Mobile Web Information Systems

Brahim Medjahed	University of Michigan, Dearborn, USA
Mustafa Rafique	IBM, Ireland
Abderahmane Lakas	UAE University, United Arab Emirates

Jawwad Shamsi	FAST - National University, Pakistan
Salima Benbernou	Université Paris Descartes, France
Abdelkarim Erradi	Qatar University, Qatar
Abdelmounaam Rezgui	New Mexico Tech., New Mexico, USA
Qi Yu	Rochester Institute of Technology, NY, USA
Xumin Liu	Rochester Institute of Technology, NY, USA
Jiangang Ma	University of Adelaide, Australia

Track Mobile Web Searching

Vincent Gauthier	Telecom SudParis, Institut Mines Telecom, France
Hongyu Huang	Chongqing University, China
Vaskar Raychoudhury	India Institute of Technology at Rokeeie, India
Ioanna Rossaki	National Technology University at Athens, Greece
Gallacher Sarah	Heriot-Watt University, UK
Haoyi Xiong	Institue Telecom, France
Guoqiang Zhang	China Academic of Science, China
Jingyu Zhou	Shanghai Jiao Tong University, China
Rongbo Zhu	South-Central University For Nationalities, China

Track Context- and Location-Aware Services

Payam Barnaghi	University of Surrey, UK
Thanh Van Do	Telenor and Norwegian University of Science and Technology, Norway
Georgia Kapitsaki	University of Cyprus, Cyprus
Soo Dong Kim	Soongsil University, South Korea
Fei Li	Vienna University of Technology, Austria
Seng Loke	La Trobe University, Australia
Atif Manzoor	Trinity College Dublin, Ireland
Rene Meier	Lucerne University of Applied Sciences, Switzerland
Davy Preuveneers	K.U. Leuven, Belgium
Sita Ramakrishnan	Monash University, Australia

Track Data Management in the Mobile Web

Francesco Guerra	University of Modena and Reggio Emilia, Italy
Phuc Nguyen Van	Van Phòng Thành uy Thành pho Ho Chí Minh, Vietnam
Ismael Navas-Delgado	University of Malaga, Spain
Marco Brambilla	Politecnico di Milano, Italy
Dimitris Plexousakis	Institute of Computer Science, FORTH, Greece

Prateek Jain	Kno.e.sis, Wright State University, USA
Devis Bianchini	University of Brescia, Italy
James Geller	New Jersey Institute of Technology, USA
Matthias Klusch	DFKI, Germany
Stefano Montanelli	Universitá degli Studi di Milano, Italy
Flavius Frasincar	Erasmus University Rotterdam, The Netherlands
Andrea Cali	University of London, Birkbeck College, UK
Sebastian Link	The University of Auckland, New Zealand
Mirko Bronzi	Universitá Roma Tre, Italy
Christophe Guéret	Vrije Universiteit Amsterdam, The Netherlands
Tore Risch	Uppsala University, Finland
Disheng Qiu	Universitá Roma Tre, Italy
Roberto Mirizzi	Hewlett-Packard Laboratories, CA, USA
Simon Scerri	Digital Enterprise Research Institute (DERI), National University of Ireland, Ireland

Track Mobile Cloud Services

Markus Endler	Pontificia Universidade Católica do Rio de Janeiro (PUC-Rio), Brazil
Claudia Canali	University of Modena and Reggio Emilia, Italy
Jorge Sá Silva	University of Coimbra, Portugal
Didier Donsez	Université Grenoble 1, France
Luciana Arantes	Université Pierre et Marie Curie, France
Marcelo Pias	Globosense, UK
Claude Chaudet	Telecom ParisTech, France
Simon Duquennoy	SICS, Sweden
Pedro José	Marron University of Duisburg-Essen, Germany

Track Mobile Internet of Things

Alessandro Bassi	Bassi Consulting, France
Julien Bourgeois	UFC/FEMTO-ST UMR CNRS 6174, France
Francesco Niglia	Studio Francesco Niglia, Italy
Fernando Lyardet	SAP Research, Germany
Michael Marschollek	Peter L. Reichertz Institute for Medical Informatics, Germany
Massimo Mecella	Sapienza Università di Roma, Italy
Klaus Moessner	University of Surrey, UK
Constandinos Mavromoustakis	University of Nicosia, Cyprus
Anna Fensel	FTW Telecommunications Research Center Vienna, Austria
Vicente Pelechano	Universidad Politecnica de Valencia, Spain
Jukka Riekki	University of Oulu, Finland
Dumitru Roman	SINTEF/University of Oslo, Norway

Track Mobile Web Security and Trust and Privacy

Marco Casassa-Mont	HP Laboratories, UK
Elisa Costante	TU/e, Eindhoven, The Netherlands
Roberto Di Pietro	Universitá di Roma Tre, Italy
Joaquin Garcia-Alfaro	Telecom Bretagne, France
Giovanni Livraga	Universitá degli Studi di Milano, Italy
Kui Ren	Illinois Institute of Technology, USA
David G. Rosado	University of Castilla-La Mancha, Spain
Marianthi Theoharidou	AUEB, Greece
Vicenc Torra	IIIA-CSIC, Spain
Meng Yu	Virginia Commonwealth University, USA

Track Mobile Networks, Protocols and Applications

Paolo Nesi	University of Florence, Italy
Brian Lee	Software Research Institute, Ireland
Daniel Galindo	Telefónica I+D, Spain
Sven Abels	Ascora GmbH, Germany
Ignacio Soto	Universidad Carlos III, Spain
Sergio Castillo	Universidad Industrial de Santander, Colombia
Francisco Cano	Satimo, France
Zhefu Shi	Microsoft, USA
Ailisto Heikki	VTT, Finland
Marco Manso	Tekever, Portugal

Track Mobile Commerce and Business Services

Yuhong Yan	Concordia University, Canada
Jian Yang	Macquarie University, Australia
Morad Benyoucef	University of Ottawa, Canada
Dirk Schnelle-Walka	TU Darmstadt, Germany

Track HCI in Mobile Applications

Antonella de Angeli	University of Trento, Italy
Maria Francesca Costabile	Universitá degli Studi di Bari, Italy
Takahiro Hara	Technology Osaka University, Japan
Simon Harper	SCS, University of Manchester, UK
Maristella Matera	Politecnico di Milano, Italy
Anders Morch	InterMedia, Norway
Abdullah Namoun	Manchester Service Research, Manchester, UK
Eamonn O'Neill	University of Bath, UK
Fabio Paterno	National Research Council (CNR), Italy
Maria Da Graca Pimentel	University of Sao Paulo, Brazil

Alistair Sutcliffe	UCLIC, University College, London, UK
Shari Trewin	IBM Thomas J. Watson Research Center, USA

Industry Track

Thiab Taha	University of Georgia, USA
Domenico Cotroneo	Federico II University of Naples, Italy
Nanjangud C. Narendra	IBM Research, India
Arne Brring	52 North, Germany

General Track

Rachid Anane	Coventry University, UK
Irfan Awan	University of Bradford, UK
Soon Ae Chun	City University of New York, USA
Tomoya Enokido	Rissho University, Japan
George Ghinea	Brunel University, UK
Hamidah Ibrahim	University Putra Malaysia, Malaysia
Makoto Ikeda	Fukuoka Institute of Technology, Japan
Carsten Kleiner	Hannover University of Applied Sciences and Arts, Germany
Mourad Ouziri	Université Paris Descartes, France
Florin Pop	University Politehnica of Bucharest, Romania
Wenny Rahayu	La Trobe University, Australia
Indrakshi Ray	Colorado State University, USA
Yun Rui	Shanghai Advanced Research Institute, China
Alberto Sillitti	Free University of Bolzano, Italy
Geerish Suddul	University of Technology, Mauritius
Yi Yang	Carnegie Mellon University, USA
Jian Yu	Swinburne University of Technology, Australia
Sherali Zeadally	University of the District of Columbia, USA
Yiqiang Chen	Chinese Academic of Sciences, China

External Reviewers

Liang Chen	Rochester Institute of Technology, NY, USA
Danny Hughes	Technical University of Cluj-Napoca, Romania
Meriem Laifa	University of Ottawa, Canada
Denisa D. Rodila	Technical University of Cluj-Napoca, Romania
Carlos Oberdan Rolim	Universidade Federal Do Rio Grande Do Sul, Brazil
Sanjin Sehic	Vienna University of Technology, Austria

Table of Contents

Keynote Abstract

Mobile Web Services

Location-Awareness

Design and Development

Social Computing and Society

Development Infrastructures and Services

UI Migration and Human Factors

Web of Things, Network and Federation

Demonstrations

Tamed Agility in Developing Mobile Business Systems

Volker Gruhn and Matthias Book

paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen
Gerlingstr. 16, 45127 Essen, Germany
{volker.gruhn,matthias.book}@paluno.uni-due.de

Abstract. Mobile systems are determined by requirements which tend to change over time. Agile approaches seem to address this, but fail to provide reliable project plans, budget estimations and capacity forecasts. Thus, mobile enterprise applications demand for tamed agility, reconciling advantages of agile development and plan-driven approaches. In this paper, this tradeoff is addressed. The Interaction Room method is introduced as a low-tech method to support value-oriented development of mobile applications.

Keywords: Mobile systems, agile development, modeling, team interaction.

1 Introduction

Today, software engineers have at their disposal a multitude of notations to specify virtually all aspects of an information system, and tools to support considerable parts of the development process – and yet software projects in practice often run over time or budget, do not meet quality or functionality expectations, or are even aborted before completion. In most cases, the reason for such wasted efforts can be traced to communication problems. Curtis et al. identified the "thin spread of application domain knowledge" as a major factor contributing to project troubles already almost 25 years ago [1]. While method support has matured considerably since then, the complexity of today's information systems has multiplied as well. Especially in mobile information systems that enable the execution of business processes "on location", systems need to be finely tuned to the users' real-world work environment and the business and usability requirements mandated by it. The increasing flexibility and interconnection of business processes introduce whole new levels of complexity and thus an even higher need for business and technical stakeholder communication [2].

This is confirmed by our observations of mobile software development practice in a multitude of medium- and large-scale projects, mostly in the insurance, financial services and healthcare industry, over the past years: While domain experts and technical experts both strive for a joint understanding of the mobile information system that shall be built, more or less obvious barriers and negligence soon tend to impede communication. These observations suggest that better support for team members' communication and collaboration is required. However, most approaches to provide collaboration support revolve around development tools instead of addressing the underlying cognitive aspects [3].

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 1–8, 2013.

Agile process models have been introduced to ensure more frequent and consistent alignment of business and technical stakeholder positions through continuous feedback cycles [4]. However, while agile process models encourage (and actually depend on) intensive communication, they typically do not provide explicit operational support for it. Instead, we observe that the different background, expertise, culture and goals of different stakeholder groups (software engineers, domain experts, managers) often lead to poor communication and thus the following effects, which we believe to be the prime contributors to project risk:

- **Poor understanding/overview** of business domain requirements and technical design rationales: Domain experts tend to be sidelined once the initial requirements analysis process has been completed, with discussion shifting to the technical aspects (which especially in mobile systems can shape design decisions that affect the business quite significantly). This usually does not occur consciously, but implicitly due to the choice of more technical modeling notations, focus on technical design decisions, etc. However, without a consistent and continuous understanding of both business and technical aspects of the system, conflicts and errors become apparent only at late stages when they are costly to fix.
- **Negligence of value and effort drivers:** We often observe that teams focus on those aspects of a system that are well understood, since it is easy to produce voluminous specifications and functioning artifacts for them. However, this often keeps the team's attention on the more trivial aspects of a system, while a blind eye is turned (consciously or subconsciously) to the actual value and effort drivers – i.e. those components that hold particular value for the viability and effectiveness of the system, or those that demand higher effort due to their intrinsic complexity or special requirements. The negligence of the value and effort drivers in favor of boilerplate functionality generates an illusion of project progress that is dangerous because those system aspects that would merit most attention, and should shape the design of the rest of the system, are not addressed until project resources are more scarce, and possibly conflicting design decisions have already been established.

The above issues are related to a lack of understanding of project aspects that are not explicitly expressed in typical models of mobile information systems, but crucial to project success. In order to prevent these issues from growing unnoticed until they pose significant risks to a project, we aim to make them more visible and tangible for all stakeholders, and thus bring them into the focus of interaction before they can develop into problems.

2 The Interaction Room

In order to achieve this, we advocate the creation of so-called "interaction rooms" for complex software projects. An interaction room is a physical room that is outfitted to visualize and facilitate discussion of key aspects of an information system: Around a conference table, the walls of the room are covered with large sketches of models describing those system aspects that are most critical for project success. To intuitively create and manipulate these maps, we prefer the room's walls to be furnished with large magnetic whiteboards to draw and stick model fragments on. Each of the four walls is dedicated to a particular modeling perspective, as described below – however not displaying it in the dreaded fine-printed wallpaper-size completeness, but focusing on those aspects that are essential for orienting the team members' discussions, in order to keep the models clear and the approach pragmatic:

- **Process map:** One wall is dedicated to the dynamic aspects of the business domain, i.e. the relevant parts of the business processes that the system shall support. It provides a visual overview of the system's main functional requirements, and can thus serve as a reference for understanding the system and its context (including its mobile aspects), making high-level design decisions, prioritizing artifacts, etc.
- **Data map:** Another wall is dedicated to the static aspects of the business domain, i.e. the relevant parts of the entities, documents etc. that the processes work on. These models should initially just visualize the business relationships so as not to anticipate design decisions; however, these relationships may evolve to more closely reflect the technical data structures as the project progresses.

Complex software systems are typically not implemented on a clean slate – rather, a new system is usually replacing a legacy system that is integrated in an existing enterprise software landscape – especially when an existing system is "upgraded" to support mobile business process execution. While the previous two maps focus on the vision of the system being built, the next two focus on particular challenges of brown-field software development:

- **Migration map:** Replacing a legacy system requires careful planning of the transition from the old data structures to the new ones. One wall of the interaction room is therefore dedicated to visualizing the legacy data structures already in place, and mapping out their transitions to the new system's structures. This involves determining entities, attributes and relations that should be added, converted, removed, restructured etc. – operations that require thorough understanding of the business and technical aspects of both systems, and can be discussed here side by side.
- **Integration map:** Another wall is dedicated to visualizing the application landscape that the new system will be integrated in – this includes other in-house or external systems and services, and their interfaces, as well as the distribution of data entities across a mobile system. The map does not aim to be a detailed interface specification, but a guide to the communication and coordination requirements that the new system will have to satisfy.

Many software projects produce copious amounts of detailed specifications, some of which may describe trivial standard aspects, while others spell out important details of a system's static and dynamic properties. In either case, the sheer volume of information makes it difficult for team members from different backgrounds to keep an overview of the whole system, have a firm grasp of the larger dependencies between features, components and data structures, and notice the important details and pitfalls among the trivia. The process, data, migration and integration maps address this challenge by deliberately remaining incomplete, and instead showing only the most critical aspects of a system. This does not preclude business and technical experts from having more detailed, more complete models on their own desks, or temporarily bringing them into the room for discussion. The interaction room's maps, however, should not be treated as a complete specification, but as an overview that provides orientation on the project's goal, state, and interdependencies, and thus makes the more detailed, actual specifications accessible.

3 Identifying Value and Effort Drivers

Following the concept of value-based software engineering [5], we are striving to make the **value** that is inherent in an organization's business processes – and thus the value the mobile information system is expected to provide/support – explicit in the models of the system used in the engineering process. At the same time, we aim to highlight elements in whose implementation particular **effort** must be invested to address requirements or complexity that might otherwise be overlooked.

Rather than conceiving new types of models to express these value and effort drivers, we make them explicit on the interaction room's maps through graphical annotations that are independent of the notations used for the various maps. This has the advantage that stakeholders can use the same set of value and effort annotations in a wide variety of model types (e.g. business process models, data models, integration models, etc.) to express issues that are often of an overarching nature.

3.1 Value Annotations

Value annotations denote particularly important system aspects whose support must be ensured so that the benefit expected from the systems' features can actually materialize. Ignorance of value drivers means that a system will not be useful because its features cannot be used effectively in practice. Since system models typically focus on the features, but not the quality attributes expected of them, we let stakeholders denote these with the following value annotations:

Financial responsibility. Even if they are quite detailed, models of a business process or software component may not convey the amounts of money or other valuable assets are processed by them. Some components may perform calculations (e.g. for life insurance benefits or oil tanker reinsurance underwriting) of such volume that any errors – especially when accumulated over many transactions – can lead to significant financial risks for the organization and its clients. This annotation therefore indicates that extreme care must be taken in ensuring the component's correct implementation and testing in close cooperation with business experts.

Number of users. Aside from performance considerations, a high number of users imposes particularly high standards onto the quality of a component's user interface: A dialog that only a few people work with (e.g. the controlling department) does not need to have similarly fine-tuned usability, and must not be suitable for such a diverse audience, as e.g. the point-of-sale interface used by an insurance company's hundreds of field sales agents. This annotation indicates that even small improvements made in such a component can yield a significant efficiency gain in a lot of users' work.

Image sensitivity. Any interaction of an organization with outside parties – customers, suppliers, media etc. – affects the organization's image, i.e. its public perception. Even for interactions that are not deemed crucial in terms of the other value annotations, quick response times and professional presentation (e.g. of the online complaints handling interface or the invoice layout) can affect the company image positively. This annotation therefore indicates features that might seem to be candidates for low prioritization, but whose quality can have a significant "soft" impact on a company's success.

3.2 Effort Annotations

Effort annotations serve as warning signs for system components that should be expected to require more work than meets the eye, such as performance or security requirements, the need for compliance with legal regulations, etc.:

High reliability. Reliability covers aspects such as the continuous availability of components or data, the interruption-free processing of tasks, etc. It is especially crucial in distributed and mobile systems, where mobile workers may depend on certain functionality or data being available even if their network connection to the server breaks. This annotation therefore indicates that developers need to take extra precautions to ensure the availability of the annotated resources (application logic or data structures) under all circumstances, or make provisions for dealing with periods of temporary unavailability.

Resource demands. Beside components that obviously require significant computing resources (i.e., processing power and storage space), such as an insurance company's customer database, information systems may be exposed to resource demands that are harder to identify, especially when mobile components are involved, or when demand is not constantly present, but occurs only in more or less predictable peaks – example scenarios might include annual mailings of pension fund statements, but also unexpected peaks in insurance claim filings after a hailstorm. In these cases, processes or components that may look peripheral or trivial at first glance may actually require sophisticated scaling techniques. This annotation therefore not just indicates which components require large processing resources, but also where opportunities for mobile resource optimization or back-end resource scaling (e.g. through cloud computing) should be explored.

Security standards. Data privacy and integrity are important cross-cutting aspects in most business information systems. Some components, however, may be subject to particular security requirements, such as the need for digital signage of certain records to ensure non-repudiation, the need for anonymization of certain records before processing, or the enforcement of confidentiality of certain parts of records (in an insurance company, e.g., not all staff members that are allowed to access a client's contracts may also be allowed to see that client's medical history). This annotation highlights data structures for which such particular security mechanisms need to be implemented that are not part of the organization's overall security standards, and not obvious from the model itself.

Legal regulations. For some business processes, there may be legal prescriptions that govern how the process must be executed. As opposed to company policies that usually describe preferred best practices, legal policies often do not provide an immediate value for the organization, but are motivated by external interests such as market or consumer protection. This annotation therefore highlights processes or data structures that need to be implemented in particular, non-obvious ways in order to satisfy applicable legislation. It can also indicate areas that are known to become regulated in the future, even if the law's details are still under discussion, and thus serve as a warning that a certain degree of flexibility should be built into that system component.

3.3 Complexity and Uncertainty Annotations

In addition to the above annotations that highlight particular value or complexity inherent in designated processes, data structures or components, we use two further annotations to point out complexity and uncertainty of a more general nature:

Implementation complexity. To highlight highly domain- or technology-specific value or effort drivers that do not fall into one of the above-mentioned categories, this annotation can be attached to any model element, where it serves as a warning that implementation of this process, component or data structure will require more effort or needs to satisfy higher quality standards than immediately meets the eye. Examples might be complex integration with third-party services, historically evolved conventions for the proper formatting of legacy data structures, additional effort for the adaption of COBOL components, etc.

Uncertainty. In contrast to the previous annotation, which marks complexity that is already known to some project stakeholders, and shall be communicated to other stakeholders, this annotation highlights parts of a model that stakeholders feel they do not yet understand sufficiently. This uncertainty may be due to a lack of application domain knowledge or technical expertise, which needs to be remedied through more in-depth research and discussion.

4 Interaction Room in Practice

To illustrate how all the elements we have described work together in practice, Fig. 1 shows a process map from the example of a new claims processing component for an insurance company. The business process is sketched as a UML activity diagram on a relatively high abstraction level here, so business and technical experts can come to a joint understanding of the key claims management steps to be supported.

Here, the *appraise case including history* activity has been annotated with an "image sensitivity" icon, indicating that the details of the case appraisal procedure pose not just a technical question, but involve more far-reaching customer relationship implications. Similarly, the *calculate insurance benefit* activity is annotated with an "financial responsibility" icon. While the details of this activity seem to be well understood already (as it has a UML sub-activity indicator referring to an existing detail specification), the annotation indicates that particular care should be taken in implementing and testing this step, as calculation errors could have considerable financial consequences. While the models on the walls of an interaction room cannot accommodate sufficient detail to work out the solutions to these warnings *in situ*, the annotations ensure that they are considered by developers as the models are refined further.

We are currently applying the interaction room approach in projects with a large software development company, an insurance company, and a large bank. Managers and developers have been very open to the idea in each company, expressing that the room addresses a real issue that has been troubling them. In the case of a complex migration project at a large bank, we already observed that applying just a subset of the visualization concepts relevant to migration, combined with guidance by an interaction room "coach", yielded significantly better understanding between stakeholders.

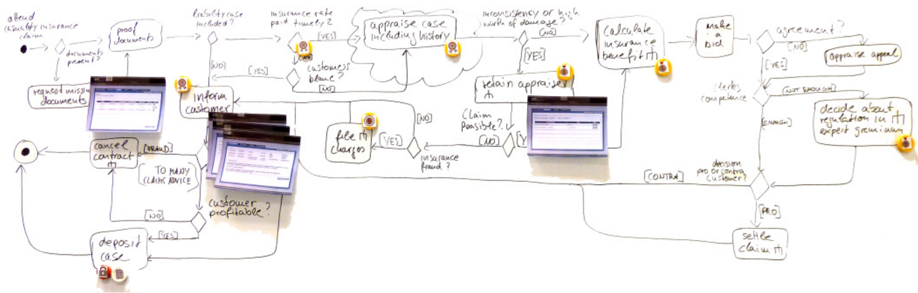


Fig. 1. Process map for an insurance claims processing component

5 Related Work

Joint Application Design [6] and explicit documentation of design rationale [7] were notable earlier approaches to make different project stakeholders more aware of what they are building, and why. The concept of "team rooms" is also proposed in agile methodologies such as Scrum [8], however, apart from planning tools such as Kanban boards and burndown charts, their walls are typically not dedicated to modeling system aspects. Although agile methods encourage interaction between stakeholders, they provide virtually no methods for focusing it on other aspects than progress.

A number of our annotations express system attributes that can also be formulated as non-functional requirements (NFRs). Adding NFRs to process models has been proposed before, e.g. in Control Cases [9] and PyBPMN [10], which both extend the Business Process Model and Notation (BPMN) with specification elements for attributes such as performance, reliability, security etc. While most of these approaches are formal extensions to specific languages, we deliberately propose a more informal set of annotations for pragmatic use with any language, whose aim is not a detailed and exhaustive NFR specification, but sparking discussions and raising awareness among project stakeholders (which may ultimately lead to laying down such specifications).

In the software engineering field, related works can also be found in the software visualization and program comprehension communities. However, most of these approaches focus on existing software, work closer to the source code level, and are therefore more exclusively geared toward developers. In contrast, we strive to foster joint understanding of the overall system by technical and business stakeholders.

6 Conclusions

In this paper, we introduced the Interaction Room as an approach to foster collaboration in heterogeneous teams, and to focus that collaboration on those aspects that are particularly critical for project success. Since the interaction room promotes the use of models to foster system understanding, but does not require those models to be detailed, precise and complete, our approach bridges agile and traditional paradigms in a way that might be considered as "tamed agility".

Even if the underlying complexity of a system and its business domain can never be completely overcome (especially when mobile business processes are involved, which may be exposed to numerous unforeseen situations), we expect that the use of an interaction room should help all team members to adopt a pragmatic perspective of a project that enables easier understanding of its details and better control of its risks.

References

1. Curtis, B., Krasner, H., Iscoe, N.: A field study of the software design process for large systems. Communications of the ACM 31(11), 1268–1287 (1988)
2. Kautz, K., Madsen, S., Nørbjerg, J.: Persistent problems and practices in information systems development. Inf. Sys. J. 17(3), 217–239 (2007)
3. Whitehead, J.: Collaboration in software engineering: A roadmap. In: Proc. Future of Software Engineering (FOSE 2007), pp. 214–225. IEEE Computer Society (2007)
4. Larman, C., Basili, V.: Iterative and incremental developments – a brief history. IEEE Computer 36(6), 47–56 (2003)
5. Boehm, B., Huang, L.: Value-based software engineering. ACM Software Engineering Notes 28, 4–10 (2003)
6. Davidson, E.: Joint application design (JAD) in practice. Journal of Systems and Software 45(3), 215–223 (1999)
7. Burge, J., Carroll, J., McCall, R., Mistrík, I.: Rationale-based software engineering. Springer (2008)
8. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall (2002)
9. Pavlovski, C.J., Zou, J.: Non-functional requirements in business process modeling. In: Proc. 5th Asia-Pacific Conference on Conceptual Modeling (APCCM 2008), pp. 103–112. Australian Computer Society (2008)
10. Bocciarelli, P., d'Ambrogio, A.: A BPMN extension for modeling non-functional properties of business processes. In: Proc. 2011 Symposium on Theory of Modeling & Simulation (TMS-DEVS 2011), pp. 160–168. Society for Computer Simulation (2011)

An Effective Approach to Parse SOAP Messages on Mobile Clients

Geerish Suddul[1], Nimal Nissanke[2], and Nawaz Mohamudally[1]

[1] University of Technology, Mauritius
Republic of Mauritius
{g.suddul,alimohamudally}@utm.intnet.mu
[2] London South Bank University
London

Abstract. This work presents an approach to address the bottleneck in SOAP processing on mobile devices during Web Service invocations. It takes the form of a unique SOAP response message parsing strategy which is implemented on the mobile device client. It consists of deriving the structure of a SOAP response message at the time the request is constructed. The derived structure is used as input in our special parser, named Structure Identification Parser (SIP), to extract the returned values. Our experiments show that our approach reduces SOAP response parsing time by around 85%, contributing to reduce the response time by nearly 48% when compared to the Apache Axis framework.

Keywords: XML Web Services, SOAP Parsing, Mobile Computing.

1 Introduction

Implementation of the Service Oriented Architecture [1] via Web Services is mostly based around three XML standards: SOAP [2], Web Services Description Language (WSDL) [3], and the Universal Description Discovery and Integration [4]. SOAP is the lingua franca for the Service Oriented Architecture (SOA) based Web Service communication, which allows disparate applications to share data over heterogeneous networks and protocols.

Nowadays, mobile devices form part of the distributed computing landscape, with continuous improvements in processing speed and storage capacity. Mobile network infrastructures are also gaining widespread deployment while at the same time becoming faster with reduced latencies. Such technological advancements have given rise to a new breed of mobile applications which require even more computing resources to accomplish tasks such as speech recognition, natural language processing, computer vision and graphics, machine learning, augmented reality, planning and decision making [5]. Such functionalities are usually provided to the mobile applications as Web APIs in the form of Web Services. Yet, not enough attention is being paid to the problems caused by some of the key technologies underpinning the development of Web Services in mobile environments, despite progress in other respects, e.g. general frameworks such as Schall et al.[6] and Sanchez-Nielsen et al. [7]. This

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 9–21, 2013.

applies, in particular, to XML-based SOAP, the data representation language used for expressing requests and responses in messages exchanged between Web Services and their clients. As SOAP was originally designed for communication in wired networks for fixed computers, it is typically verbose and may even seem to consist of redundancies.

SOAP requires more bytes for processing, as it uses textual representation to represent numbers which involves a costly conversion process and complicates the serialization and deserialization phase during parsing. SOAP also introduces superfluous tags including namespaces and attributes that surround data values being transferred [8]. This only exacerbates the problem associated with parsing time, as SOAP parsers have to evaluate every character in the file with respect to a series of conditions. Large SOAP messages also cause an increase in the transmission latency over the narrow bandwidth and intermittent connection in mobile networks. Therefore, such networks do not favor SOAP based interactions. From an application point of view, we consider that the parsing and transmission latencies with SOAP can deteriorate the response time of both data centric applications, which already suffer from read and write latencies and real time applications that have to deal with stringent time constraints. Increasing application response time can potentially contribute to making them unreliable. The overheads of XML processing, low capabilities of mobile devices and well-known limitations of wireless networks could in combination pose a strong challenge to proper integration of mobile devices with Web Services. The aim of this research is to optimize the performance of mobile devices when interacting with SOAP based Web Services.

2 Literature Review

The literature is rich on XML and SOAP optimization techniques for Grid computing and, to a less extent, on mobile computing. We review here a few relevant works dealing with optimization techniques and classify them under the following four main categories.

2.1 Serialization and Parsing APIs

This sub-section deal with work done towards optimizing one or more of the four XML processing stages described in [9]: parsing, accessing, modification and serialization. The work of [10] based on a schema-specific parsing resulted in the implementation of an optimized and efficient SOAP parser. Yet, the conversion of floating point numbers to XML textual representation accounts for nearly 90% of the parser's time in processing them and, therefore, the approach is highly resource intensive and is thus not appropriate for mobile devices. Other approaches such as that due to Head et al. [11] have focused on explicit schema usage but the main issue in these approaches is the cost of storage and reading time incurred by mobile devices. Abu-Ghazaleh, Lewis and Govindaraju [12] focus on serialization and deserialization of SOAP messages, rather than parsing, in an attempt to solve the issue in Chiu et al [10]. In their serialization model, every first invocation of a method call is serialized completely, while any subsequent following calls with the same structure, i.e. the same SOAP message, are not serialized again. Instead, the original pattern of the serialized output is used repeatedly for subsequent similar requests, thus reducing the overhead in the whole process. There seem to be a possible indication that serialization and deserialization are good candidates for performance improvement, but at the same time, it also becomes necessary to investigate the time taken to perform such type of processing on mobile devices. In contrast to the DOM and SAX parsing model, J. Zhang presents the Virtual Token Descriptor [13], also known as VTD-XML, which we consider as a memory efficient parser, since it eliminates performance overhead for object creation in DOM and SAX. Nevertheless, the streaming capabilities of VTD suffers from the same problem as DOM based parsers, that is, the data can only be accessed when parsing is complete. This makes the approach ineffective for resource constrained devices, where the memory could easily be exceeded with large SOAP response messages.

2.2 Binary XML Representations

When the verbosity of textual XML representation becomes a problem in certain applications, binary XML provides an alternative format that can enhance the performance. This format is considered as a form of XML representation that can be read and written without going through textual XML, in between the two operations, in the usual manner. Pericas-Geertsen [14] states that binary XML techniques can be divided into Infoset-based and schema-based variants. Infoset-based binary XML is considered as a self-contained alternative representation to an XML document, while schema-based formats rely on the presence of schema to create the representation.

Although the binary XML improves the performance for resource-constrained devices, in our view it is still not a sound solution, as we strongly believe that the performance issue is associated with the XML parsers and not entirely with XML itself. Binary XML can improve push and pull based parsing. Specific XML syntax features can be replaced or skipped; this can significantly reduce CPU cycles involved in parsing. However, all existing examples have used SAX based parsers, which provides forward-only stream based parsing, which implies several scanning of the document. As far as DOM based parsing is concerned, binary XML cannot bring much of an improvement, since most of the CPU cycles are spent on creating the in-memory tree representation of the document and not on tokenization.

2.3 File Compression Techniques

XML documents are different from normal text documents in that they have a structure, which is exploited by the XMill compressor [21]. It separates the structure of the document (tags) from the data (text), creating different groups with the related items and applying different compressors to each such group. This technique is proven to have better performance compared to GZip [22] on many kinds of XML data. Also, text document converted to XML and compressed with XMill can be smaller than the same text document compressed with GZip. Timing measurements indicate that XMill is, roughly speaking, as fast as GZip for both compression and decompression.

XML documents are very good candidates for compression since they consist of a relatively high degree of textual redundancy. While compression reduces their size, it also increases their processing time. This is mainly because compression adds another layer of processing for compression and decompression. This solution greatly reduces the size of the message for bandwidth consumption, but deteriorates the power consumption by the device by imposing a high level of processing. Also, when we consider XML messaging in terms of SOAP, the documents are small and contain more structural information instead of text. Hence, XML compression techniques so far cannot viably improve performance, especially in the case of resource-constrained mobile devices.

2.4 Hardware Acceleration

The progress made in hardware development also contributes to alleviate the issue of performance with XML processing through parallel processing or pipelining. For instance the appearance of multi-core CPU architectures opened a new avenue to performance improvement. Pan et al. [15] presents an approach on wired computers, where an XML document is divided into chunks of same size which are parsed in parallel. This approach is closely related to lazy parsing [16]. VTD in Zhang [13] is a good candidate for hardware implementation with its symmetric data structure with flat-array, for XML representation. The flat array allows efficient memory management before and after acceleration which is not possible with the callback and event strategies inbuilt in StAX and SAX. So far, there is no VTD hardware available, and thus it is not possible to give a concise evaluation of this approach. Hardware acceleration approach classically focuses on improving processing throughput but also requires complex special hardware. However, we believe that demands from modern mobile applications cannot be met solely by making mobile devices more powerful.

3 Proposed Solution

The idea behind our approach implies the generation of a client stub at design time that is to be integrated into the Web Service mobile client application to be used at runtime. The stub performs remote invocations, derives the response structure and uses SIP to parse the response. The main difference of our stub generation technique compared to existing techniques is its ability to collect additional information from the WSDL and SOAP schema, used to derive the structure of each operation invoked by the stub. The main steps involved in the process of executing a remote invocation at runtime, based on our structure derivation technique is shown in Fig.1. Once the stub gets hold of the method call, it derives the response structure in step 1 and then moves on to the actual invocation in step 2. A typical stub which conforms to the JSR-172 specification [17], for example, deals directly with the second step which creates the SOAP request message. We make extensive use of the information from the WSDL file to predict the structure of the SOAP response message, corresponding to the sent request. In step 3, the generated SOAP request is sent to the service, and the stub blocks while waiting for a response. Once the SOAP response is received, it is parsed using our Structure Identification Parsing (SIP) approach [23] in step 4. The parser is fed with the response structure derived from step 1 so as to avoid conventional parsing and use the more effective navigation technique. Once the returned values are extracted and deserialized in step 6, they are returned back to the calling application. The key step is the derivation of the response structure object, which contains all the necessary properties related to the structure of response SOAP message expected from the service.

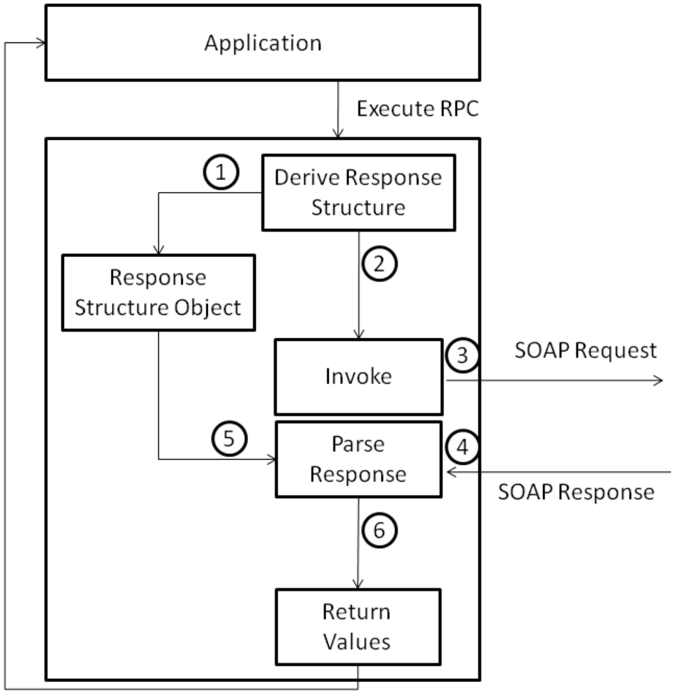


Fig. 1. RPC invocation from the client side, using structure derivation technique and SIP

3.1 Client Stub Generation Technique

The Stub can be generated manually or automatically by using the WSDL file which provides all the necessary information to call the service. Our stub generation approach is designed for use at development time by the application developer. It is based on the JSR-172 specification, which we extend to appropriately deal with data type mapping, element mapping and binding. We categorize the different properties extracted from the WSDL file by the stub generator as: Stub properties, RPC Methods & Properties, and Stub Utilities. Stub properties are used for invoking remote procedure calls on the Web Service. The main stub properties are: The target service endpoint address, available from wsdl:port element, the username and password for authentication and the SOAPAction_URI, which is the URI for the HTTP

(SOAPAction) header, available as a sub element from the wsdl:operation element. The stub utilities relate to all internal operations of the stub.

The RPC Methods & Properties represent all methods that can be invoked on the service by using the properties of those methods such as their qualified method name, input parameters and return types. The properties take the form of classes such as Type, Element and ComplexType to describe the input parameters and the return type of an invocation at runtime. The Type class is an enumeration of allowable parameter data types. Element represents additional properties of a parameter like name, minoccurs, maxoccurs, nillable, used to determine whether they hold multiple values. The ComplexType represents non primitive data types, and consists of a sequence of elements. The stub uses an operation class which is derived from the 'operation' element, to invoke a remote operation on the Web Service. Each operation contains a qualified name, as an instance of the QName class, and multiple instances of the Element class representing input and output parameters. These two parameters correspond to the message tags, as can be depicted in the WSDL file extract in Fig. 2. Each message represents a one-way communication from a computer to another. The figure shows a particular request and response case, where one message is used as an input and another message as an output. The complete description is available from the 'operation' element. The input message defines the input parameter which the service expects to receive and the output message defines the expected return value from the service. There can also be another message defined by the SOAP specification which characterizes error messages than can be returned by the service. This message is represented by a fault tag. Based on the partial WSDL extract in Fig.2, the stub generates the method 'calculateSquare' which takes one parameter 'num' which is of type integer and returns another integer value. Return values are encoded similar to input parameters using an object array. The stub decodes this object array and extracts the values for the calling application.

```
<message name="calculateSquare">
   <part name="parameters" element="tns:calculateSquare"/>
</message>
<message name="calculateSquareResponse">
   <part name="parameters" element="tns:calculateSquareResponse"/>
</message>

<portType name="NewMathService">
   <operation name="calculateSquare">
      <input  message="tns:calculateSquare"/>
      <output message="tns:calculateSquareResponse"/>
   </operation>
</portType>
```

Fig. 2. Partial extract the WSDL file for a Web Service which executes one operation that takes and one parameter and returns its square value

3.2 SOAP Response Message Structure Generation

The well defined rules of SOAP and the interface definition provided in the WSDL are also used to build the request message corresponding to an RPC. We exploit the same technique used in building SOAP requests to derive the structure of the SOAP response for operations which follow the request and response message execution pattern, as in Fig.2 above. As an example, below are their corresponding generated SOAP request and response messages at runtime with value of parameter ‘num’ for the method ‘calculateSquare’ set to 2. Fig.3 shows the SOAP request extract and Fig. 4 shows the response extract.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:calculateSquare xmlns:ns2="http://ac.utm.gs/">
      <num>2</num>
    </ns2:calculateSquare>
  </S:Body>
</S:Envelope>
```

Fig. 3. SOAP request message for the CalcuateSquare operation

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:calculateSquareResponse xmlns:ns2="http://ac.utm.gs/">
      <return>4</return>
    </ns2:calculateSquareResponse>
  </S:Body>
</S:Envelope>
```

Fig. 4. SOAP response for the CalculateSquare return value

The first three lines from the SOAP response message, Fig.4, represent the standard SOAP elements starting with the XML definition, the SOAP envelope and its body. The next line is the returned message with its namespace definition and the returned value. Lastly, a set of closing tags for the operation, body and envelope elements. All the required elements of this response, as well as the request, Fig.3, described in the WSDL file are extracted by the stub generator. Thus, it is possible to predict the structure of the response message at development time during stub generation. All properties common to the SOAP response message are available from our stub generator. Our parser (SIP) works by skipping over XML elements with predefined length. Thus properties of the SOAP response message having constant length, is calculated as shown in Table 1 below and is used with SIP. This information is then stored as properties within a class representing the SOAP structure.

Table 1. Properties of elements in a SOAP response message

Property	Element	Number of characters
Start Tag	< >	2
Close tag	< / >	3
SOAP Envelope Tag	< e n v e l o p e >	10
Namespace prefix	:	1
Namespace target	X M L N S =	6
XML declaration Tag	<? XML version="1.0" ?>	23
SOAP Body Tag	< b o d y >	6

All the remaining necessary parameters that constitute the response are the properties related to the operation, the SOAP envelope and SOAP body. Every operation requires a namespace prefix, a namespace target URI and qualified method name. Properties of the envelope are the envelope namespace prefix and target URI. The body element requires a body namespace prefix and target URI. Most of the namespace parameters are already well defined in the SOAP specification, and used by the stub to build a SOAP message. We show the predefined namespace definitions used by the SOAP envelope and body in Table 2. As for the operation namespace, it is often defined by the SOAP engine, on the server side, that processes the request and generates the response. The value for the namespace URI parameter is encoded as an attribute in the WSDL file root element named definitions and the attribute, targetNamespace, corresponds to the operation request and response messages, in Fig. 3 and 4. The prefix is the same used for the SOAP request message creation. The namespace value is extracted during the stub generation and is concatenated together with the name of the operation tags. All elements with a namespace are stored as qualified names, which consists of namespace URI, local part and prefix properties.

Table 2. Namespace definitions in SOAP response

Prefix	**Target URI**
ENV	http://schemas.xmlsoap.org/soap/envelope/
ENC	http://schemas.xmlsoap.org/soap/encoding/
XSD	http://www.w3.org/2001/XMLSchema
XSI	http://www.w3.org/2001/XMLSchema-instance

Therefore, when a remote operation is invoked, we combine the namespace information with the different mappings from the stub generation and the constant elements in a SOAP message, to derive the complete structure of the response message at the client side. This step is done before the message is actually sent on the network.

3.3 Structure Identification Parser

Our Structure Identification Parser (SIP) follows the pull parsing model and it provides a consistent and simple API. It occupies the smallest possible amount of memory on the mobile device and thus provides a minimal set of functionalities, which include: keeping track of parser state, the ability to move the next state and to read the event associated with the current state. Lexical Analysis is the core functionality of the general SIP approach and is responsible for converting the input stream into tokens which are used to produce events. The tokenization process is a Finite State Machine driven by input characters. Once tokens are identified, they are buffered and compiled into a block. The block represents a pair of parent tags and all their corresponding child elements. This is achieved with the conventional tokenization process, that is, identification of specific characters that raise specific, known, events in the parser. Conventional approach is used until another equivalent block is identified within the same XML document, after which the parser switches its tokenizer to a different mode which we define as navigation-based parsing. A cursor is used to navigate along the set of existing patterns, based on the length of each tag, calculated from the identified pattern. It therefore skips a set of computationally intensive token creation processes. SIP works well when XML document is in the canonical form [18] and consists of repeating patterns in its structure, such as with RSS files [19] and SOAP responses with arrays. Streaming support is provided, which implies that parsing starts as soon as the first input is ready. Although namespace support reduces parsing performance by introducing more validating conditions and prefix maintenance, it is unavoidable as it can be found in almost all of existing XML grammars nowadays.

As SIP is designed to skip known structural elements in an XML file, we adapt it to use the derived structure of the SOAP response. Thus, the original SIP parser avoids structure identification as it is fed with a pre-identified structure which it uses to extract the values based on navigation by length. This implies a SOAP navigation wrapper class on top of the existing structure parser, which manipulates the tokenizer's cursor and moves directly to the required text node to extract the value. This class holds the length of each specific structural elements corresponding to the response message. Extraction of text node values is based on conventional tokenization until the end delimiter is found. We integrate this modified version of the SIP parser into the Apache Axis client side framework, which we use as a prototype to parse response messages.

4 Experiments

We design and run a set of experiments to compare the performance of SOAP processing with SIP, which has been enhanced with the response message structure derivation technique. We compare the results with the conventional approach provided by the Apache Axis Framework, which uses the inbuilt kXML parser [20]. We implement the SIP parser prototype with the SOAP structure derivation technique, in J2ME. Device characteristics are described in the table 3 below. We run the experiments on the WiFi network to avoid unpredictable network latencies and packet loss in cellular networks.

Table 3. Characteristic of Devices used in experiments

Name	Role	CPU	Memory	Operating System
5800N	Client	ARM-11 434 MHz	128 MB (Internal)	Symbian v 9.4, S60 v 5.0
HPS	Server	Intel Core 2 Duo T6570 2.10 GHz	2.90 GB RAM	Windows XP, SP3

The main experiment consists of invoking simple Web Service operations which return array of strings, integers and doubles. The client is defined to send SOAP requests with a small payload which generate responses with large payload. The first request calls for an array containing 1000 elements by passing a simple integer in the operation, representing the size of the array. Each request is performed 100 times and the average values of all parameters are recorded. In each experiment, the minimum size of the returned array is 1000 and the maximum size is 10000, with intervals of 1000 elements.

In Table 4, we compare the parsing time of kXML and SIP across the three data types. SIP brings about a reduction of around 86% in the parsing time for double and integer data-types. We notice a reduction of nearly 81% for parsing with string data-type. On average, SIP parsing takes around 85% less time than the kXML parser, across the three data types.

Table 4. Comparison of parsing time (milliseconds) for SIP and kXML

		Double	Integer	String
SIP	Mean Parsing Time (ms)	355.90	301.10	245.70
	Std Dev	184.28	179.15	164.40
kXML	Mean Parsing Time (ms)	2481.70	2114.90	1285.30
	Std Dev	1403.10	1187.75	725.07

Table 5 shows the mean response time of the SIP against kXML for the three different data types. On average, SIP provides a reduction of 48% in response time, across all data types when compared with the Apache Axis kXML.

Table 5. Comparison of response time (milliseconds) to process SOAP messages with SIP and kXML

		Double	**Integer**	**String**
SIP	Mean Response Time (ms)	1644.30	1347.00	1477.10
	Std Dev	812.12	611.36	773.38
kXML	Mean Response Time (ms)	3238.80	2743.00	2521.20
	Std Dev	1750.64	1438.63	1344.58

It is clear from our measurements that SOAP processing with the Structure Identification Parser (SIP) provides a significant gain when compared to the kXML parser embedded in the Apache Axis framework. SIP brings about an overall reduction in the response time for SOAP invocations, though it does not explicitly deal with message size reduction. Its unique parsing technique is based on the SOAP structure prediction through the WSDL file and therefore contributes to the reduction of processing on the resource-constrained mobile devices.

4.1 Limitations

Our SOAP response derivation and parsing technique has been applied to WSDL files of type document/literal style. Also, the parser supports only W3C defined namespaces and their prefixes, limiting its scope to servers respecting the W3C strict definition of SOAP and WSDL. Our experiments also concentrate on measuring the processing time which consists of encoding the request message, sending the message, waiting for the server side processing and decoding the response message and interpreting the results on the client. We do not explicitly measure CPU processing cycles and battery life.

5 Conclusion and Future Works

We define in this paper an approach to demonstrate that mobile devices can be full legitimate participants in Web Service interactions and, in general, distributed computing, without necessarily relying on better hardware support. Parsing, although considered as a costly process, can be performed more efficiently on mobile devices. As SOAP does not necessarily have repeating structures, our structure tokenization model, SIP, has been adapted to use the SOAP structure prediction technique. It exploits the WSDL file which consists of all the necessary information required to construct a SOAP response message together with the request, before the RPC invocation on the service. SIP uses this generated structure to tokenize and therefore parse the SOAP response by avoiding conventional parsing. Our experiments demonstrate that this approach brings about some degree of performance improvement as opposed to the existing Apache Axis SOAP framework. We believe that SIP provides a rather straightforward solution which can be easily integrated into existing SOAP frameworks for mobile devices. There are practically no barriers to its adoption, and it can be deployed on different platforms as a library with an API for the programmer.

Two main areas of improvement have been identified as future works related to the proposed approach. The first consists of identifying a more appropriate serialization and deserialization technique for the parsing process. From our experiments we notice that double data types take more time to parse than integers and string. We attribute this result to the effect of deserialization, which is the process of converting the text based SOAP values into the in-memory structure representing the data type. Our parsing technique does not explicitly deal with different deserialization technique, but merely use inbuilt data type conversion of the programming language. We can therefore reasonably assume that most of the processing overhead for SIP is due to deserialization. Secondly, we also identify the need to provide asynchronous message execution patterns for mobile clients, which are meant to receive notifications anywhere and anytime. We propose to investigate the possibility of integrating asynchronous messaging support by exploiting the server side components with HTTP. This is not straightforward as the HTTP protocol is mostly defined for short-lived execution patterns, and asynchronous requests enable a variety of complex communication strategies which varies from simple request-response patterns to callback and polling.

References

1. MacKenzie, M.C., Laskey, K., McCabe, F., Brown, P., Metz, R.: OASIS Specification: Reference Model for Service Oriented Architecture, version 1.0. Organisation for the Advancement of Structured Information Standards (2006), http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf
2. W3C. W3C Recommendation-SOAP Version 1.2 Part 1: Messaging Framework (2007), http://www.w3.org/TR/soap12-part1/
3. W3C. W3C Recommendation-Web Services Description Language (WSDL) Version 1.1. (2001b), http://www.w3.org/TR/wsdl
4. OASIS. UDDI Version 2.04 API Specification. Advancing Open Standards for the Information Society (2002), http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm
5. Bhal, P., Han, R.Y., Li, L.E., Satyanarayanan, M.: Advancing the State of Mobile Cloud Computing. In: MCS 2012 Proc.of the Third ACM Workshop on Mobile Cloud Computing and Servicespages, pp. 21–28 (2012)
6. Schall, D., Gombotz, R., Dorn, C., Dustdar, S.: Human Interactions in Dynamic Environments through Mobile Web Services. In: IEEE International Conference on Web Services, ICWS 2007, vol. 9(13), pp. 912–919 (2007)
7. Sanchez-Nielsen, E., Martin-Ruiz, S., Rodriguez-Pedrianes, J.: An open and dynamical service oriented architecture for supporting mobile services. In: Proceedings of the 6th International Conference on Web Engineering, pp. 121–128 (2006)
8. Apt, N., Deutsch, K., Jain, R.: Wireless SOAP: optimizations for mobile wireless Web Services. In: Proceedings of 14th ACM International Conference on World Wide Web, pp. 1178–1179 (2005)

9. Lam, T.C., Ding, J.J., Liu, J.: XML Document Parsing: Operational and Performance Characteristics. IEEE Computer 41(9), 30–37 (2008)
10. Chiu, K., Govindaraju, M., Bramley, R.: Investigating the limits of SOAP performance for scientific computing. In: 11th IEEE Symposium on High Performance Distributed Computing, pp. 246–254 (2002)
11. Head, M.R., Govindaraju, M., Engelen, R.V., Zhang, W.: Benchmarking XML Processors for Applications in Grid Web Services. In: Proceedings of the International Conference on Web Services, ICWS, pp. 346–352 (2006)
12. Abu-Ghazaleh, N., Lewis, M.J., Govindaraju, M.: Differential deserialization for optimized SOAP performance. In: Proceedings of the ACM/IEEE Conference on Supercomputting, pp. 21–31 (2005)
13. Zhang, J.: SOAP Processing: A Non-extractive Approach. In: Zhang, L.-J., Jeckle, M. (eds.) ECOWS 2004. LNCS, vol. 3250, pp. 152–167. Springer, Heidelberg (2004)
14. Pericas-Geertsen, S.: Binary interchange of XML Infosets. In: XML Conference and Exposition (2003)
15. Pan, Y., Lu, W., Zhang, Y., Chiu, K.: A Parallel Approach to XML Parsing. In: Proceedings of 7th IEEE/ACM International Conference of Grid Computing, pp. 223–230 (2006)
16. Takase, T., Tajima, K.: Lazy XML Parsing/Serialization Based on Literal and DOM Hybrid Representation. In: Proceedings of the 2008 International Conference on Web Service, pp. 505–512 (2008)
17. Sun. JSR-000172 J2ME Web Services Specification, version 2.1. Sun Microsystems Inc. (2004b), http://jcp.org/aboutJava/communityprocess/final/jsr172/index.html
18. W3C. W3C Recommendation-Canonical XML version 1.0. World Wide Web Consortium (2001a), http://www.w3.org/TR/xml-c14n
19. RSS Advisory Board. RSS 2.0 specification, RSS Advisory Board (2009), http://www.rssboard.org/rss-specification
20. Haustien, S.: kXMl verion 1.0 (2001), http://kxml.sourceforge.net/index.shtml
21. Liefke, H.: XMill: An Efficient Compressor for XML Data. In: Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 153–164 (2000)
22. Deutsch, L.P.: RFC 1952: GZIP File Format Specification Version 4.3. IETF (1996b), http://www.gzip.org/zlib/rfc-gzip.html
23. Suddul, G.: Optimising Web Service Invocation for Mobile Devices. Ph.D. Dissertation, BISE, University of Technology Mauritius, Pointe-aux-Sables, Mauritius (2012)

An Efficient Interaction Framework for Mobile Web Services

Abdullah Almuaibid and Xumin Liu

Department of Computer Science, Rochester Institute of Technology, USA
aaa9722@rit.edu, xl@cs.rit.edu

Abstract. The increased number of smart phone users requesting information from the Internet has resulted in times when the *average access time* (the time a user must wait to download his/her requested service) is quite lengthy.The use of the wireless carrier's broadcast channels to transmit mobile web services has been proven to be an economical way to handle the increased volume of users. However, the existing approaches that minimize the average access time fail to take into account the dynamic popularities of web services when design the allocation of service data among broadcast channels. This paper presents an efficient interaction framework that provides guidance of optimizing service data allocations among multiple broadcast channels. This framework incorporates both current usage data (*i.e.,* popularity over specified time period) and the size of service data for channel allocation, thus resulting in the lowest average access time. The effectiveness and efficiency of the proposed solution have been demonstrated in the conducted experimental study.

Keywords: Mobile web services, broadcast, Greedy, Dynamic Programming.

1 Introduction

The last few years have witnessed an explosion of activities around mobile computing, including mobile communication, mobile devices, and mobile applications. Portable mobile devices, such as smartphones and tablet computers, play an important role in people's life and have achieved widely spread popularity. With the support of various application development environments, such as AdobeAIR, Android, Aqua, and BatteryTech, a large and ever-increasing number of mobile applications have been developed. Examples of such applications include educational tools, weather channels, social tools, inventory checkers, bill reminders, and personal finance helpers. Web services have been intensively used to facilitate the development of these applications running on resource-constraint devices. Such web services, which are consumed in a mobile device, are called as *Mobile Web Services* (MWSs).

Broadcasting is considered as an effective approach to delivering mobile web services to a large group of mobile users. It facilitates the exchange of web services among mobile devices and the wireless network and due to its scalability,

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 22–34, 2013.

the number of users does not affect the response time for retrieving the information. Another advantage is the fact that broadcast does not consume significant amounts of power on the client side [1–4]. In general a broadcast mobile web service delivery infrastructure consists of three parts: *Service Providers, Wireless Carriers* and *Mobile Users* [2]. Service providers develop their own mobile web services and publish them to wireless carriers (*e.g.,* AT&T, T-Mobile and Sprint). The wireless carriers disseminate the service descriptions and service data to their mobile device users. Service users can request for service data based on the service description via the wireless carriers.

There are several works proposed to design an efficient broadcasting framework for efficient service delivery[1, 2, 5]. They focus on minimizing *average access time*, i.e., the waiting time that a user needs to take to receive the response after sending a request, and *tuning time*, i.e., the period that a mobile device needs to stay active in order to listen to the channel and download the desirable data. The main idea is to use index channels containing *service index records* and *data index records.* A service index record contains the information including a service key, the id of the broadcasting channel, and the location of the service description in that channel. A data index record contains the information including a data set name or key, the id of the broadcasting channel, and the location of the data set in that channel. Service users can first listen to this index channel, retrieve the channel id and location of desired service/data, directly listen to the corresponding channel, precompute the waiting time, and only stay active when the desirable service/data arrives. The optimization is achieved by using efficient indexing mechanisms (such as B trees or Hash tables) to design the index channels. There are two major limitations of the existing work. First, it assumes that all services have the same possibilities of being requested. However, it is not the case in many real world scenarios. Some services might be more popular than others. Meanwhile, the popularity of a service may change over the time too. For example, hotel services may be requested more frequently in holiday seasons than in other time. The frequency of a service being requested should be considered when evaluating the average access time or tuning time. Second, it assumes that the allocation of service data among channels is static, which should be changed over the time to achieve the best performance in a long run.

In this paper, we propose an efficient framework for delivering mobile web services in a broadcasting environment. We mainly focus on minimizing average access time since the optimization on running time can be done in the similar way. Our contributions are summarized as following:

– We propose the measurements to evaluate the overall performance of service delivery in a broadcasting environment. We propose to compute the average access time for a single channel as well as the one for multiple channels. We consider both request frequency of each service and the length of the service data. /The measurements are used as the guidance for optimization.
– We describe and compare three optimization approaches, brute-force, greedy, and dynamic programming methods, to optimize the allocation of services among multiple channels, i.e., minimizing the multichannel average access

time. As the conducted simulation study suggests, a greedy algorithm can achieve an approximately optimal result with much better performance than a dynamic programming approach.

The remainder of this paper is organized as follows. Section 2 describes an interaction framework we propose for delivering mobile web services in a broadcasting environment. Section 3 formally defines the optimization problem. Section 4 describes three optimization solutions, including brute force, greedy, and dynamic programming. Section 5 presents our experimental study results. Section 4 describes and compares three optimization solutions. Section 6 discusses the representative related works and differentiate them with our work. Section 7 concludes the paper.

2 An Interaction Framework for Mobile Web Services

In this Section, we present a framework for mobile web services in a broadcasting environment, which supports to dynamically allocate web services into channels based on their popularities. The intuitive idea of the allocation is that, the more popular a web service is, the shorter cycle the channel where the service data is in should have. We also acknowledge the dynamic nature of a service's popularity. Therefore, the framework periodically examines the request frequencies of each service and optimizes the channel allocations accordingly.

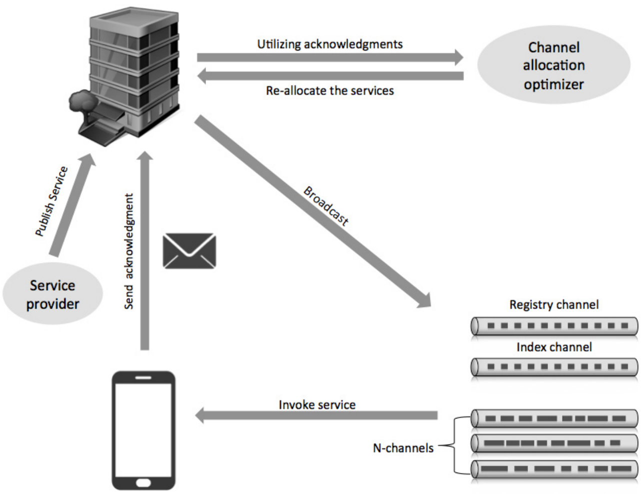
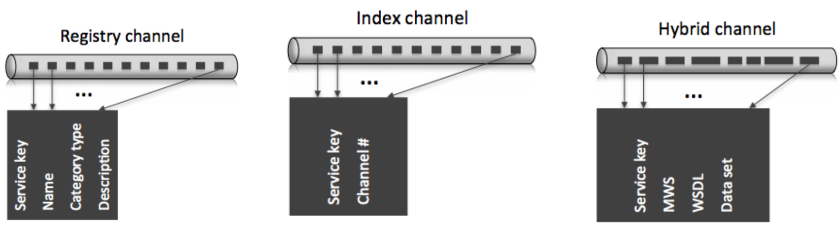


Fig. 1. Mobile web service reference model

As shown in Figure 1, this proposed architecture consists of two main components. One is the *mobile application*, which, among other things, facilitates the user's ability to explore the available mobile web services. Additionally, it allows the mobile user to communicate automatically with the wireless carrier's broadcast server. This line of communication is necessary because, as we know, the broadcast channels themselves can only communicate in one direction. Therefore, once the mobile user downloads his/her desired mobile web service, the mobile application will send a *miniscule acknowledgement* directly to the broadcast server so that the user's choice will be recorded. (It is important to note that this acknowledgement is transmitted as anonymous data, as knowing which mobile user is located where and what mobile web server he/she is using/requesting could be considered an invasion of privacy.) These acknowledgements are being tabulated for the purpose of determining the popularity and frequency of each web site's usage. The data will be used in the calculations which determine the broadcast channel scheduling.

The other main architectural component is the *broadcast channel configuration*. It consists of three core channels: a Registry channel, an Index channel, and a Hybrid channel.



Registry channel Index channel Hybrid channel

Fig. 2. Core channels

The following are descriptions of each type of main channel:

1. *Registry Channel*: This channel (Figure 2) functions like a menu in that it is a listing of all the mobile web services that are available in the geographic area in which the mobile device is located. When first entering a region, the mobile device's mobile web services application will download and store the contents of the registry file, which will be used as a reference when looking for mobile web services in this vicinity.
2. *Index channel*: This channel (Figure 2) provides a listing that informs the mobile device user where its desired mobile web service will be broadcasted (*i.e.,* in which of the many hybrid broadcast channels).
3. *Hybrid channel*: The "hybrid" channel (Figure 2) is so named because it transmits each mobile web service with its associated service key, WSDL, and corresponding data. Similar to the index channel, each record begins with the service key identifying number. The "combined" construction of the hybrid channel also reduces the tuning time and average access time

from what is needed when separate channels are used for the broadcast of the various mobile web services' components (as in [1, 2]).

The contents of the hybrid channels are dynamically updated (skewed) based on two factors/thresholds:

(a) *Period of time (t)* e.g., *3 hours and*;
(b) *Pre-defined number of requests/user acknowledgments, (r)* e.g., *1000*

These two attributes determine the most highly demanded mobile services and assign them to the hybrid broadcast channels based on that, with the result being that the most popular will be in less crowded channels.

The proposed architecture, with its two main components, plays an important role in reducing the number of ticks (*i.e.,* the time measurement for each "bucket," where each mobile web service could occupy several buckets) the user needs to connect with the required broadcast channels [6]. Hence, reducing the average access time experienced by the user before the desired mobile web service is available for download.

3 Optimization Problem Statement

3.1 Single Channel Average Access Time

In a single channel, one or more mobile web services could be there. We use S_i to denote a MWS in the channel, where $0 < i \leq m$ and m is the number of total number of the services in the channel. Let L_i be the length of the i-th MWS data, i.e., the number of buckets it occupied in the channel. Let W_i be the weight of the i-th MWS in the channel, determined by the number of *acknowledgments* generated by mobile users who accessed to that particular service, i.e., $\sum_{i \leq m} W_i = 1$. The total length of the broadcasting cycle in the channel is denoted as T. The single channel average access time refers to the duration a user needs to take to download the data of any service in the channel. It is computed as following.

$$\sum_{i=1}^{m}(L_i + \frac{T-1}{2}) \times W_i \tag{1}$$

Taking into account that mobile web services are cyclically broadcasted into the channel, this equation (1) calculates the average access time by adding together the "best-case scenario" of accessing a mobile web service and the "worst-case scenario" of accessing the same mobile web service and then calculating the average.

3.2 Multichannel Average Access Time

Suppose that there are K broadcasting channels, each is noted as Ch_i with a range of $1 \leqslant i \leqslant K$. C_i is the number of mobile web services in the i-th channel. Let S_{ij} be the j-th MWS in i-th channel, L_{ij} be the length of the service, T_i

be the length of the cycle of i-th channel, W_{ij} be the request popularity of the service, i.e., $\sum_{i\leq K,j\leq C_i} W_{ij} = 1$. The multichannel average access time refers to the duration that a user needs to take to access any service in any channel. It is computed as following.

$$\sum_{i=1}^{K}(\sum_{j=1}^{C_i}(L_{ij} + \frac{T_i - 1}{2}) \times W_{ij}) \tag{2}$$

3.3 Methodology

In general, our two-component architecture (*i.e.,* mobile application and broadcast channel configuration) plays an important role in reducing the *average access time* experienced by the user (*i.e.,* before his/her desired mobile services can be accessed). Delving into the details, one will find a broadcast channel allocation process that is based on real-time requests (popularity) of mobile web services, which are accounted for by the accumulation of user 'acknowledgements.' To this continuously updating scheduling procedure, the addition of a time threshold has been incorporated. This feature permits the distribution of mobile web services to be based on/limited to a specific timeframe. When the specified time has elapsed:

1. the "counters" are re-set,
2. the current popularity can then become "visible" again, and
3. the allocation is updated.

The "period of time" threshold prevents the possibility that a mobile web service will be given priority based on out-of-date accessing data.

Therefore, the problem is stated as: *Given M mobile web services and N broadcasting channels, find the optimal allocation of services among the channels to achieve the minimal multichannel average access time.*

4 Service Allocation Optimization

In this section, we study and compare three types of optimization methods to solve the problem stated in Section 3, in terms of their efficiency and effectiveness of them.

4.1 Brute Force

The idea of the brute force approach is straight ward, i.e., enumerating all possible allocations, computing the multichannel average time for each of them, and pick the best one. It is effective as the optimal allocation can always be achieved. However, it is computational expensive, which is explained using examples as following.

Example 1 configuration:

- 3 hybrid broadcast channels
- 5 mobile web services

As none of the broadcast channels should be left empty (*i.e.,* a minimum of one mobile web service in each channel), only two basic distribution formulas are possible: 3/1/1 with a total of 10 combinations and 1/2/2 with a total of 15 combinations. As one can see this relatively small-sized example has produced a grand total of 25 mobile web services allocation permutations which, according to our process, will each have to be tested to determine which is the most efficient mobile web services distribution.

In the second example (see below), we increase the number of available mobile web services to six to see what effect this may have on the number of possible allocation combinations. The number of available hybrid broadcast channels remains at three. The six mobile web services now generate three possible basic distribution formulas: 4/1/1, 3/2/1, and 2/2/2.

A 4/1/1 allocation is represented in the formula below:

$$\frac{\binom{6}{4} \times \binom{2}{1} \times \binom{1}{1}}{2} = 15 \tag{3}$$

Total: 15 combinations

A 3/2/1 allocation is represented in the formula below:

$$\binom{6}{3} \times \binom{3}{2} \times \binom{1}{1} = 60 \tag{4}$$

Total: 60 combinations

A 2/2/2 allocation is represented by the formula below:

$$\frac{\binom{6}{2} \times \binom{4}{2} \times \binom{2}{2}}{3} = 30 \tag{5}$$

Total: 30 combinations

With this second example, we can see that changing just one of the factors slightly (*i.e.,* number of MWSs) results in an increase to 105 allocation permutations (vs. the 25 combinations in example number 1). In another word, the computational complexity is increased in an exponential way. Therefore, the Brute Force algorithm-based allocation scheme is an inefficient methodology for our purposes.

4.2 Greedy

Our second testing scenario involved the use of a Greedy algorithm. As our starting point, we began with the Greedy algorithm presented in [6]. Since our data was significantly different than theirs, where each data set has the same size,

Algorithm 1. Greedy Method

```
1: Parameters:
2: listOfMWS: an ArrayList of input mobile service data
3: numberOfChannels: number of channels
4: numberOfCurrentChannels: initial as 0
5: Sorts all available mobile web services (MWS) (found in ArrayList listOfMWS)
   based on weight and length.
6: while numberOfCurrentChannels < numberOfChannels do
7:    for each channel Ch_i do
8:       for each service in the channel do
9:          compute the SAATs if the channel is splitted after this service into two
            channels
10:      end for
11:      stops if SAAT decreases
12:      compute the MAAT for the best splitting in this channel
13:   end for
14:   choose the splitting that yields the best MAAT
15:   numberOfCurrentChannels += 1
16: end while
```

we modified the measurement to fit it into our scenario. The revised algorithm is described as following.

As described in Algorithm 1, the process starts with one channel. It sorts the MWS based on its request frequency and and length. The intuition behind the sorting is that, the more popularity a service or the longer its data is, the less number of services should be assigned to the channel. Here we assigned the weights, 0.75 and 0.25, to the popularity and length of a service, respectively, when sorting them. The weights are also adjustable.

For each current channel, the algorithm finds the best partition of it to yield the best single channel average times after splitting the channel into two channels (Lines 8-10). Suppose there are N services in the channel, there would be $N-1$ possible partitions. This process starts from the first service in the first channel and all other services in the second channel. It then gradually moves services from the second channel to the first one. As the services are sorted beforehand based on the popularity, the single channel average times would first increase and then drop after it reaches the peak. Once such dropping is detected, the process stopped.

After going through all channels, the algorithm examines the best partition for each channel, which is considered as a candidate partition for the current step. It computes MAAT for each such candidate and chooses the one that yields the best MAAT (Line 14). After performing this partition, the current number of channels will be increased by 1 as services are allocated among one more channels than before.

This process iterates the number of current channels achieves the number of available channels.

4.3 Dynamic Programming

In this Section, we describe a dynamic programming approach, following the idea proposed in [6]. The optimization is based on the formula below. Following that is an example of the steps that Dynamic Programming goes through in its search for the best outcome (in this case lowest AAT possible for all available MWSs).

$$dpOptSol_{i,K} = LAAT(SAAT_{im} + dpOptSol_{m+1,K-1}) \tag{6}$$

K = number of the hybrid channels
$LAAT$ = lowest MAAT
m = total number of MWSs -1
i = start index of MWSs in a channel $(1 \leq i \leq m)$
j = end index of MWSs in a channel $(i < j \leq m)$

Here, we use an example to illustrate the process of dynamic programming. Suppose there are six mobile services and three broadcast channels. We first sort six MWSs based on their request frequencies and lengths, similar to the sorting process in greedy algorithm. Suppose that the final ordering result is: MWS3, MWS5, MWS1, MWS4, MWS6, and MWS2, where MWS3 ranks the highest and MWS2 ranks the lowest. Based on the Equation 6, the allocation or configurations of DP for three channels are listed as follows:

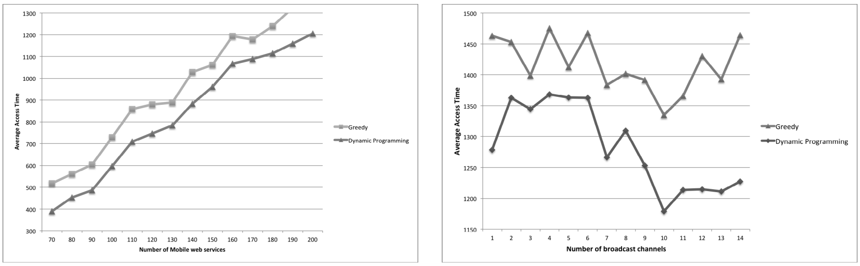
MWS3	MWS5	MWS1MWS4MWS6MWS2
MWS3	MWS5MWS1	MWS4MWS6MWS2
MWS3	MWS5MWS1MWS4	MWS6MWS2
MWS3	MWS5MWS1MWS4MWS6	MWS2
MWS3MWS5	MWS1	MWS4MWS6MWS2
MWS3MWS5	MWS1MWS4	MWS6MWS2
MWS3MWS5	MWS1MWS4MWS6	MWS2
MWS3MWS5MWS1	MWS4	MWS6MWS2
MWS3MWS5MWS1	MWS4MWS6	MWS2
MWS3MWS5MWS1MWS4	MWS6	MWS2

For each allocation, we compute the SAAT for each channel, then compute the overall performance as the sum of the SAATs of all channels. Dynamic Programming is more efficient than brute force for two reasons. First, it only checked those allocations that have the potential of achieving the best performance. In this example, only 10 allocations are examined, while 30 allocations are examined in a brute force approach. Second, as the result of each computation will be stored, repeated calculation will be avoided. For example, the SAAT for the channel (MWS2) will be only computed once instead of four times.

5 Experimental Study

We performed experiments to compare the performance of the greedy and dynamic program mining approaches. To our best knowledge, there is no publicly available event log data sets that suitable for our experimental study. We thus conducted our experimental study on a synthetic data set. We developed a Java-based simulator which generates random size mobile web service description and service data. All experiments were carried out on a Macbook Pro with 2.3 GHz Quad-Core Intel Core i7 processor and 16GB DDR3 memory under Mac OS X operating system.

In the first set of experiments, we examined the effectiveness of the greedy algorithm and dynamic programming algorithm when the number of broadcast channels is fixed (i.e., 10), and the number of mobile web services varies from 70 to 200. As depicted in Figure 3 the DP algorithm does come up with lower MAAT.



A fixed number of broadcast channels (10) with a variable number of available mobile web services

A fixed number of mobile web services (200) with a variable number of available broadcast channels

Fig. 3. Comparison of MAAT of Greedy and DP algorithms

In the second set of experiments, we examined the effectiveness of the two algorithms when the number of the mobile web services is fixed (i.e., 200) and the number of broadcast channels varies from 1 to 14. As depicted in Figure 3, the DP algorithm achieves slightly better MAAT than greedy algorithm.

In the third set of experiments, we compared the efficiency of the two algorithms. We use the number of calculations performed during the entire optimization process to indicate the cost. The number of broadcast channels is 30 and the number of mobile web services varies from 74 to 100. As depicted in Figure 4, the greedy algorithm is significantly more efficient than DP algorithm. Moreover, it also exhibits a better scalability with number of mobile web services.

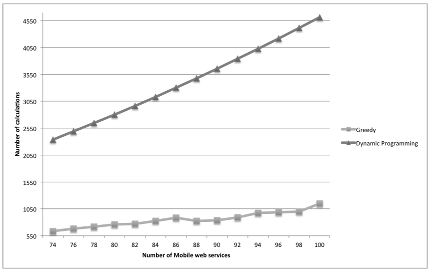


Fig. 4. Comparison of the number of calculations performed in Greedy and DP algorithms

6 Related Work

With a firm understanding of the mechanics of web services, further (more specific) research on mobile web services provided us with novel approaches and valuable "food for thought." One such example is the concept of accessing web services via cell phone broadcast networks [1, 2]. Another paper describes different broadcast channel structures that were invented to share any data through a cell phone broadcast network [2]. Yet another research introduces the concept of *skewing the data* to accommodate the fluctuations in the number of requests for different types of data [1, 2, 6].

Not surprisingly, a substantial number of researchers have included the use of broadcast channels in their mobile web services solutions due to the advantages of scalability, energy efficiency, and unlimited user capacity that they bring to the table [1, 2, 6, 7] The challenge becomes the allocation of the data among the broadcast channels so that the *average access time* (AAT) is consistent (regardless of the "popularity" of the data requested). One approach is to allocate bandwidth to the transmission of less frequently requested data while leaving broadcast to handle the more popular data, as was suggested in [8] and [9]. A different concept is to send out the most popular information requests at increased intervals on the broadcast channel, thereby skewing the data allocation [6].

In essence, Yee *et al.* 's [6] research is based on the theory that by using multiple broadcast channels (instead of the usual single channel design), response time between mobile devices and web service providers will be improved and the added benefits of improved "fault tolerance, configurability and scalability" could be realized. The team's unique approach called for partitioning the data (and scheduling it) among several broadcast channels with their simplified GREEDY approximation algorithm. In their experiments, they compared the performance of their algorithm (GREEDY) to FLAT, Bin-Packing (BP), VF^K, and Dynamic Programming (DP) on data items of the same size. Their goal was optimization balanced with practicality, thus taking into consideration the limitations of the on-line and mobile environments. As they had hoped, the GREEDY algorithm produced near-optimal performance within its less complex (when compared to previously used algorithms) operational mode [6].

Tackling more complicated wireless transactions, Yang *et al.* [2] sought to apply efficiencies to accessing composite mobile services (a service that depends on other services to work), which are defined by Business Process Execution Language (BPEL) graphs. They discovered that the *closest-first-access* algorithm had advantages over the *depth-first-access* and *breadth-first-access* algorithms no matter the type of composite service (*i.e.,* sequential, parallel, or hybrid). Due to the intricacies of these operations, the team sought to break down the access method into three processes: service request flow, receiver flow, and service execution flow. Their proposed broadcast channel configuration was made up of seven channels with each having its own designated purpose. More specifically, they were each assigned a certain type of content, such as M-services, wireless data, service descriptions, composite services, or service registry information.

Additional efficiencies for the above configuration were sought by the team through "selective tuning," which would allow the mobile devices to rest (doze) and only activate when its required content became available. This was accomplished by developing their "index channel method," which consisted of a *M-service index channel* and a *data index channel*. Both channels provided service or data location information ("arrival time") in index format to the mobile device, which allowed it to doze while waiting and thus reduced its tuning time (and energy consumption).

In the research above (and historically speaking), those designing and delivering mobile web services have had to contend with the limitations (*i.e.,* battery life, bandwidth, and storage) that are inherent in most mobile devices. Contrary to this thinking, Chou and Li [10] designed a mobile distributed computing platform (based on Android technology) that facilitates distributed services computing on mobile devices utilizing real-time and over IP communication capabilities. Their Web Service Initiation Protocol (WIP) enables mobile devices to act like/appear as web service providers. Along the same lines, Aijaz *et al.* [11] set out to prove that mobile devices could host asynchronous mobile web services (Mob-WS). Their scenario had web service requestors and web service providers both being mobile devices. They established peer-to-peer communication between the parties by leveraging Bluetooth® technology.

All of the research mentioned above and what has been published up until the date of this paper have not considered utilizing skewed hybrid broadcast channels (disseminating variably-sized mobile web services) to reduce *average access time.* This paper is devoted to the design and execution of this conceptual model which includes its innovative channel design and allocation approach.

7 Conclusions

We proposed an efficient framework for delivering mobile web services in a broadcasting environment. We proposed the measurements to evaluate the performance of the framework, including single channel average time and multiple channel average time. We formally defined the optimization problem. We described and compared three optimization solutions. We conducted an experimental study, which showed that the greedy algorithm achieves the best performance overall. We plan to extend our work in the future to incorporate tuning time as well as the performance of updating the indexing channels into the optimization process.

References

1. Yang, X., Bouguettaya, A.: Semantic access to multichannel m-services. IEEE Transactions on Knowledge and Data Engineering 21(2), 259–272 (2009)
2. Yang, X., Bouguettaya, A., Liu, X.: Efficient access to composite m-services. In: Proceedings of the 2009 IEEE International Conference on Web Services, ICWS 2009, pp. 381–388. IEEE Computer Society, Washington, DC (2009)
3. Yee, W.G., Navathe, S., Omiecinski, E., Jermaine, C.: Efficient data allocation over multiple channels at broadcast servers. IEEE Transactions on Computers 51(10), 1231–1236 (2002)
4. Yee, W.G., Navathe, S.B., Omiecinski, E., Jermaine, C.: Bridging the gap between response time and energy-efficiency in broadcast schedule design. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 572–589. Springer, Heidelberg (2002)
5. Yang, X., Bouguettaya, A., Liu, X.: Semantic-based access to composite mobile services. Int. J. Web Service Res. 8(3), 70–100 (2011)
6. Yee, W.G., Navathe, S., Omiecinski, E., Jermaine, C.: Efficient data allocation over multiple channels at broadcast servers. IEEE Transactions on Computers 51(10), 1231–1236 (2002)
7. 3GPP2: Cdma2000 high rate broadcast-multicast packet data air interface specification (2004), http://www.3gpp2.org/public_html/specs/C.S0054-0_v1.0_021704.pdf (accessed December 18, 2011)
8. Acharya, S., Alonso, R., Franklin, M., Zdonik, S.: Broadcast disks: data management for asymmetric communication environments. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, SIGMOD 1995, pp. 199–210. ACM, New York (1995)
9. Yee, W.G., Navathe, S.B., Omiecinski, E., Jermaine, C.: Bridging the gap between response time and energy-efficiency in broadcast schedule design. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 572–589. Springer, Heidelberg (2002)
10. Chou, W., Li, L.: Wipdroid - a two-way web services and real-time communication enabled mobile computing platform for distributed services computing. In: Proceedings of the 2008 IEEE International Conference on Services Computing, vol. 2, pp. 205–212. IEEE Computer Society, Washington, DC (2008)
11. Aijaz, F., Adeli, S.M., Walke, B.: Middleware for communication and deployment of time independent mobile web services. In: Proceedings of the 2008 IEEE International Conference on Web Services, pp. 797–800. IEEE Computer Society, Washington, DC (2008)

A Two-Leveled Semantic Web Service Description of the Pervasive Information System

Achour Fatma, Jedidi Anis, and Gargouri Faiez

Multimedia, Information Systems and Advanced Computing Laboratory
Sfax University

Abstract. To begin with, the semantic web service adaptation is based on the addition of contextual information to this web service description. So the technique of adapting the web service allows the user to establish reusable, interoperable, flexible and context-aware applications. The basic feature of the pervasive applications is the context-awareness. What is more, the pervasive systems are based on the NNN paradigm (aNywhere, aNytime, with aNything). Therefore, these systems must be used in different contexts depending on the user environment, his profile and the used terminal. Indeed, this type of system is dynamically adapted. Furthermore, we seek to offer a dynamic description of the web services. This objective, however, can be achieved by adding contextual information structures to the web service description. In the literature, more research works strive to combine the contextual description with the web service description by using the OWL-S structure. In fact, the main objective of this paper is to propose structures with two levels for the contextual description of the pervasive system. The first level is used to select the adapted web service to respond to the users needs, while the second level is aimed to offer the user the adapted description to respond to these needs.

Keywords: Pervasive, Information system, Ontology, OWL, OWL-S, Design, Model.

1 Introduction

In the last decade, new technologies witnessed a rapid evolution. This evolution stepped up the complexity of the users mobile environment and pervasive computing becomes a reality. As a result, a large number of pervasive computing projects around the world try to prove the viability and usefulness of this vision in a variety of domains. Unfortunately, along with the new and existing possibilities, difficult challenges come into being. As pervasive applications are highly distributed and mobile in nature, pervasive system developers have to transcend all the challenges found in the fields of distributed systems, mobile computing, as well as the new set of challenges that did not exist in either the distributed systems or mobile computing alone.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 35–48, 2013.

In this, adaptability is deemed to be a key aspect in the pervasive computing systems. Therefore, the system needs to adapt its functionality and behavior depending on the context of the user and the resources that are available at any instant. Although many research works highlight the idea of implementing and establishing adaptation in pervasive systems, none of them is interested in the design of adaptation in pervasive systems. We attempt, in this work, to suggest a system that can be conceptually adapted to the pervasive system. This system is based on semantic web services. Therefore, we have proposed a classification of contextual information for pervasive systems and used the OWL-S structure to integrate such information into the web services. In fact, the OWL ontology is a solution to support greater automation in the service selection and invocation and the automated translation of the message content between the heterogeneous inter-operating services and the service composition.

This paper is organized as follows. First, we study some concepts related to our research domain such as the pervasive system, the semantic web services and the existing OWL-S extension to describe the semantic web services. Then, we present the architecture to adapt the pervasive system functionalities. Finally, we present our purpose that concerns the proposal of contextual description of the pervasive system.

2 State of the Art

2.1 The Semantic Web Service

To realize the vision of Semantic Web services, several research works create a semantic markup of Web services that makes them understood by the machine. They are also developing the agent technology that exploits this semantic markup to support the automated Web service composition and interoperability. The convergence of the semantic web with the service oriented computing is manifested by the semantic web services technology. In fact, it addresses the major challenge of automated, interoperable and meaningful coordination of web services to be carried out by intelligent software agents. Additionally, each semantic service description framework can be characterized with respect to (1) what kind of service semantics are described, (2) in what language or formalism, (3) allowing for what kind of reasoning upon the abstract service descriptions. The Semantic Web services can use the OWL-S process model and grounding in order to manage their interactions with other web services.

2.2 The OWL-S Ontology Presentation

OWL-S is a Web Services ontology that specifies a conceptual framework to describe the semantic web services. OWL-S is also a language based on the DARPA work of its DAML program and takes the result of DAML-S (DARPA Agent Markup Language Service). It was incorporated into W3C in 2004, within the interest group of semantic web services at the OWL recommendation [3].

The original purpose of OWL-S is to implement semantic web services. OWL-S is based on OWL to define the abstract categories of entities, events in terms of classes and properties. OWL-S uses this ontology language description to define a particular ontology for the web services. This ontology is used to describe the web service properties as well as its services available to the public. The OWL-S structure regroups a set of ontology. Each one provides a functionality to describe the web service semantically. The ontology main classes described by OWL-S [4] are defined by the following figure (see Figure 1). The necessity to use the OWL-S ontology is justified by the creation of a semantic web service that has a dynamic description. This dynamic is provided by the addition of contextual descriptions to the OWL-S structure. The description depends on the use of the context of a pervasive system.

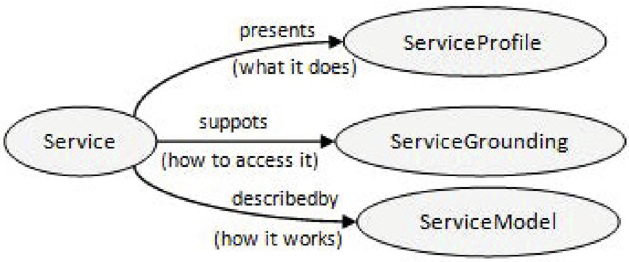


Fig. 1. The principal OWL-S classes

2.3 The Existed OWL-S Extension

Several research works take the advantage of the existing OWL-S structure to describe the different contexts. In this paper, we present two research works of of Qiu et al., [5], [6] and Ben Mokhtar [7]. Qiu et al., research works propose an adaptation system based on the service composition approach. To do this, the authors offer three context categories [5]: the user context, the web service context and the environmental context. The user context ("U-Context") specifies the context information about the user. In this context, the authors defined two types of contextual information: the user static context (profile, interest, and preferences) and the user dynamic context (location, current activity and task trying to achieve). The web service context ("W-Context") includes the not-functional contextual information (price, execution time, confidence degree). The environmental context ("E-Context"): this category collects the context information about the user's environment (time, date, etc.). Each context category is represented by the OWL ontology and is integrated in the existing OWL-S extension ontology to introduce the OWL-SC (OWL-S for context) [6]. The latter is intended to describe a general contextual information (see Figure 2) based on the users description. The proposed structure focuses only on the user context

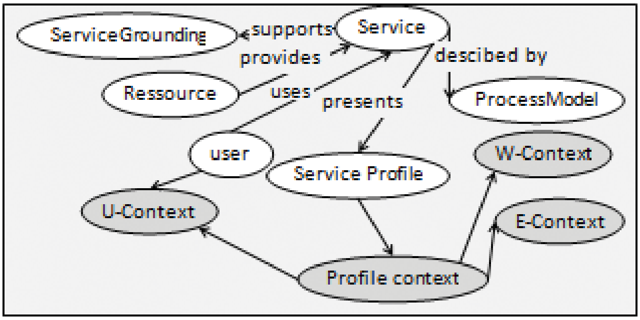


Fig. 2. The OWL-SC Ontology [6]

description. However, it presents a vision for the integration or the addition of more information to the OWL-S structure.

Ben Mokhtar et al., research works propose a system to adapt the web services to a pervasive environment [7]. The context definition includes the description of four types of contextual information: the context sensors, services, devices and users. In addition, the contextual adaptation in this work is based on the service representation and the user task representation. In the service representation, the authors describe the services using OWL-S extended with context information. This information is decomposed into a high level context attributes, preconditions and contextual effects. However via the user task representation, the user task representation is performed while extending the OWL-S service model ontology. To do this, the authors propose to integrate the quality conditions service descriptions and the context conditions required by the user task in the OWL-S structure. The context information description is performed by means of an OWL extension ontology, the adaptation is carried out by applying a finite-state automaton (see Figure 3). Ben Mokhtar et al., their proposal concerns the contextual information integrated in the OWL-S structure, but this information is

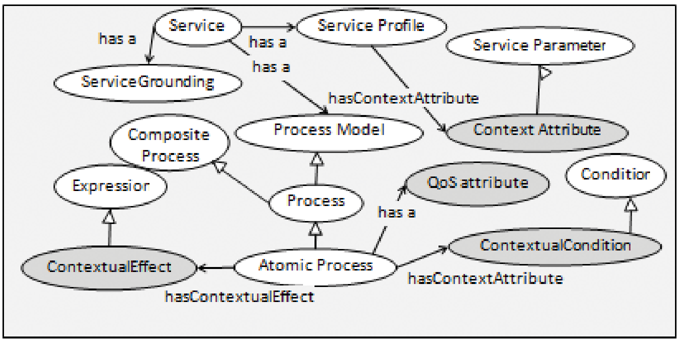


Fig. 3. The OWL-S ontology extension for the pervasive system [7]

not related to the pervasive system description. We focus on, via this work, the integration of pervasive contextual information in the OWL-S structure.

2.4 The Pervasive Systems

Weiser [1] believed that the exponential evolution of data, software, hardware and connectivity would generate new environments rich of computing elements that lack proper interaction. As a result, he presented a paradigm where the computing elements would disappear from the users consciousness while functioning homogeneously in the background of his environment. The final objective is to provide the user with omnipresent and seamless services available whenever and wherever they are needed [2].

When the digital services kept increasing, many applications appeared. These applications can sometimes replace human beings. Thus, technology becomes a principal factor that could affect and improve life quality and business productivity by ensuring higher interoperability between the different business partners and the surrounding dynamic environments. With all these dynamic elements, a growing demand for easier, mobile, transparent and seamless interaction is sought in order to adapt it to the users situations, abilities and needs.

2.5 The General Model Used to Describe the Pervasive System

The pervasive system description is based on a general model [9]. The latter is a UML class diagram representing the intersection result of four existing models to describe the pervasive systems: SOUPA, Activity, and CSCP COMANTO. The classes defined in this diagram (see Figure 3) are described below:

- **Agent:** this class is used for the presentation of different actors in a pervasive system. It assembles the human actor and device actor.
- **Person:** presents all human actors.
- **Device:** presents the peripheral devices in a pervasive system.
- **Service:** presents the services offered by each device.
- **Network:** regroups the characteristics of different types of network.
- **Location coordinate:** represents the spatial relation between the different locations in a pervasive system.
- **Preference:** presents the information profile of a person who realizes the activity in a pervasive system.
- **Activity:** presents the characteristics of the activity requested by the user.
- **Rules:** regroups the different rules of activity, person and network interacting in a pervasive system.
- **Time:** presents the characteristics of a temporal thing and the relation of different things in a pervasive system.
- **Location:** represents the characteristics of localizations of human and mobile devices in a pervasive system.
- **Role:** represents the role of the users in the pervasive system.

The class descriptions are classified into six contextual information categories: the user context, the device context, the network context, the location context, the service context, and the application context. We consider a semantic web service to adapt the pervasive application to each category of contextual information.

3 The Purposed Architecture to Adapt the Pervasive System

The proposed architecture to adapt the functionality of the pervasive system to the users needs regrouped two description levels and an adapter. The adapter is used to apply an adaptation rules to the destined web service description. In this paper, we are interested in the description levels (see Figure4). The goal of the generic web service creation is to communicate with the different contexts by selecting the specific web service to respond to the users needs. We used the web service discovery technique to ensure this level functionality. The web service discovery is the process of finding a suitable web service for a given task. The specific web services are created to select the adapted description to respond to the users need. The fulfillment of a specified web service is provided by the generic web service. The next section is devoted to the presentation of the contextual description used for each level.

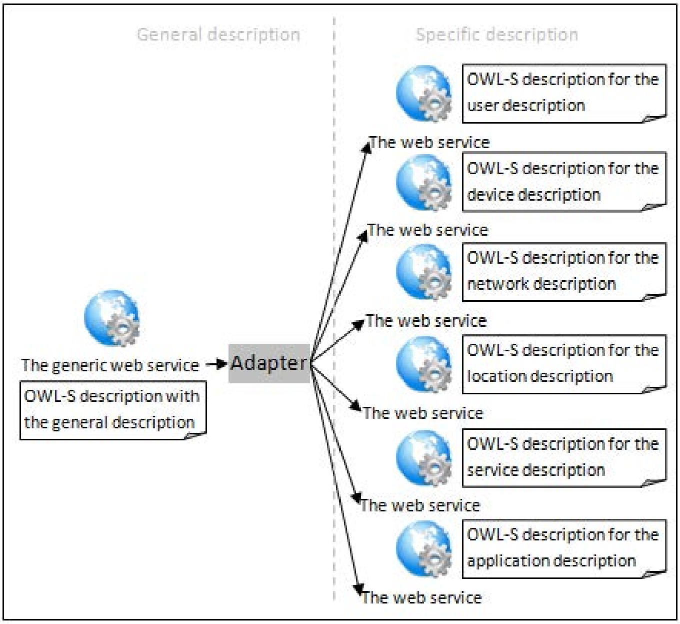


Fig. 4. The purposed adaptation architecture

4 The First Level of the Web Service Description

The first level is represented by a semantic web service that generally describes the pervasive environment. It has all the necessary information about each web service context shown in the second level. This information is described in an extended OWL-S ontology (see Figure 5). The extended OWL-S ontology includes the basic information examining a pervasive system.

The pervasive context is presented by the "PervasiveContext" OWL class. The activities in a pervasive system are presented by the "A-Context" OWL class. They exist in a device. The latter is symbolized by the "D-Context" OWL class. Each "D-Context device offers services in a pervasive system. The services are presented by the "S-Context" OWL class. The latter regroups the characteristics of the services provided by the pervasive system. All the devices existing in a pervasive system are interconnected via multiple networks. These networks are modeled through the "N-Context" OWL class. The two classes "D-Context" and "U-Context" represent the entire agent that exists in a pervasive system. For this reason, we placed the two classes as a sub-class of the "Agent" OWL class.

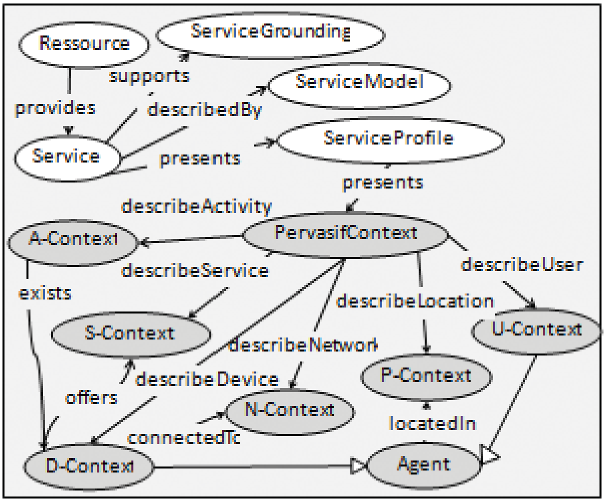


Fig. 5. Proposed structure for the first level of description

Each of the classes presented in the proposed OWL-S structure will be transformed in ontology. The latter regroups the classes and the attributes shown by the OWL semantic relation "owl:onProperty". The ontological structures are used to detail the contexts defined in the pervasive system.

5 The Second Level of the Web Service Description

The second level is the most detailed contextual description. It is at this level that we discover the low-level description of the pervasive system (see Figure **??**). To define each context, we have to get a set of extended OWL-S structures corresponding to the number of contexts described in a pervasive system.

5.1 The OWL-S Extension to Describe the Physical Context (OWL-SPHC)

The suggested OWL-S extension of the physical Context is an ontology that describes the user physical context as well as the set of devices and persons existent in a pervasive system. The physical context includes the contextual information related to the location and the location coordinates of an agent in a pervasive system. The idea here is to add the physical context description to the existing OWL-S ontological structure. The OWL-S extension structure is suggested in figure 6. The adaptation of the user to the physical context is mainly manifested by the constraints expressed by the rules related to the execution of the applications in a pervasive system. These constraints can be summarized by the example in which an application is running in a location such as train station (where there is huge noise) or in a location such as a house (where there is silence). The time constraints occur when an application is executed in the same location but at different times. For instance, the execution constraints of an application during a work break are not the same as when it is running.

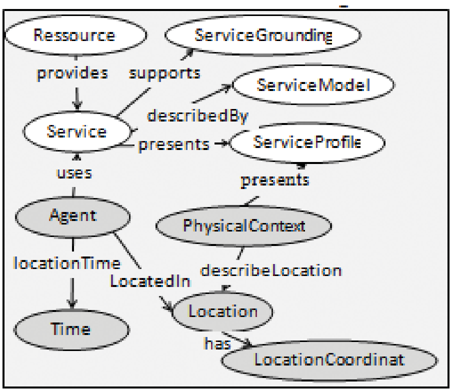


Fig. 6. The structure OWL-SPHC

5.2 The OWL-S Extension to Describe the Application Context (OWL-SAC)

The OWL-S structure is designed to describe the application context is an ontology structure that defines the activity context executed in the pervasive system.

The activity context includes both the contextual information related to the activity and the application that uses the rules in a pervasive system. The idea here is to add the activity context description to a standard OWL-S ontology. We present the suggested structure of the OWL-S extension in the following figure (see Figure 7).

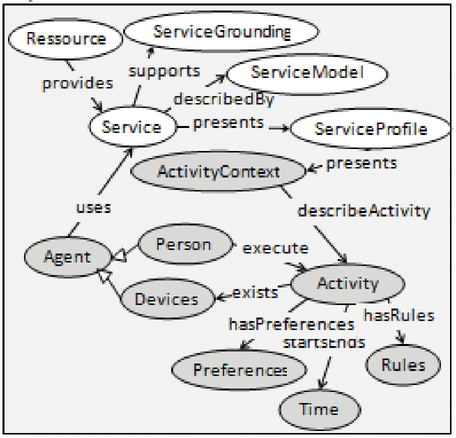


Fig. 7. The structure OWL-SAC

The adaptation of an activity in a pervasive system manifests itself in the activity characteristics and the rules implementing this activity. The devices characteristics may affect the activity performance. The activity execution begins and ends at a given moment in time. A computer activity can be executed in a specific platform may not be executed in another environment. In other words, audio or multimedia activity can be executed only in multimedia environments.

5.3 The OWL-S Extension to Describe the User Context (OWL-SUC)

In fact, the user is the focal point of any information system. In the pervasive system, the user is defined by two information characteristics: the static characteristics (name, age, sex, etc.) and the dynamic characteristics. The dynamic characteristics are defined by the preferences and the users environment (location, time, etc.). To describe the user context, we consider the OWL-S extension structure presented in figure 8 which combines the two types of characteristics of the user context.

The adaptation of the applications to the users characteristics manifests itself essentially during the interaction between the human-machine interfaces and the method of executing an activity. An activity can be executed according to the users preferences. In addition, the users static characteristics can impact on the methods of using a pervasive system since this type of system is adaptable to any user.

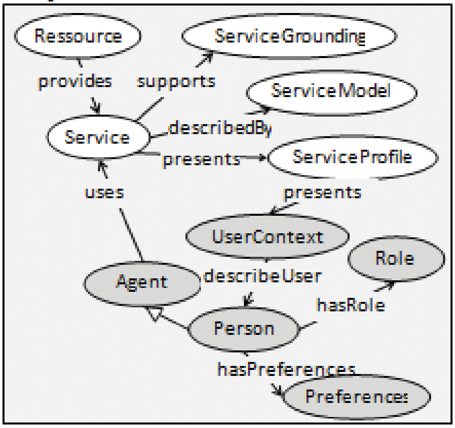


Fig. 8. The structure OWL-SUC

5.4 The OWL-S Extension to Describe the Service Context (OWL-SSC)

The OWL-S structure used to describe the service context as an ontology which contains information about the services provided by the pervasive system. Each service in the pervasive system has a set of rules. Since we have an intelligent system, this system must provide services according to the users desires and preferences such as the physical features, the available networks, etc. The OWL-S ontology is proposed as follows (see Figure 9).

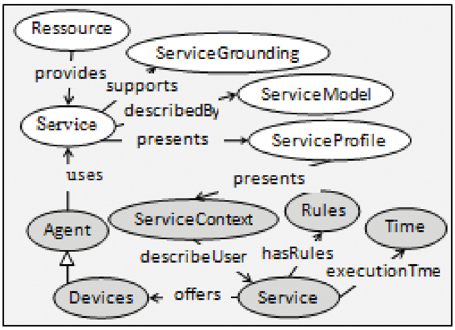


Fig. 9. The structure OWL-SSC

This ontology is meant to describe the services provided by the pervasive system and to adapt an application to the user's desires and preferences. In other words, a single system must interact with any type of users, networks, devices, etc.

5.5 The OWL-S Extension to Describe the Device Context (OWL-SDC)

The OWL-S structure is created to describe the device context and the activity profiles in the pervasive information system. Each device has a configuration, rules and preferences. In the mobile system, a new communication method has emerged to satisfy the users needs. Such a method paves the way for the propagation of intelligent systems through the invention of smart phones such as the Blackberrys, the iPhone... and touch pads such as the iPad. In order to ensure the adaptation, the pervasive system must capture a material list to ensure the answer to the query in accordance with the hardware configuration (see Figure 10).

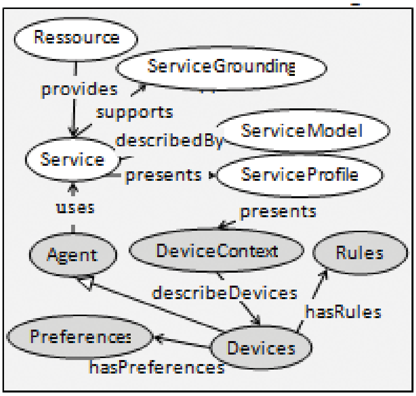


Fig. 10. The structure OWL-SDC

The material side is very important since the pervasive system is accessible anywhere, anyhow and to anything. Indeed, such a system can be executed according to the existing hardware.

5.6 The OWL-S Extension to Describe the Network Context (OWL-SNC)

The proposed OWL-S structure regroups the network context description and combines the characteristics of all the available networks in the pervasive information system. In this type of information system, there are several types of wireless networks (3G, WiFi, GSM, UMTS ...). Actually, each type of network has its own characteristics, access rules and preferences. To describe the network context, we propose the following OWL-S structure extension (see Figure 12).

The network selection or adaptation in the pervasive system is provided by the proposed ontology (see Figure 11) since the web service will use this structure to describe its services semantically. We can give real examples to describe

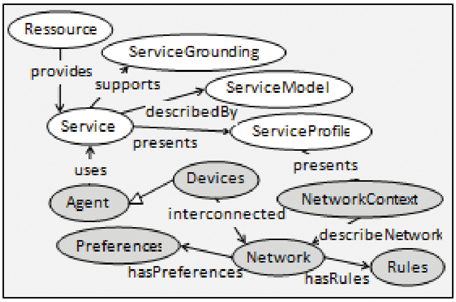


Fig. 11. The structure OWL-SNC

the network in a pervasive system: the traveler can check and send his documents according to the characteristics of the wireless networks available in the airport, the patient can make an appointment through his PDA, the manager can supervise the workers remotely, etc.

6 An Example of Using the Proposed OWL-S Extensions

In this paper, we present an OWL-S extension for each context. In this section, we present a scenario to give an idea about the description related to the pervasive system. The presented scenario (see Figure 13) is obtained after applying the representation of the rules exemplified in the figure below. In other words, the rules are defined by a set of conditions. The latter resulted from the users request values by means of logic operators ("AND" or "OR"). The value used in this presentation is manipulated by the two operations "getValue()" and "setService()". The result may be one service or several services according to the logic operator (see Figure 12).

6.1 The Scenario Definition

A user in an airport waiting for a flight. He has a large multimedia file. He wants to send it to the wireless network in a transmission time that does not exceed 30 minutes.

6.2 The Scenario Result

In this scenario, three types of context are defined; the physical context, the activity context and the network context. Among the classes of the physical context Physical Context, we have the class location that has the attribute Location Name. In the existing structure, we note the existence of three possible values: airport, work and home. In this situation, the sensor detects that the user is at the airport. This value is returned by the method getUserLocation(). Across the context of activityActivityContext, we can extract the type and the execution time of the activity. The type activity is returned by the attribute ActivityType of the class Activity in this situation which includes three possible values: play online, chat on facebook and send multimedia documents. The potential value in this case is expressed by the method getActivityType(). The activity execution time is returned by the attribute ExecutionTime of the class Time which includes, in this situation, the values: 15 minutes, 30 minutes and 45 minutes. The potential value in this case is expressed by the method getExecutionTime(). The final context is the network context NetworkContext. It allows returning the network to be used in this scenario. The class Network has an attribute Bandwidth which includes the bandwidths value of all networks stored in this structure. This value is perceptible by means of virtual sensors.

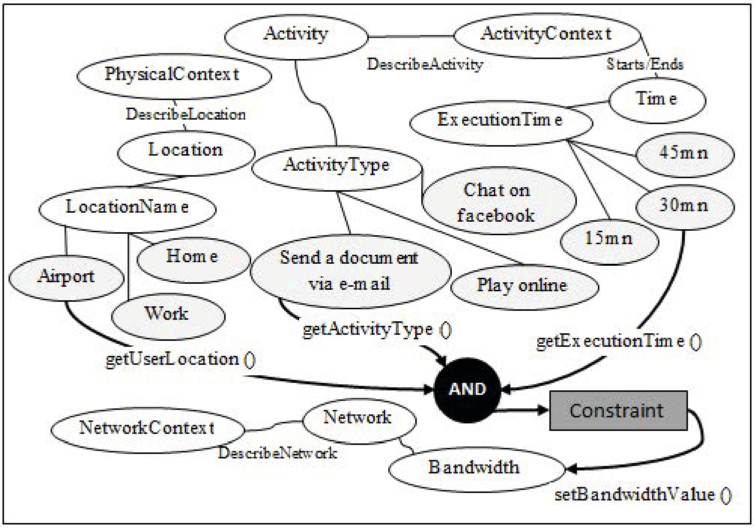


Fig. 12. The scenario description

7 Conclusion

This paper, indeed, is aimed present the two leveled description of the pervasive system based on the contextual information derived from a generic model. Hence, we suggest a classification of this information. The result of this classification is six ontology; the user context ontology, the physical context ontology, the network context ontology, the activity context ontology, the device context ontology, and the service context ontology. These ontology are implemented in the semantic web services to give the pervasive system developers an adapted description. This set of ontology is integrated into an existing OWL-S structure to create the semantic web services and to determine the second level of description. Also, we propose a generic description for the first level. We kept the model of generic structure to ensure communication between the different contextual information In future work, we intend to suggest a set of generic rules that can be applied to this ontology to adapt the functionality of the pervasive system.

References

1. Mark, W.: The Computer for the Twenty-First Century. Scientific American, 265–268 (1991)
2. Park, I.-S., Kim, W.-T., Park, Y.-J.: A Ubiquitous Streaming Framework for Multimedia Broadcasting Services with QoS Based Mobility Support. In: Kahng, H.-K., Goto, S. (eds.) ICOIN 2004. LNCS, vol. 3090, pp. 65–74. Springer, Heidelberg (2004)
3. David, L.M., Mark, H.B., Drew, V.M., Sheila, A.M., Massimo, P., Katia, P.S., Deborah, L.M., Evren, S., Naveen, S.: Bringing Semantics to Web Services with OWL-S. World Wide Web, 243–277 (2007)
4. David, M., Mark, B., Jerry, H., et al.: OWL-S: Semantic Markup for Web Services W3C. Member Sumbission (2004)
5. Lirong, Q., Zhongzhi, S., Fen, L.: Context Optimization of AI planning for Services Composition. In: ICEBE, pp. 610–617 (2006)
6. Lirong, Q., Liang, C., Fen, L., Zhongzhi, S.: Context optimization of AI planning for semantic Web services composition. Service Oriented Computing and Applications, 117–128 (2007)
7. Ben Mokhtar, S., Fournier, D., Georgantas, N., Issarny, V.: Context-aware service composition in pervasive computing environments. In: Guelfi, N., Savidis, A. (eds.) RISE 2005. LNCS, vol. 3943, pp. 129–144. Springer, Heidelberg (2006)
8. Anind, K.D.: Understanding and Using Context. Personal and Ubiquitous Computing, 4–7 (2001)
9. Achour, F., Jedidi, A., Gargouri, F.: The generic model for pervasive information system. In: Proceedings of the 3rd International Conference on Information and Communication Systems, ICICS, pp. 18:1–18:7 (2012)

Enabling Location-Based Services on Stationary Devices Using Smartphone Capabilities

Thomas Graf, Sebastian Zickau, and Axel Küpper

Service-centric Networking
Technische Universität Berlin, Germany
thomas.bs.graf@campus.tu-berlin.de,
{sebastian.zickau,axel.kuepper}@tu-berlin.de

Abstract. In recent years, location-based services became more and more popular. They serve information based on a user's or device's location. Many of these services are accessible over web interfaces via browsers, which would make them also interesting for stationary devices. However, these don't possess good positioning capabilities necessarily. There are still many of them that are connected to the Internet in a wired way, thus not being able to use WLAN positioning or similar approaches. Static devices usually rely on IP geolocation, which approximates their position only coarsely. Smartphones, on the other hand, come with inbuilt positioning technologies that allow to pinpoint the device and its user with very high accuracy. A potential solution to this dilemma is a connection between different device types. Highly accurate position information can subsequently be transferred in an automated and seamlessly integrated manner from a smartphone to a stationary device.

Keywords: Location-based services, positioning mechanisms, geolocation, mobile and stationary devices.

1 Introduction

Location-based services (LBS) and related applications are becoming more and more popular. Nowadays, the integration of location data into a multitude of objects of interest, such as photos or blog posts is very popular. However, LBSs are much more than just the mentioned geotagging example. They can be defined in many different ways. Brimicombe and Li give the following definition: "Location-based Services (LBS) are the delivery of data and information services where the context of those services is tailored to the current or some projected location and context of a mobile user." [1]

One of the most prominent examples for LBSs is finding points of interest, such as restaurants, in close vicinity to the current position of a user or device. Other location-influenced information could be bus departure times and explanations of historic monument backgrounds. Most of the time, such information is directly queried by the user, e.g., by accessing a location-aware application

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 49–63, 2013.

(app) on a smartphone, which means that the user controls and initiates the position transfer to the location-based service. This form is called reactive LBS, because the service only responds with location-aware information on demand. In comparison to this, proactive LBSs constantly have knowledge of users' positions and provide them with information based on certain events, e.g., passing by a cinema. In this case, such an event could be used by the cinema operator to offer cheaper tickets to a nearby passerby. A more advanced approach is called geo-fencing [2], where entering and leaving predefined areas is monitored [3]. Aside from using only single positions, services can also work with a user's location history.

Considering that the usage of these services includes sharing location information with third parties and often also making them publicly available, privacy concerns automatically arise. Even though many LBSs are primarily designed for mobile devices, they can also be used by stationary devices, such as desktop computers and similar devices. Location-based access control (LBAC) [4] is an example, as it would allow restricting access to important information based on the user's whereabouts [5]. These devices usually rely on IP geolocation, which approximates their position only with a city-level granularity at most. Smartphones, on the other hand, come with inbuilt positioning technologies that allow pinpointing the device and its user with very high accuracy. A potential solution to this dilemma is a connection of smartphones and stationary devices. This paper discusses different concepts and presents a prototype implementation.

After the introduction, an overview of related work is given. In Chapter 3, the most important technologies regarding positioning and device connections are presented. Chapter 4 introduces system requirements and discusses the realization concepts, as well as the implementation experiences. The paper rounds up with chapters on evaluation, summary and outlook.

2 Related Work

Connecting smartphones and stationary devices in order to exchange generic data is not something entirely new. A range of applications exist that incorporate some sort of connection mechanism to other devices, most often wireless local area network (WLAN) or Internet-based connections. However, none of them can directly be used to provide positioning information to stationary devices.

In 2009, Bump technologies published a smartphone application (Bump) that allows file exchange with other smartphones or computers. The connection is established by bumping the smartphone on the computer's space bar while accessing the company's website. A location-based matching algorithm uses time and position of both devices to detect and connect them. The actual connection is only permitted after an additional display-based authentication check. Data is not directly transferred between the devices, but routed over the website. According to the website[1], the entire connection is encrypted so that no private

[1] `https://bu.mp/`

data is accessible by a third party. Airdroid[2] is a second, similar application that additionally lets a computer take control over a smartphone. In contrast to Bump, it works only in a shared WLAN via a web-based interface. No third party server is required. Aside from the connection method, it includes a graphical authentication mechanism. The stationary device accesses the web interface, which is hosted by the smartphone. A quick response code (QR-Code) [6] is generated and displayed. The user has to scan this code with the smartphone's camera, proving that the computer, from which the smartphone is accessed, is the correct one. The latest version of Airdroid features the access to smartphone data over cellular networks.

3 Geolocation Basics and Connection Concepts

In the following section an overview of used positioning techniques and approaches is given for stationary devices.

3.1 IP Geolocation

The most important, automatic and basic way to discover the location of a static device that lacks technologies like GPS and WLAN is to use its Internet Protocol (IP) address. Each device, or more precisely each network interface, is assigned with an IP address when it connects to a network, e.g., the Internet. The main problem with geolocating a device through its IP address is that the address itself does not contain any location specific information at all. Therefore, other ways that involve additional context information and measurements have been developed to get a location to a corresponding IP address in real time. They can be classified by the approach they take to acquire an IP address location into basically three groups: network delay-based, topology-based and hybrids. The most important are: Constraint-based Geolocation (CBG) [7], Octant [8] and Street-level Geolocation (SLG) [9].

3.2 Comparison of Common Positioning Technologies

Almost every modern smartphone provides software functionality and sensors for the following three positioning systems. Some of them are tied deeply into the operating system of the mobile device. In Table 1, an overview of the most common and well-known positioning systems is given with details on accuracy, power consumption, and availability. It becomes apparent that positioning technologies available to smartphones are much better than IP geolocation. There are several drawback, e.g., GPS is mostly not suitable for indoor positioning and cellular positioning can be influenced by deflections from buildings and other large objects. The WLAN method provides relatively good results but is heavily dependent on a location provider, which is mostly operated commercially.

[2] `http://airdroid.com/`

Table 1. Key values of the different positioning technologies

Technology	Method	Accuracy	Consumption	Availability
IP Geolocation	CBG	143 km		indoor & outdoor
	Octant	35 km		indoor & outdoor
	SLG	1.68 km		indoor & outdoor
GPS	GPS	<7.8 m	6.616 W/s	outdoor
	D-GPS	10 cm–55 cm	6.616 W/s	outdoor
	A-GPS	<7.8 m	6.616 W/s	outdoor
WLAN	Lateration	13 m–40 m	2.852 W/s	indoor & outdoor
	Fingerprinting	2 m–4 m	2.852 W/s	indoor
Cellular	Cell-ID	50 m–35 km	1.013 W/s	indoor & outdoor
	E-OTD	50 m–150 m	1.013 W/s	indoor & outdoor

Furthermore, the table shows that the energy consumption of GPS, WLAN and cellular positioning differs a lot and should be taken into account in the realization phase.

3.3 W3C Geolocation API

As location-based services become more and more important to daily life and widely known, a standardized set of always available methods to locate users is needed. Since HTML5 was near completion, the world wide web consortium (W3C), which is always thriving to allow better user experiences through standardization, seized the opportunity and integrated a new programming interface, called Geolocation API [10], into HTML5. The API features a set of JavaScript functions that delegate positioning to the user client and its attached devices. This way, any website can request accurate positioning information from any user. How the position is actually computed is not part of the specification and is entirely up to the software developers that implement the Geolocation API. However, the most common ways include GPS, WLAN- and cellular network positioning. It is important to keep in mind that the position determined in such a way is not necessarily the true position of the user, but might be spoofed or simply erroneous. In order to secure the user's privacy, the specification mandates that each position request from a website needs to be approved by the user, e.g., through an interactive dialog that asks for permission. In the case that the user declined the request, the browser or user client is not allowed to send the position.

3.4 OpenMobileNetwork

To become more independent from proprietary data providers, a lot of effort was made in the last years, resulting in the creation of open location datasets. The OpenMobileNetwork (OMN) [11] is one of the most prominent examples. Its goal is to map all existing base stations together with semantic data, like cell traffic and neighboring cells. Additionally the map is extended by semantically connecting these base stations with other objects in a linked data approach. Building on top of that basic framework, network usage can be monitored. Ultimately, this open, semantically structured dataset is to be integrated into the Linked Open Data Cloud (LOD)[3], which consists of already 295 other datasets that cover different themes of real world objects and semantics, allowing new and complex applications to be built with the available cell data. Currently, 7237 cells are contained in the OMN. However, the OMN is not a location server itself. It is rather a huge database of cell locations, cell ranges and other data. This data can be queried through a web service[4]. Such a dataset could also be used in this work as an alternative to the Google Location Server to infer smartphone locations.

3.5 Device Linking Concepts

In order to build a system that allows positioning information to be transferred from smartphone to any stationary device in a reliable and secure way, the different available concepts to link devices are analyzed. Currently, there are four major technologies that are used for data transfer: Universal Serial Bus (USB), Bluetooth, Near Field Communication (NFC) and standard IP over LAN/WLAN.

Universal Serial Bus. Prior to the universal serial bus, there was a variety of ways and different interfaces to connect peripheral devices to a computer, including RS232, SPI, I2C and PS/2 [12]. This heterogeneous environment was rather complex and cumbersome. In the early 1990s, a consortium of several IT companies, among them Intel, Microsoft and IBM, designed the USB standard: A single connection method to bring computers and peripheral devices together in an easy manner. In 1996, the specification of USB 1.0 was released. USB 3.0 [13] is the current version and supports up to 5 Gbit/s transmission rates. USB is realized as a cable-based master/slave communication that allows up to 127 devices to be attached to a USB host. Two different communication protocols allow file transfer between devices: mass storage class protocol (MSC) and media transfer protocol (MTP). The latter is currently used in many up-to-date smartphones and features several improvements in comparison to the older MSC, such as the file system being maintained by the USB device and not by the USB host [14].

Bluetooth. Similar to USB, Bluetooth's intention is also to unify previously existing technology and improve user experience and simplification in general [15] [16]. However, it is completely wireless. It was initially developed by the Swedish company Ericsson in 1994 as a radio frequency alternative to infrared-based ways to connect devices and is currently maintained and advanced by the

[3] `http://lod-cloud.net/`
[4] `http://www.openmobilenetwork.org:8080/sparql/`

Bluetooth Special Interest Group (SIG). The current version (3.0) allows data rates of up to 3 Mbit/s directly over BT and up to 24 Mbit/s over WLAN with Bluetooth only used as the network setup initializer. Bluetooth allows up to seven slaves to communicate with a master device in the 2.4 GHz band within a standard range of 10 m (Class 2) [17]. Bluetooth is supposed to work in many different scenarios. For each use-case exists a standardized specification, called profile, that defines how communication between nearby devices should look like.

Near Field Communication. Radio frequency identification (RFID) technology with contactless smartcards allows short range, wireless, one-way data exchange between devices. Near field communication is based on RFID, but allows arbitrary communication in both directions. It was developed by Sony and NXP Semiconductors (formerly a part of Philips) in 2002. Since 2004, an institution called NFC-Forum[5] is in charge of further standardizing and developing the technology. The underlying principle of both RFID and NFC is electromagnetic induction, realized by inbuilt metal coils, through which an electric current flows. The NFC device can measure the amplitude, frequency and phase of the received voltage of the electromagnetic field and thus it can be used as a carrier frequency for data transmission. NFC is also a master/slave technology and has a range of 10 cm on average (20 cm maximum) [18].

4 Realization Concepts and Implementation

This section describes the system requirements and conceptual models, as well as the main implementation details.

4.1 Requirements

The system is supposed to provide a range of features, which are listed below. The most important function is the ability to transmit a single position to the stationary device, to be used by a third party application.

This also includes the integration of the system into existing location-based services, which is needed for them to get access to the location data. This can be achieved by building an integration for modern browsers, as they are the gateway to most LBSs. However, other location aware applications that are not web-based do also exist. In order to support them too, a centralized point that provides location data on a stationary device would be useful. A location server on the Internet or even a local one could be a possible solution.

The information of a location should at least be composed of the most important values: latitude, longitude and accuracy. This information is the same as the data that is returned by the Google Location Server. Furthermore, additional information can and should be provided as well. This includes the time at which the position was computed, a position provider identifier, like WLAN or GPS,

[5] http://www.nfc-forum.org/

and a rating that indicates the quality, accuracy and reliability of the transmitted location. Furthermore, multiple energy profiles need to be supported. They allow an adjustment of energy consumption and accuracy. The system also needs to work without WLAN, because otherwise, a stationary device could already locate itself rather accurately. Open location data (e.g., provided by the OpenMobileNetwork) should be incorporated to form an alternative to proprietary providers.

Among non-functional requirements is a high compatibility with different stationary devices. This is important, because they come in many formats, especially regarding connection interfaces. Some devices might only feature USB, others Bluetooth, IP via Internet or LAN. It is notable that devices, which use NFC technologies are still very rare but become more important in the future, e.g., in the context of mobile payment. User privacy protection and position data integrity throughout the system are further requirements that need to be met. The system should also be highly automated and seamlessly integrated into existing devices and applications.

4.2 Conceptual Model

Several models were constructed that are suited to connect smartphone and stationary device, enabling LBSs on the latter by acquiring location data from the smartphone. Due to the nature of the connection methods between both devices, two different kinds of models are possible: direct connection via USB, Bluetooth or NFC and indirect linking over the Internet by using a TCP/IP connection.

Aside from this, three different stationary device application layouts were devised. The first consists of a dedicated client that connects to a nearby smartphone. The acquired location information is pushed into a file on the local file system, which can be accessed by LBSs. In the case of location-aware browsers, a plugin configures them to use this new location source. The second model is similar to the first one; however, it deploys a dedicated, local location server instead of a file. This allows an encrypted and authenticated way (using TLS) to provide LBSs with location data. In comparison to the location file, data manipulation is no longer possible. The third layout consists only of a plugin for browsers, which also handles the smartphone connection. Due to security restrictions in browsers, direct hardware access is not possible. Therefore, it is only usable in an IP connection scenario.

Considering all requirements, a direct connection model with a dedicated location server (model 2) was deemed to be the solution of choice (Fig. 1). The system should support as many different devices as possible. The general way to achieve this is to embed more than one connection technology. Bluetooth is chosen as the main transmission medium. This results from the circumstance that it is incorporated in many devices and phones. Even more important is that it allows a wireless connection, in comparison to USB, and thus features much better automation and background potential. There are some devices that do not support Bluetooth. In order to reach these devices, USB can be used not

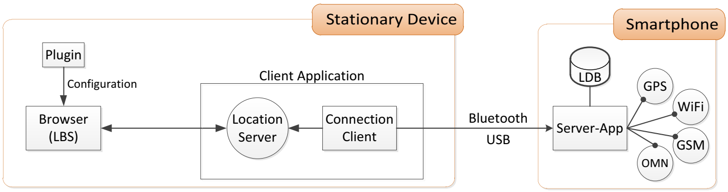


Fig. 1. Final connection model

only as a backup strategy, but also as a second transmission medium that allows, for example, the usage of docking stations.

4.3 Implementation

The system basically consists of a smartphone application (Android platform) that acquires positioning data and a client software on stationary devices that requests this information from nearby smartphones. Both components are described in this section.

Smartphone Application Overview. The application features a set of activities that allow users to access different functionalities and configure the app's behavior as they like. The structure of the underlying application components is depicted in Fig. 2. The entry point to the Android application is within the *Base Application* component. It contains three sub-components that are globally accessible and also need longer initiation times (e.g., AES encryption).

From the entry point, the primary activity (main menu) is started. Other activities like the Bluetooth server or settings can be accessed from here. *Location Handler* is the most important component, especially since it contains all the logic and mechanisms regarding location management. It allows starting and stopping the whole positioning process at any time, which is executed as a background service and reports location events back to the *Location Handler*. Depending on the user preferences, GPS, WLAN, cellular, OMN or a combination of them is used to acquire positions. Once an incoming position is registered, a special, time-interval-based position algorithm is executed that determines whether it should be kept or discarded. Aside from the normal location providers that are made available by Android, an extra component handles requesting location information from the OpenMobileNetwork.

Persisting location data happens in two different places. An encrypted SQLite database (SQLCipher [19]) is used as the main persistence component. Every other component gets its location information from this database. The only exception is USB location transfer, which uses MTP and thus cannot directly interact with the smartphone application. The *USB Location File Manager* overcomes this restriction by saving location data as encrypted files on the smartphone storage system. Users are able to specify how long location data should

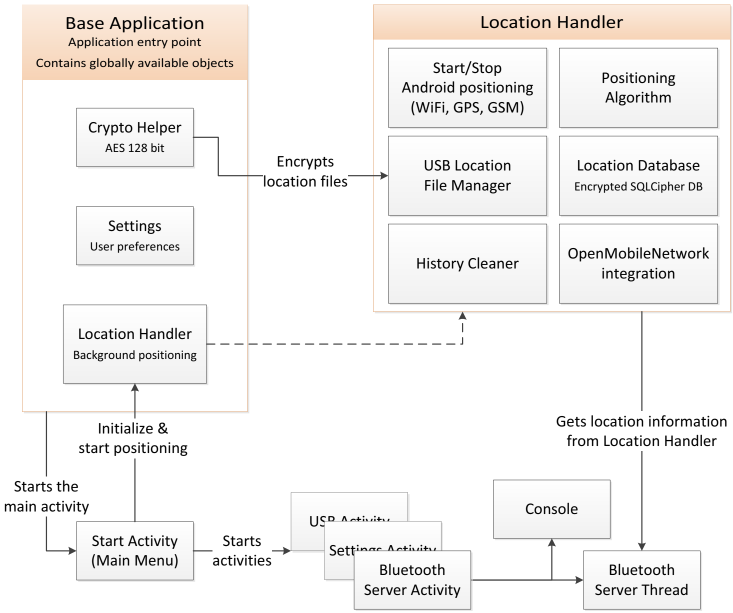


Fig. 2. Component overview of the smartphone application

be saved in the system. The *History Cleaner* component makes sure that both persistence modules are purged from older data. One component that makes use of the gathered location data is the Bluetooth location server that is integrated in an extra activity. It deploys a special server thread that listens to requests in the background.

Positioning Algorithm. After location updates are started, the current time is saved, which serves as a basis for time interval computation. Depending on the configuration, more than one location provider is activated. Some of them, like GPS, take some time to yield results, which might be as long as several minutes in extreme cases. Furthermore, each provider computes positions with different accuracy levels that increase from cellular positioning over WLAN to GPS. By executing multiple providers at the same time, coarse positions are generated very soon and can be replaced with better ones from other providers that take longer to do so. On the other side, this also means that a multitude of position fixes is generated and registered asynchronously. A positioning algorithm that is based on Android developer's algorithm [20] and extends it in several ways, filters all these positions for the best one. The developer's algorithm uses timeliness and accuracy to decide whether an incoming position fix is better than the previous one. However, multiple positions are often accepted at certain points in time. For example, this could be a GPS, WLAN and cellular position within a few seconds. This leads to a very cluttered location history. The main extension of the new algorithm (Fig. 3) to improve this behavior is the allocation of time slots, beginning with the moment that the positioning process is started.

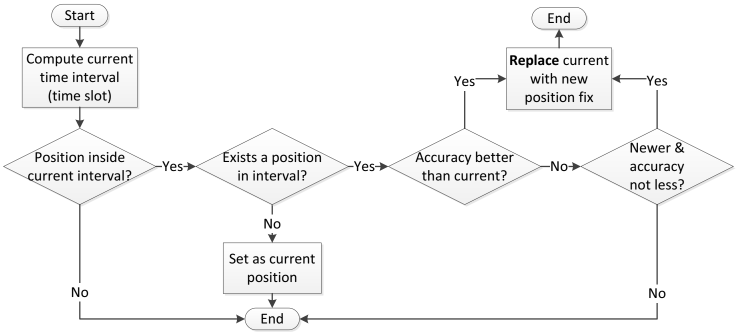


Fig. 3. Interval-based position replacement algorithm

Each slot is as long as the update interval and is supposed to contain exactly one position fix. That way, only a limited number of positions is saved over time, leading to a well-arranged location history. For example, after 60 minutes with an update interval of one minute, exactly 60 history entries should exist. Nonetheless, the primary objective is still to get the most accurate and up-to-date positions. Therefore, old positions need to be replaced by better ones if a time slot is already occupied. If the position does not fit into the current time slot, which could happen if it is too old, it will be discarded. Otherwise, it might be a potential candidate for that slot. If it is still empty, the new position is saved. Otherwise, the position with the best accuracy is determined. In case that the new position is better the old one is replaced. If not, then a replacement is only performed, if the incoming position is newer and at least has the same accuracy as the old one. This ensures that each time slot has exactly one position that has the best accuracy. By using time slots with a length equal to the update interval, it can be assured that the system always is in possession of the most up-to-date location information.

Stationary Device Client. The stationary device client (Fig. 4), programmed in Java, is the counterpart to the smartphone application. It is supposed to run continuously in the background, but should also be configurable and controllable by the system user at any given time. It consists of a graphical user interface (GUI) and a separate local location server. The GUI is the main access point of the application and features a range of adjustable settings, as well as a console that allows program and location status tracking. Aside from that, the connection client, which contacts surrounding smartphones and requests location data from them in 10 second intervals, can be started and stopped from here.

Incoming location information is forwarded to the location server, which acts as an endpoint that other applications on the stationary device can approach in order to get location data. The whole application can run seamlessly in the background.

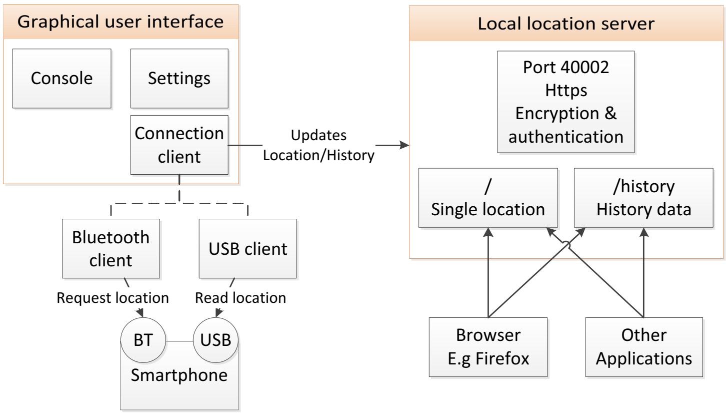


Fig. 4. Component overview of the stationary device client

The local location server is powered by the open source software Jetty [21]. It runs on localhost and exposes two URLs that can be accessed only by applications on the same machine over HTTPS. It is therefore not possible to leak sensitive information to other entities in the network. The purpose of this server is to mimic a location service like the Google Location Service (GLS), with some minor adjustments. This means that sending a query to the server should result in a reply that contains location information represented in JSON. This specific functionality is administered via the first URL, which is the servers root: */*. The server also makes history information available through a second endpoint: */history*. Running on port 40002, which is a non well-known port [22], HTTPS requests are served. The server provides its own X.509 certificate. Connections over this port are encrypted and authenticated.

5 Evaluation

The smartphone to stationary device connection system was tested and evaluated. The single tests are representative to a point that allows at least indicating trends and an approximate appraisal. Integration into browsers was tested with the most recent Firefox versions, i.e. 19 and 20. Being able to toggle between the standard GLS-based positioning and the smartphone system is working without problems. When the latter is activated, Firefox contacts the local location server upon website request and forwards the acquired position to the inquiring website. This never failed during all tests. The only problem that sometimes occurs is that Firefox tends to cache positions from time to time, resulting in the need to reload the website. The tests and evaluations further concentrate on two areas: position quality and transmission process. The former deals with the kind of positions that are provided when entering buildings. The second area of tests is about Bluetooth position transmission from smartphone to stationary device.

Positioning When Entering Buildings. The system is designed as an automated background service that might operate for a longer time without consuming too much energy. As a consequence, positions are only acquired after certain time intervals. This causes the position that is actually transmitted at a certain point in time, e.g., the moment when arriving at a computer, to vary a lot regarding timeliness and accuracy. In the test scenario, a person with a smartphone arrives at a university campus. While walking to the actual workplace, the smartphone is keeping track of its position. Walking this path takes roughly five minutes. The intention is to transmit the position to a computer at the work place near the entrance of a building (inside). Two different position values are collected. One is the last location that was recorded outside. This should simulate situations in which no cellular reception or WLAN is available inside the building. In that case, the last known location transmitted to the stationary device is the most recent history value of the smartphone. The other value is the first position acquired after arriving at the stationary device. This represents the case that the smartphone does have at least cellular reception and possibly also detects WLAN access points inside. The test is repeated several times, with four different application configurations. Figure 5[6] depicts the measurements of one configuration setting that uses all three positioning technologies, as well as an update interval of one minute.

Outdoor positions for all measurements deviate from the real one between 63 and 130 meters on average, depending on the configuration used, while being connected to the Deutsche Telekom D1 network (GSM-900 MHz). The positioning results are good enough to locate the stationary device in a small area, which is sufficient for most LBS scenarios. Table 2 presents an overview of the accuracy results of all four configurations. With a continuous GPS observance, the accuracy for outdoor positions could be improved a lot. At the same time, this would also cause the phone battery to be drained very quickly and is therefore probably not desirable for most use cases, even though it could be implemented very quickly. Longer update interval times are found to be useful to a certain point. In order to decrease energy consumption, a higher value is advisable, but this also causes less accurate results. A common situation, in which intervals of more than 2 minutes make sense, is when a person is staying inside a building for a longer time and moves from one stationary device to others over time. In this case, it is likely that the devices are close together, leading to only minor position changes, which causes even positions from several minutes ago to be sufficient in terms of accuracy.

[6] Map data ©2013 GeoBasis-DE/BKG (©2009), Google.

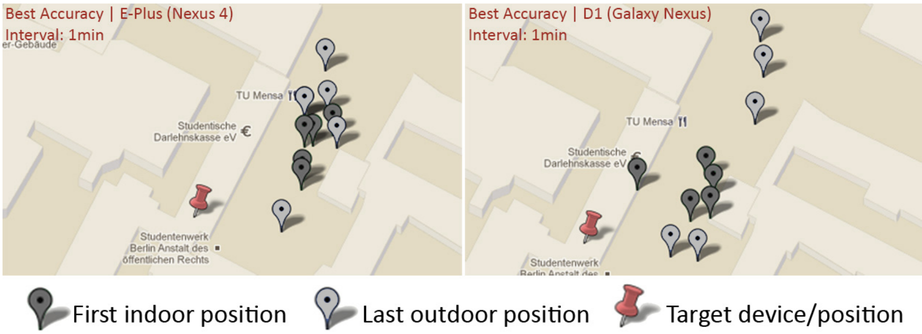


Fig. 5. Measured outdoor (light) and indoor (dark) positions for two different cellular networks (E-Plus 1800 MHz and Deutsche Telekom D1 900 MHz)

Table 2. Results for last outdoor and first indoor positions in the Telekom D1 network

Telekom D1 (Galaxy Nexus)	Deviation (in m)	Best accuracy Interval: 1	Low energy Interval: 1	Low energy Interval: 2	Low energy No WLAN
	Min	38	35	39	190
Last outdoor fix	Max	92	138	195	190
	Average	63	61	130	190
	Min	25	26	26	190
First indoor fix	Max	46	32	190	190
	Average	39	28	61	190

Bluetooth Transmission. Even though Bluetooth is a standardized technology that is supposed to provide fast and secure connection setups, there are still some things that need to be done before data can be exchanged. This includes the enforcement of authentication and encryption key exchange. The former is only required once. After devices are paired, they can initiate connections without further user interaction. In a test suite, the process of moving into range of a stationary device (a computer running 64 bit Windows 7, Intel processor) with a smartphone for the first time is examined. For this purpose, the client application on the stationary device and the Bluetooth server on the smartphone are started while still being out of range. Next, the smartphone user moves to the computer as if he wanted to use it for a task that involves location awareness. If only one smartphone is near, the maximum time until the location is transferred is 53 seconds after entering the stationary device's Bluetooth range. On average, the location arrived at the stationary device after 33 seconds. Those values only apply for the cases in which pairing is still necessary. Assuming that a smartphone is already paired with a stationary device, the time decreases to roughly 12 seconds, which is the fixed time needed to scan for Bluetooth devices. After the first location transmission, it only takes less than one second for subsequent transmissions.

6 Summary and Outlook

The aim of this work was to devise and implement a system between smartphone and stationary devices, allowing automated transfer of location data and seamless integration into existing applications, such as location-based services. The first step to this goal was to find, analyze and evaluate different device connection technologies. In the next step, different realization models were devised and evaluated. The best one was implemented. Two device connection methods were integrated into the smartphone application: Bluetooth and USB. Furthermore, the OpenMobileNetwork was used as an alternative to proprietary location data providers. Test results showed how the system behaved when moving into a building and which positions were reported in that moment. Different application settings were applied to show how they affect accuracy in this scenario. The results indicate that a stationary device can reliably be pinpointed to a small area, which is much better than the currently used IP geolocation approach.

In future work, some aspects of the system could be further improved. In the current state, USB transfer is only implemented for Windows systems. Supporting Linux and other operating systems in this area can be achieved by integrating additional USB libraries. The system could also be extended to other stationary devices, e.g., ticket machines and cashiers. Concerning accuracy, several improvement possibilities arise. The OpenMobileNetwork operators plan to extend the database to WLAN access points in the future. Using this data, WLAN positioning with an underlying open data set source could be performed and the accuracy could thus be increased to 13-40 m. Another approach would be to use more sophisticated algorithms for lateration. Several alternatives exist for that purpose. The most promising ones are based on Taylor series calculations, non-linear regression and Kalman filtering. Since most stationary devices are located inside buildings, an improvement could be made by integrating specialized indoor positioning systems. Using such an approach, the accuracy could be as good as room-level or even in the centimeter range.

Additionally to these improvements the system could also be extended to other application scenarios. In a setting were doctors on a hospital premises are moving around during a working day, user-IDs can be used to tag the medical records and diagnoses together with location and time information. In another scenario this system can improve the anti-theft system of a company when the smartphones are used to check whether a stationary device is in place while being in close proximity to it or using it.

References

1. Brimicombe, A., Li, C.: Location-based services and geo-information engineering, vol. 21. Wiley (2009)
2. Bareth, U., Küpper, A., Ruppel, P.: Geoxmart-a marketplace for geofence-based mobile services. In: 2010 IEEE 34th Annual Computer Software and Applications Conference (COMPSAC), pp. 101–106. IEEE (2010)

3. Küpper, A.: Location-based Services - Fundamentals and Operation. John Wiley and Sons, Ltd. (2007)
4. Atallah, M.J., Blanton, M., Frikken, K.B.: Efficient techniques for realizing geo-spatial access control. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, pp. 82–92. ACM, New York (2007)
5. Ardagna, C.A., Cremonini, M., De Capitani di Vimercati, S., Samarati, P.: Access control in location-based services. In: Bettini, C., Jajodia, S., Samarati, P., Wang, X.S. (eds.) Privacy in Location-Based Applications. LNCS, vol. 5599, pp. 106–126. Springer, Heidelberg (2009)
6. Uitz, I., Harnisch, M.: Der qr-code - aktuelle entwicklungen und anwendungsbereiche. Informatik-Spektrum, 339–347 (2012)
7. Gueye, B., Ziviani, A., Crovella, M., Fdida, S.: Constraint-based geolocation of internet hosts. IEEE/ACM Transactions on Networking, 1219–1232 (2006)
8. Wong, B., Stoyanov, I., Sirer, E.: Octant: A comprehensive framework for the geolocalization of internet hosts. In: Proceedings of the NSDI, vol. 7 (2007)
9. Wang, Y., Burgener, D., Flores, M., Kuzmanovic, A., Huang, C.: Towards street-level client-independent ip geolocation. In: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, NSDI 2011 (2011)
10. World Wide Web Consortium: Geolocation API Specification - W3C Proposed Recommendation 10 May 2012 (May 2012)
11. Uzun, A., Küpper, A.: Openmobilenetwork: Extending the web of data by a dataset for mobile networks and devices. In: Proceedings of the 8th International Conference on Semantic Systems, I-SEMANTICS 2012. ACM, New York (2012)
12. Hulzebosch, J.: USB in der Elektronik. Franzis Verlag (2008)
13. USB Implementers Forum, Inc: USB Revision 3.0 Specification (2008)
14. USB Imeplementers Forum, Inc: Media Transfer Protocol v.1.1 Spec and MTP v.1.1 Adopters Agreement (April 2011)
15. Labiod, H., Hossam, A., De Santis, C.: Wi-Fi, Bluetooth, Zigbee and WiMAX. Springer-Verlag GmbH (2007)
16. Bluetooth SIG Inc.: Fast facts — bluetooth technology website (January 2013), http://www.bluetooth.com/Pages/Fast-Facts.aspx
17. Gessler, R., Krause, T.: Wireless-Netzwerke Für Den Nahbereich: Eingebettete Funksysteme: Vergleich Von Standardisierten und Proprietären Verfahren. Springer DE (2009)
18. Finkenzeller, K.: RFID-Handbuch: Grundlagen und praktische Anwendungen von Transpondern, kontaktlosen Chipkarten und NFC. Hanser Verlag (2012)
19. Zetetic LLC: Sqlcipher - encryption for sqlite (2011), http://sqlcipher.net/
20. Google Inc: Obtaining the current location — android developers (2012), http://developer.android.com/training/basics/location/currentlocation.html
21. Eclipse Foundation: Jetty (2013), http://www.eclipse.org/jetty/
22. Touch, J., Kojo, M., Lear, E., Mankin, A., Ono, K., Stiemerling, M., Eggert, L.: Service name and transport protocol port number registry (April 2013)

Rule Based Preferential Context Sharing in Location Aware Mobile Advertising

Vikash Kumar[1], Joachim Zeiss[1], and Marco Happenhofer[2]

[1] The Telecommunication Research Center Vienna (FTW), Austria
[2] A1 Telekom Austria AG, Austria

Abstract. This paper introduces a novel context management system for mobile phone users that allows them to control which part of context data they want to reveal to their service provider and which part to keep on their device. With the use of semantic rules, users can set their custom privacy and preference profile in a way that targeted advertisements can still be served despite them not wanting to reveal these details to the provider. By shifting some part of the reasoning task to user device, the mobile service provider is still able to provide highly targeted advertisements by combining the private and publicly available parts of user information.

Keywords: Semantic rules, Privacy, Context management, Location aware advertisements.

1 Introduction

The current trend in All-IP promises a lot of new services for telecom (telco) operators who are trying to compensate their revenue declines in voice minutes and data networks with value added services. Presence and Location Based Services (LBS) are often suggested as one of the candidates for such services. However, so far the LBSs have not lived up to their promise completely. The reasons are not obvious, but both these types of services have a common thread, the possible lack of privacy protection, which prevents privacy aware users to consume those kinds of services.

Presence is the expression of technical capabilities and the social willingness to communicate as standardized in the Internet Engineering Task Force (IETF) and Open Mobile Alliance (OMA). This information is restricted to authorized users and is already realized free of charge by most Instant Messaging (IM) and Internet telephony tools available on the web. This makes it more difficult for the telco service providers to compete with such free of charge services and to earn money with the presence data. Currently, there are two well known approaches for enriching presence information and charging a fee on it. The first one is to extend the presence data with additional information about preferences, current location of the user, his calendar and so on. The second approach aggregates presence data from several users to derive high level presence from it. For example a public land mobile network (PLMN) could aggregate the location areas of

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 64–78, 2013.

the mobile phones in its network and could derive traffic jams in its coverage area. This example simply illustrates the knowledge gain due to smart presence aggregation, but it also ensures the privacy of each person is protected. This is because with the information "there is currently a traffic jam on motor highway X", we are not able to derive the location of any dedicated subscriber in the network.

Presence data like location, comments like "on vacation", age, sex, preferences, current activities and so on are personal and highly sensitive and should be protected from unauthorized access. This kind of data is of interest to several (legal and illegal) businesses, mostly for the marketing departments which create profiles, assign users to profiles and deliver them personalized advertisements based on their profiles. This business case is quite well known from the Internet. However, presence data like location, is much more real-time than static preferences and opens up possibilities of new business models for advertisements. In traditional advertisement based services, the profile data of each user is stored at the service provider's database and it uses this information to select an appropriate advertisement. Enriching such a traditional service with real-time presence data requires continuous updates of this data at the service provider's database. Unfortunately, there is often the latent thread due to which a service provider, intentionally or unintentionally, distributes the personal presence data of its users to others. As this data gets public, it might harm the reputation of those users. Therefore, privacy sensitive subscribers tend to avoid sending their personal private presence data to service providers. In most cases, the retrieval of location data is an essential precondition for fulfilling the location specific service functionalities and therefore users who withhold such information can not participate in location based services.

In this paper, we address the problem of providing personalized targeted advertisements to users which currently requires access to their personal sensitive presence data and might compromise their privacy. Our approach suggests to push the advertisement selection logic to the user's device, that is alone responsible for their personal data, instead of sending it to the advertisement service provider. This ensures that privacy sensitive data remains at the user's device and is not transmitted over the network or to other domains.

The rest of this paper is organized as follows. In section II we present the related work in this area, in section III we describe the approach we follow to achieve our objective and in Section IV we describe the architecture and implementation details of the complete system. In section V we describe our evaluation setup and corresponding results for this system. Finally the last section provides a conclusion to our present work and sketches our future plans in this area.

2 Related Work

There has been considerable interest in the semantic community on modelling an intelligent framework for context aware ambient devices. They have been broadly classified in [1] based on the underlying reasoning approaches viz (a) Ontological reasoning, (b) Rule-based reasoning, (c) Distributed reasoning and (d) Other reasoning techniques. In [2] an architecture for supporting context aware systems *CoBrA* is proposed which gathers context data from various devices and stores it in a central broker where all the relevant reasoning is performed. In [3] a framework for context aware service prioritization and triggering of events is presented with an emphasis on modeling contexts and rules. In [4] a shareable context aware semantic model for mobile devices is described targeting mainly pedestrians using an interactive map based interface for guidance. Context information is modeled in an RDF Ontology and queries expressed in RQL. SOCAM [6] provides a middleware architecture for managing context information in the use case of Smart Home. A change in context leads to actions triggered by a set of rules provided by the service developers. The focus however is completely on the middleware layer and all the reasoning must be centralized. There is no provision for context processing in a situation where the user does not want to reveal his/her exact location (or other context data).

While each method has its merits, most of these approaches focus more on *imperfection* and *uncertainty* of context data and do not take into account user privacy as a prime objective. In our view, user being the most vital stakeholder in context based services should be able to decide the type and granularity of information he/she would like to reveal to various service providers. Semantic rules is one of the prime techniques using which we intend to achieve our objective of customized context sharing. Some notable works relating to semantic rule based data management can be found at [5,8,9]. Our work takes inspiration from all these while delving into the Location Aware Campaigning(LAC) use case which is different from the ones chosen by them.

3 Approach

While social networking and content mining sites like Facebook, Google, Amazon, etc. provide very accurate personalized recommendations to a user, they make use of customer's private data for achieving this. The user has to share a bunch of personal data during registration at these sites and then they also observe customer behavior based on the sites they visited, products they browsed, etc. The user has little or no control over this as also knowledge of which part of his personal data or behavior is being passed by the companies to third party service providers. Our aim is to achive the goal of personalized ad recommendation to user while allowing them to decide what part of their personal data should be used for this purpose.

The approach we follow in this paper is mainly with the twin objectives of preserving user privacy while at the same time finding a way to extract the useful data needed for personalized advertisements. To achieve this, we model all the user profile information and campaign data in a structured format using ontologies. Users can choose which part of their personal profile information will be stored on their mobile device and which part on the telecom service provider's server. The campaign/deal provider on their side can express the targeted group of consumers they want their advertisements to reach using semantic policies.

More specifically, they can specify the age group, interest group, sex, ethnicity, etc. of the users they want to focus their attention on. This system also allows telco service providers to specify their business policies like according special privileges to their "platinum members" both on the subscriber side as well as the advertiser side.

In our use case, we focus especially on LAC information as a showcase of how a user can choose to express the exact granularity of location he/she wants to reveal to third parties and still receive the location aware campaign data in the best possible way.

4 Implementation Methodology

Our prototype implementation consists of three main modules: (a) client side module enabling reasoning over data available on user's mobile device (b) a server side implementation of the semantic reasoner as well as ontology and (c) advertiser's console enabling merchants to annotate their webpages with microdata and set policies for their target audience, etc.

4.1 Architecture

As depicted in Figure 1, Location Aware campaigning is distributed across client software, LAC service provider (server) and merchants. All of the three may apply their own policies: The LAC server has policies for campaign pre-selection and applies them while generating the campaign list for the user; the merchants may have policies related to time of day or product offerings and the user has policies based on location, buddy lists, interests, calendar entries, etc. All parties

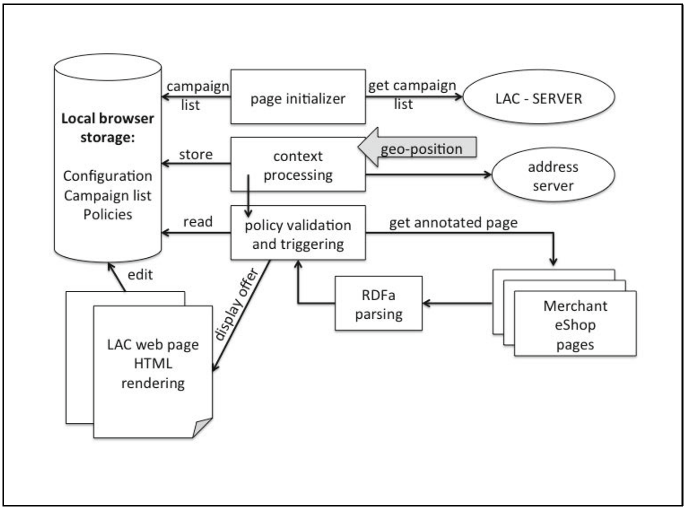


Fig. 1. LAC Service Setup

may keep their reasoning and data private or may share information or policies if they are willing to.

Our LAC use case highlights the relationship between a local browser agent web page or web-app on client side, a network (server) side agent and the campaign provider or merchants' web presence. Other independent entities can be mashed up for address mapping or web hosting. Both the LAC client and the LAC server use respective information available with them to decide about the campaign to be presented to the user. Entities of the architecture are:

- LAC client, as stand alone web-app or integrated part of a web page
- LAC server, mediating the campaigns between user and merchants
- Merchants' campaigns, represented as online shop web pages annotated with campaign relevant information

The following walk through explains the generic functionality of our prototype:

1. The LAC server pre-selects the number of users to send configuration info for campaigning. This decision is based on several types of information like user's online status, subscription, time of day, interests and other policies. Merchant's (campaign provider's) policies may help to further reduce the number of users to be informed. In the current prototype implementation, it is left to the LAC client to first send a request to the LAC server to initiate the necessary configuration sending which of course will only happen if the user and the available campaigns fulfill the conditions of the campaign policies. When the LAC web page (or web-app or widget) is loaded in the client browser, it sends a LAC configuration request to the server, which will trigger the aforementioned campaign selection actions.
2. Now that the configuration has reached the client side, it is stored for further processing. The reasoning process is triggered if there is some change in context data or user's policy data. If the outcome of this reasoning process concludes that a particular set of campaigns should be retrieved or that the user has entered a new campaign location area, actions are taken to obtain data from the respective merchants' URLs.
3. Based on reasoning over available campaigns, the LAC client sends an HTTP request to obtain the specific campaign data to be displayed in the browser.
4. On reception of the merchant's web page, the user client parses the containing RDFa (Resource Description Framework in attributes) [12] or Schema.org[1] annotations and hands this information over to the reasoning process again to integrate this extra information coming from the merchant into the decision process. If user policies say so, then the RDFa/Schema.org information about offers is rendered inside the LAC web page. This is particularly useful for page rendering if the LAC client runs on a mobile device. Also, specific features of mobile clients like making a phone call or sending an SMS could be used.

[1] `www.schema.org` is a vocabulary using which developers can add structured data markup on their webpages.

Steps 2. to 4. above are repeated depending on campaign polices and context data at client side until configuration is refreshed by the server.

For the scenario of our prototype, three different campaigns will be defined: A sportswear market chain (in our example Intersport) and two different Electronics vendors Niedermeyer and Saturn.

For example, the sportswear maker as a registered merchant will publish its campaigns at the LAC service/server in form of its stores' physical addresses and related URLs to be browsed to in case the user shows interest in the campaign. Further information for server side reasoning about campaigns, goods and prices can be provided as well. This information could be uploaded to the LAC server or be requested by the LAC server when distributed reasoning over the number of users to be informed about campaigns is in progress. The merchant would also semantically annotate its web pages behind the URLs given to the campaign server with Rich snippets (more details in Section 4.3). This information is later parsed by the LAC client to decide if the campaign should be presented to the user and if so how should it be done.

The LAC service, i.e. campaign coordinator, will perform its own reasoning. For example, if the campaigner is a mobile phone operator, it would filter campaigns related to competitors. If an electronics discounter would offer mobile phones with tariffs and contracts from several providers the mobile operator could filter those offers related to its own business.

As the user has subscribed to LAC service and his/her server side policies declare interest or do not explicitly deny interest, it receives the LAC configuration data containing merchant's store locations. The campaign list is retrieved when opening the LAC client directly or from within a web page. The list of shops is reduced to a certain area, e.g. all registered shops in Vienna, because the user (LAC client) requests data for that broad area, not wanting to reveal his/her exact location.

The first reasoning step is the pre-selection of clients to communicate with for campaigning. This takes place at server side. The result is mainly a list of preselected address-to-campaign URL mappings. Other policies for pre-selections could be PEGI checks for movie or video game offers or time of day based filters because opening hours of shops in campaigns differ.

As the client is entering or exiting location, areas of registered stores reasoning is initiated on the device. Triggered by location change, the LAC client runs through the user defined, locally stored user policies and information to decide if campaign information should be displayed.

For example, the user is not interested at all in buying sportswear right now but defined his/her interest in buying a new smart phone with a certain price maximum and an electronic store annotated such an offer in its campaign web pages for certain shops in Vienna. In this case, the user would be informed as he passes by. Another example could be a user actively asking for help in buying Tennis clothes for a new course starting on the coming weekend. The LAC client could point him/her to an appropriate store to do so.

Other policies that are conceivable but have not been investigated in detail are: Birthday or Valentine's day reminders, search lists for special offers on the LAC server, local campaign blacklists or books and DVD wishlists. Modern HTML5 based mobile application run times may allow access to such context and personal information management data.

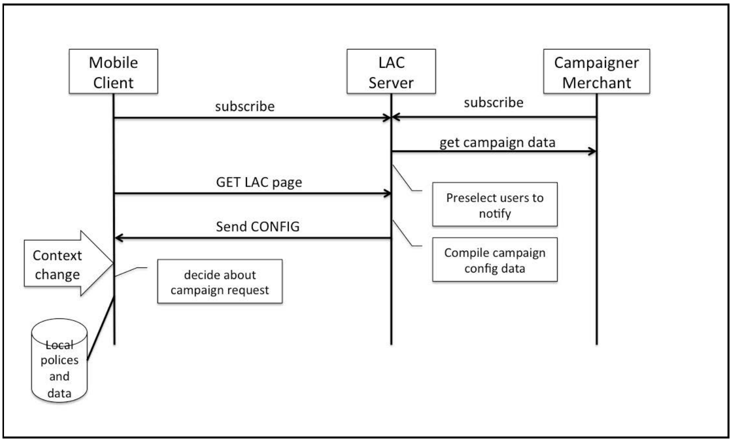


Fig. 2. LAC Service Messages

Most of the information necessary for those policies like location, interests, birthdays or blacklists are rather private and the user may not want to give it away. Therefore, his/her information is kept locally on the phone and only used by the LAC client not revealing it to the server. Also, the geo-coordinates of the user are kept private in this case. The local reasoner simply asks the server for campaign configurations.

The only information communicated to the LAC server would be the request for a list of campaigns in a certain area, or for a list of a specific category of campaigns without specifying any area details at all. However, if the user allows, details of a more or less abstracted location information could be sent to let the server determine appropriate campaigns. The client sends campaign data requests "ad-hoc" to the server and the server, after finishing its part of reasoning, provides the client with a URL pointing to the campaign data to be displayed. Figure 2 depicts the message flow for a typical campaign request. Merchants keep the control and responsibility of their offers and changes are applied instantly on their own web pages. There is no need for further information propagated towards a LAC service provider. Figure 3 shows the LAC web client for mobile clients in action.

4.2 The LAC Client

As a first step, the LAC client needs to be configured to use the correct address and user ID for sending requests to the LAC server. After that the user may define basic policies for location area triggering and campaign types the user is interested in. The location area policy defines if campaigns in same street, quarter, city of country of the user's current position should be checked. The interest policy further filters types of campaigns to be selected. Whereas the location information comes from the local device, the type of the campaign is obtained from the merchant's campaign web page.

After the user starts the LAC web app or browses a web site containing the LAC client, a request is sent to the LAC server asking for a campaign short list. Positioning is started using the browser's navigation javascript API. The LAC client implements a callback function for this API to be informed on location changes. This callback retrieves the current address from OpenStreetMap[2] servers. Once the address is available, based on the location area policy, the street name, the district, the city name or the country name is used to determine if positions of shops match the current position of the user. For example, if the policy says city wide campaign information, all campaigns of shops in the same city from the previously obtained list of campaigns will be retrieved for the further reasoning steps.

If a campaign matches a location according to the above described policy, the client requests the campaign's URL and loads it to an invisible HTML div-tag. Then the content of this div-tag is parsed for RDFa/Schema.org annotations. The “interests” policy of the user is then evaluated against the meta information of the campaign pages. If the type of offer matches with the user's interest, the parsed data is used to render campaign information and display it to the user.

4.3 Server Side Processing

The central server in our system is hosted by a telco provider. It holds information containing publicly shared part of user profiles, their public preferences in form of policies, policies of the telco provider as well as the policies and campaign information from various vendors[3].

An ontology is used for storing user profiles and campaign details. User preferences are stored in form of SWRL rules [10]. The campaign policies from the vendors as well as the policies of telco providers are also stored using SWRL. The reasoning is carried out using HermiT reasoner [11]. Message exchanges between the mobile client and the server is done via HTTP GET calls using Json data served by the server side servlets. GoodRelations ontology [7] has been extended and employed for storing the campaign information as well as user policies. The DL expressiveness of the eventual ontology is *SHI(D)*.

The choice of GoodRelations ontology for representing vendor deals was based on industry best practices where major Internet based companies are either

[2] `http://www.openstreetmap.org/`

[3] Please note that in our architecture, the campaign details may either be residing on the telco server or directly on the merchant's web page. This makes little/no difference in the message flow and hence we use both possible campaign locations interchangeably in our discussion.

Fig. 3. LAC mobile client web page

recommending or themselves using the ontology in their own systems. Search engines like Google and Yahoo officially recommend GoodRelations for sending structured information for Google Rich Snippets to Google and for SearchMonkey to Yahoo respectively[4]. We therefore argue that although the research in our project does enforce the use of a common vocabulary by the concerned stakeholders, presence of GoodRelations makes it easier to adopt such a vocabulary which is fast becoming a de-facto standard for classifying structured data on e-commerce sites. It is also compatible with various popular formats like RDFa, microdata [13], RDF/XML, etc. A motivation for the vendors to add such metadata snippets to their product details would therefore also be to increase their Search Engine Optimization (SEO) besides enabling third party functionalities like APSINT LAC [16] to be introduced on top.

Policy Based Filtering of Campaigns. Semantic policies are used for selecting the advertisements that should be targeted to a particular user. This decision is based on three parameters viz.

(a) Preferences set by the user
(b) User profile information including various forms of context data available with the telco provider and the user
(c) Filtering policies set by the vendor as well as the telco providers themselves

[4] `http://wiki.goodrelations-vocabulary.org/References`

User preferences and filtering policies are set by using SWRL rules on server side while the user profile information is stored in the Ontology. Some parts of user profile information that a user doesn't intend to share resides on his/her own native device. In this case rules are represented in the more intuitive N3 language and not SWRL aiding in simpler rule handling by the client while also catering to limited reasoning capability of these devices.

For location aware campaigning, we use location details of a user available on his/her device that may or may not be shared with the telco provider. Location in our data formalization is represented by four parameters viz: (a) cell id, (b) location area code, (c) latitude and (d) longitude. The cell ID and location area code can be roughly described as representations of the telco provider's radio network area of a city. A city's radio network area consists of several location areas and each location area is further subdivided into cells with unique IDs (both represented as integers).

User specific policies could describe the products/product category a user is interested in (e.g. wishlist) or their privacy policies describing what data they want to share with the service provider and what not. A typical user preference expressed as rule on the server side will look like[5]:

```
Offering(?x), Cellphone(?z), includesObject(?x, ?y),
  typeOfGood(?y, ?z), ActiveUser(John, true) ->
  EligibleOffering(?x)
```

This rule expresses a typical user side preference that says: If active user is John and the current offering belongs to the Cellphone category, then classify the offer as "Eligible" to be sent to the user. An extension of the GoodRelations ontology has been done to accommodate user information about user profile as well as include SWRL rules for expressing their policies.

The telco providers can similarly set up rules to control the category of their customers that receive a certain category of their advertisements. An example scenario could be filtering adult content being sent to children under 18 years of age. This example case is described by the rule below:

```
ActiveUser(John, true), hasAge(John,?n), greaterThan(?n, 18),
   Offering(?x), includesObject(?x, ?y), typeOfGood(?y, ?z),
  qualitativeProductOrServiceProperty(?z, ?m), category(?m,
  ''Adult'') -> EligibleOffering(?x)
```

Another interesting rule can be set up by the user to assert that his/her exact location (like street name, house number, etc.) be hidden from the operator and only a more coarse location (like current city) be revealed to them. In such a case, the user will get a list of all advertisements from his/her current city rather than from a specific location as represented by the following server side rule:

[5] Please note that for ease of understanding, some of the rules represented here are presented in a simplified form and actual rules in the system may be much more complex involving more entities from the ontology.

```
Offering(?x), Laptop(?z), includesObject(?x, ?y), typeOfGood
  (?y, ?z), availableAtOrFrom(?x, ?l),
  hasGlobalLocationNumber(?l, ?n), withinCity(?n, ‘‘Vienna’’)
   -> EligibleOffering(?x)
```

After getting a list of all the relevant advertisements in the city on their personal device, the user in the next step can ask for a specific campaign in their immediate neighborhood directly from the campaign URL thereby not letting the telco provider know of his/her exact location. Client rule for retrieving the exact campaign in this case will be[6]:

```
Offering(?x), availableAtOrFrom(?x, ?l),
  hasGlobalLocationNumber(?l, ?n), withinStreet(?n,
‘‘Kleingasse’’) -> EligibleOffering(?x)
```

4.4 Advertiser's Console

The advertiser of a deal in our system can use the publicly available GoodRelations snippet generator tool [14] to create rich data snippets about their deals in RDFa format. The advertiser just has to fill in an HTML form with the information about which rich snippets are to be generated (company/store/product/service) and then paste the generated code in their own XHTML/HTML5 code. In the present version of our system, they can use Protege [15] or any other semantic rule editing software to construct custom SWRL rules.

5 Evaluations

The evaluations of APSINT LAC framework have been carried out to assess the system performance and scalability along the following parameters: (a) number of users, (b) number of advertising campaigns, (c) overall number of concepts in the semantic dataset.

5.1 Setup

To evaluate the system performance, we established a LAC central server on an Intel core i5 machine with 2.4 GHz processor speed and 4 GB memory. Galssfish server 3.1.1 was used to deploy the server application and was accessed by a client running on a different machine.

Performance indicators were captured using servlet filters for each phase of testing. The server response time was logged as affected by an increase in the number of campaigns in the dataset followed by an increase in the number of users. The number of campaigns were increased in steps of 5 and number of users in discrete steps of 10, 20 and 50. For each step, two standard requests

[6] In the actual system, this rule will be represented in N3 format.

corresponding to reasoning based on server side rules (one each for location filtering and user preference) were sent to the system. Three readings were taken for each type of request sent and averaged for coming up with final results.

5.2 Results

As seen in Figure 4, the performance of the system is acceptable and scales well with the increase in number of advertisements. However, there is a definite dependency on the number of advertisements over which the system has to reason upon and as this increases, the reasoning process becomes slower (or requires more powerful machines). This part of system performance is dependent on the underlying reasoner used in the process and having used one of the latest and more efficient reasoners currently available (HermiT), we argue that our approach of distributing the reasoning task across devices makes the overall performance of this framework better. Besides distribution, filtering and processing of the campaigns on the server side to provide a smaller set of streamlined campaign data to reason over on the client will be more efficient. Furthermore, as the state-of-art in semantic reasoning becomes faster, the performance of this system will also improve as far as server side reasoning is concerned.

Figure 5 shows the performance of the system as a function of increase in the number of user profiles. Not much dependency of the system is seen on the increase in number of users as compared to the case of number of campaigns. This might be due to the fact that the campaign descriptions used in our tests were much more elaborate than the user profile descriptions resulting in lesser number of additional axioms per user compared to axioms per campaign. Some back and forth variations of the graph (seen as its zig-zig shape) can be explained by the varying network response times which affected the response more than the effect of the increase in the number of users. This justification was further

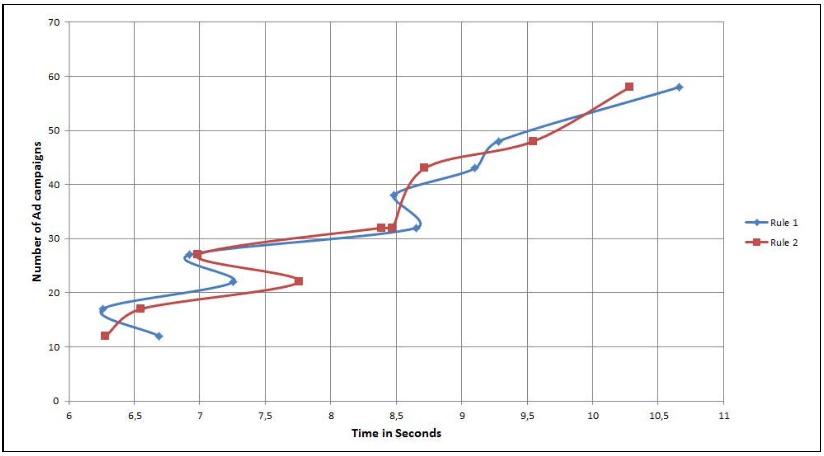


Fig. 4. System Performance based on Campaign numbers

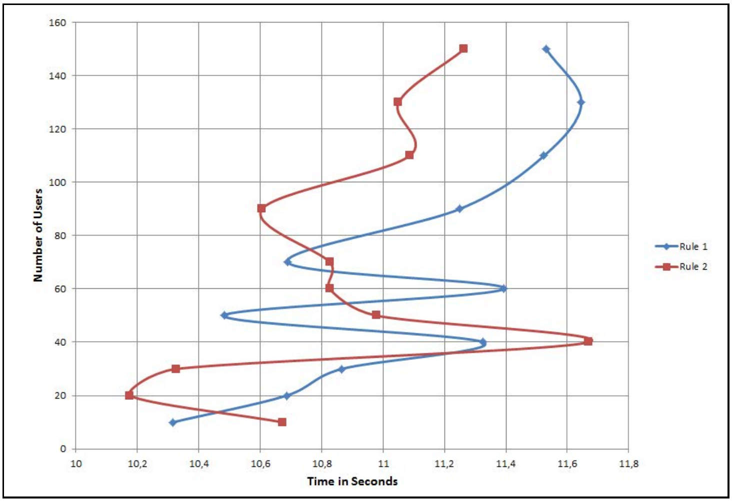


Fig. 5. System Performance based on Number of Users

proved when we repeated the experiments several times eventually to reach the same results (i.e. similar graph shape) every time.

To give a clearer picture of system performance based on ontology based metrics, in Figure 6 we show the system's reasoning performance as a function of the number of axioms and individuals in the underlying ontology. This further proves our conclusions derived from Figure 4, i.e. the performance of the system is sensitive to the increase in the number of semantic entities in the dataset.

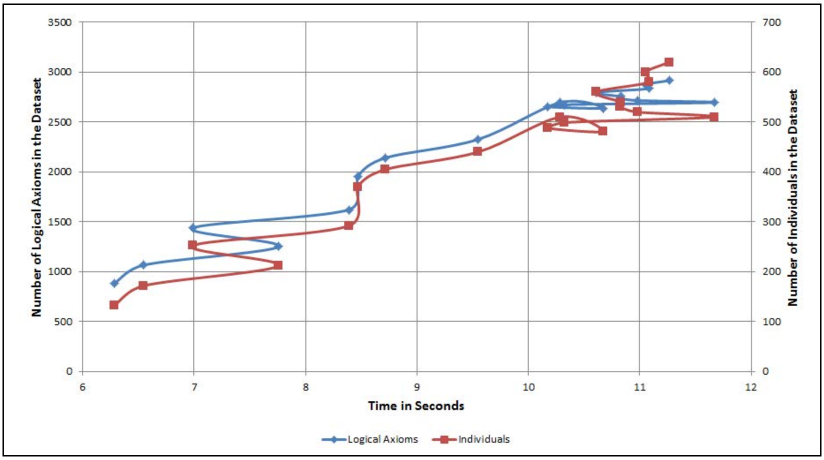


Fig. 6. System Performance based on Number of Axioms and Individuals

From these results, it is clear that for best performance in this framework, users and LAC service provider should diligently design their policies to not only get the best functionality from the system, but also to ensure that least amount of reasoning is delegated to the user's mobile device. This can be done by paying great attention to modeling the data and using policies that ensure least number of axioms are required to be reasoned over on the client.

Details of a N3 based rule editor, the underlying technology and its evaluation for performance and user acceptance are presented in [8]. An adaptation of the same policy tool was used to create many of our mobile based policies.

6 Conclusions and Future Work

In this paper we showed how semantic policies could be used in creating a highly customizable context management system which allows the user to decide what data he/she wants to share with his/her telco service provider and what to hide. The architecture of the system was elaborated upon and we also presented the details of message exchange between the user device, the telco server and the campaign provider. We have successfully demonstrated the utility of semantic rules to create policies for such a system not just by the end user but also the telco provider and the campaign provider. We further discussed about our methodologies of evaluating this system based on various metrics and presented the results. In the results, we highlighted the importance of an efficient modeling of data and policies for such systems to generate optimum performance.

Currently, we are working on evaluating the system on a bigger scale with real user mobility data to further test its scalability to be deployed in practical systems. We are also working on making the policy creation interface more user friendly so even non-expert users can create custom rules. In future, there is also scope to develop a semi-automatic tool to assist campaign providers in annotating their offers with rich snippets.

Acknowledgments. The Competence Center FTW Forschungszentrum Telekommunikation Wien GmbH is funded within the program COMET - Competence Centers for Excellent Technologies by BMVIT, BMWA, and the City of Vienna. The COMET program is managed by the FFG. We would like to thank the APSINT project team for their contributions to this work.

References

1. Bikakis, A., Patkos, T., Antoniou, G., Plexousakis, G.: A Survey of Semantics-based Approaches for Context Reasoning in Ambient Intelligence. In: Mühlhäuser, M., Ferscha, A., Aitenbichler, E. (eds.) AmI 2007 Workshops. CCIS, vol. 11, pp. 14–23. Springer, Heidelberg (2008)
2. Chen, H., Finin, T., Joshi, A.: Semantic Web in a Pervasive Context-Aware Architecture. In: Artificial Intelligence in Mobile System 2003, pp. 33–40 (2003)

3. Forstadius, J., Lassila, O., Seppanen, T.: RDF-based Model for Context-aware Reasoning in Rich Service Environment. In: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW 2005), pp. 15–19 (2005)
4. Patkos, T., Bikakis, A., Antoniou, G., Papadopouli, M., Plexousakis, D.: A Semantics-based Framework for Context-Aware Services: Lessons Learned and Challenges. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 839–848. Springer, Heidelberg (2007)
5. Kumar, V., Fensel, A., Lazendic, G., Lehner, U.: Semantic Policy-Based Data Management for Energy Efficient Smart Buildings. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM-WS 2012. LNCS, vol. 7567, pp. 272–281. Springer, Heidelberg (2012)
6. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications 28(1), 1–18 (2005)
7. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 329–346. Springer, Heidelberg (2008)
8. Zhdanova, A.V., Zeiß, J., Dantcheva, A., Gabner, R., Bessler, S.: A Semantic Policy Management Environment for End-Users and Its Empirical Study. In: Pellegrini, T., Auer, S., Tochtermann, K., Schaffert, S. (eds.) Networked Knowledge - Networked Media. SCI, vol. 221, pp. 249–267. Springer, Heidelberg (2009)
9. Kumar, V.: A Semantic Policy Sharing and Adaptation Infrastructure for Pervasive Communities. In: ACM Proceedings of the 21st International Conference Companion on World Wide Web 2012 (WWW 2012), Lyon, France, pp. 155–160 (2012)
10. Semantic Web Rule Language (SWRL), `http://www.w3.org/Submission/SWRL/`
11. The HermiT Reasoner, `http://hermit-reasoner.com/`
12. RDFa Primer, `http://www.w3.org/TR/2012/NOTE-rdfa-primer-20120607/`
13. Microdata HTML, `http://en.wikipedia.org/wiki/Microdata_(HTML)`
14. GoodRelations Snippet Generator, `http://www.ebusiness-unibw.org/tools/grsnippetgen/`
15. The Protege framework, `http://protege.stanford.edu`
16. The Advanced Prosumer Service Integration Intelligence (APSINT) Project, `http://www.ftw.at/research-innovation/projects/apsint`

TrustPos Model: Trusting in Mobile Users' Location

Javier Martín de Valmaseda, Georgia Ionescu, and Michel Deriaz

University of Geneva, Institute of Services Science, 1227 Carouge, Switzerland
{Javier.MartinDeValmaseda,Georgia.Ionescu,
Michel.Deriaz}@unige.ch

Abstract. While social games based on geo-location are gaining popularity, determining the authenticity of the players' geo-position becomes a challenge, since there are ways to counterfeit it, quite accessible to everyone. We propose a solution based on global spatial and temporal observation of the players' interactions. In this paper we present TrustPos, a trust engine model that associates a trustworthiness factor to each player based on the context of the interactions with both the game and other players. The novelty of TrustPos is the fact that our model is based on an internal network of players linked through their interactions, as opposed to previous approaches that are strongly specialized to concrete domains as peer-to-peer networks and social recommenders, not adaptable to location trust concerns.

Keywords: Location, mobile, trust, social games.

1 Introduction

Nowadays, the interactions in social networks are increasingly linked to the user's location. Different games, applications and communities based on location capabilities are growing at a rapid pace. Let us consider as an example Foursquare [1], a well-known location-based social network that allows users to post their location at a venue ("check-in") and connect with friends. By January 2013, Foursquare acquired over 30 million users worldwide, with over 3 billion check-ins, and millions more every day. This shows how much success the location-based capabilities can bring to social applications. Games are becoming more and more social, and location features can only contribute to their success.

In order to bring innovations and to connect players to the real world, games are using location-based features. For instance Ingress [2], the new location-based massively multiplayer game developed by Google's Niantic Labs. In this game, location check-ins and interactions are necessary to unlock clues about what is going on, gather objects, work together and much more.

As more games integrate the geo-location capabilities of mobile devices, game providers need to trust the user's position, they need to be sure that users are really where they claim to be, in order to avoid cheaters and to keep the game interesting and competitive.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 79–89, 2013.

However, cheating over the current location is easy since it is the mobile device that gets the geo-position and sends it to a server. When searching for Ingress on the web, it is easy to find tutorials [3] about how to fake your location in the game.

Encryption is not a solution either, as it is easy to find where the position is obtained (GPS for instance), even in obfuscated code, and then a malevolent user can simply change the data before the encryption process. Nevertheless, a global spatial and temporal observation of the full system would allow us to find cheaters. For instance, a *user A*, interacting with a *user B* in London should not be able to interact with a *user C* in New York one hour later. At least one is cheating about his position. By looking further into the history of these three users, the chance of finding out who is cheating becomes higher.

This paper presents a system called TrustPos, which is a new trust engine model based on the idea of spatial messaging (defined by [4-5]). This idea is combined with a trust model where users, transparently, are able to create trust links between one another and with the system.

Position information given by a user is also rated, for instance it has more weight if it is confirmed by a trustworthy user or if the interaction has been done with a trusty source, like a NFC reader in a shop. The trust engine is able to determinate how trustworthy the user is and gives a rate to the interaction after evaluating it.

In section 2 we start by presenting different approaches made before in the location and trust field, and then we explain our chosen path. Section 3 presents an overview of our solution. In Section 4 the trust model and the trust engine is explained. Finally we give our conclusions and future work in section 5.

2 Related Work

There are several approaches in the field of certifying and proving users' location in mobile applications. For instance, Lenders et al. [6] propose a secure geo-tagging service. Their purpose is to be able to trust in the content produced by a mobile user. Towards this goal the content is tagged with a Data Location Time certificate. Later on, a user consumer can verify the original location and time of the information. The content can be verified by checking the signature of the certificate using the public key of the location/certificate authority. The problem with this method is that they trust, or assume that the way of obtaining the location is trustworthy. They propose several methods for ensuring the way of obtaining the location, however these methods are not enough. Cheating over the current location before the certification process is easy since it is the mobile device that gets its position and sends it to the server. Our engine takes into account this factor when analysing a user's position and should be able to detect if the user is potentially trying to cheat according to previous interactions.

Saroiu and Wolman [7] have an interesting approach for trusting the location, using what they called location proofs. A location proof is a piece of data that certifies a receiver to a geographical location. Basically they rely on Wi-Fi access points or cell towers to generate these location proofs. The geographical location of the access point is embedded in each location proof, which is then transmitted to the mobile devices. Following this approach, Luo and Hengartner [8] propose what seems to be a better solution using a similar technique. Besides their own location proof architecture based on access points, they take care of privacy protection of the users and they provide a mechanism in order to deal with the cheating users.

Unfortunately, relying on external infrastructure is unaffordable in the world wide social games that we are targeting. This technique is limited to areas with this infrastructure already deployed.

There are several proposals to trusting in the field of sensor information. For instance [9-11] which present similar solutions based on a hardware device implanted in the mobile phone. These solutions use a dedicated hardware (called Trusted Platform Module hardware, TPM [12]). The main difference is that in Gilbert et al. [9] and Zhidong et al. [10] the target is a trustworthy mobile sensing platform integrated in the mobile phone while in Dua et al. [11] rely on signing the raw readings from the sensors. However these solutions are beyond the scope of our target application, a social game that will be played in several different devices that will not incorporate this specific hardware.

Several approaches have been made in the field of trust computations. These solutions are mostly oriented towards trust relations and calculations in social networks, recommender systems and peer-to-peer networks. From our point of view there are two ways of handling trust information: centralised and decentralized. The eBay site [13] is a good representation of a centralized system. In this case, each user has a global reputation that is calculated by the system, using the different rates given by other users. After each transaction, the buyer and the seller rate each other and the global reputation is automatically updated by the system.

On the other hand, the decentralised trust systems are a bit more complex. For instance, in a peer-to-peer network, peers rate each other and the reputation, of a peer is the sum of all the other peers' rate. As there is no global system or global values for each user, it is necessary to ask all the users about the reputation of a given one. These networks algorithms are widely used for recommendations in social networks, like movie recommenders.

Different surveys on Trust Computations [17-16] show that the majority of current algorithms have been proposed for special computing environment such as wireless networks, peer-to-peer systems and social recommenders. It will be complicated to use one of these approaches in a rapidly changing environment due to its complexity and specialisation.

A case of highly decentralized ad hoc networks is the Wireless Sensor Networks (WSNs). Over the past decade, several techniques like S. Capkun et al [14] and Joengmin Hwang et al [15] have been proposed for solving the positioning problem in wireless networks. J. G. Alfaro et al [16] proposes three different algorithms that enable WSNs to determine their location in presence of neighbour sensors that lie about their position. Additionally, these techniques aim at isolating the set of liars. All three of them are based on the radio signals submitted to neighbours, which allow them to compute the distance between different sets of nodes. As interesting and efficient this approach was proven to be, it cannot be used for our purposes. The most important reason is the fact that in our case, the devices do not interact with one another. Their positions are shared through a server that holds the database with all the interactions.

Another problem when thinking about applying one of these algorithms to our case is that in most of the trust models there is no awareness of the context or the time.

However for our target, time and context are important, the users interact along the time, frequently, and the context is different. For instance, having a player conquering territories physically at the same time at Tokyo and Paris is not possible. The circumstances of the event are important to our trust model; therefore, we believe that a trust model similar to the idea of FoxyTag [4] could be relevant for our objective.

FoxyTag is a speed camera warning system based on special geo-located tags (speed cameras posted by users while driving) and a trust engine to self-maintain the tags database. FoxyTag allows drivers to easily signal a speed camera or to signal that a former one has been removed. Other users driving in areas with tags receive the alert about the speed camera. The main advantage of FoxyTag is that it does not require human checks to decide the trustworthiness of the posted tags because the system uses a computational trust engine to automatically make this decision.

Our idea is to develop a trust engine based on this model in which each user action is recorded and observed. With each action trust links are created between users. We can have a global view of one specific player asking the community about the trustworthiness of this specific user. Similar to human community, when we want to know if someone is trusty, we ask other people whom we trust, or people who already have interact with this user.

3 Solution Overview

As shown in the introduction, location features are going to be part of the games, at least a distinctive feature. Location capabilities bring opportunities to innovate and to gather players in the real world. For instance, the games are going to allow the players to create their own location-based territories, and to fight against others in the same location. For this purpose, we need to trust the position given by the users to keep the game interesting and competitive.

Our proposed solution is to develop a trust engine in order to manage the trustworthiness of the different users' actions when posting the position in the game. Based on these interactions, we make a global spatial and temporal observation of the users.

Therefore, users can create transparent trust links with the system and other users and rate the different interactions. With these different values, it is possible to achieve a global view of the different users and interactions. Furthermore it is possible to make the proper decisions upon the trustworthiness of each user.

For better understanding of our purpose, we further describe the target game and two main usage scenarios.

3.1 Target Application

The main target game that we are pursuing is a location-based game. In this game, the players have the possibility of play and interact using the location features of theirs mobile phones. For instance, players will create their own location-based territories, and they will fight against others in the same location. For this purpose, and for keeping the game interesting and competitive, we need to trust the position given by the users.

When two users are playing together at the same location, they are mutually agreeing of being there. Therefore if the trust engine detects that the user is cheating, the trust that the system and the other user has in this user will be decreased.

We are working closely with EverdreamSoft a game company that produces a successful worldwide card game – Moonga [20]. Moonga is a multi player mobile game running on both Android and iOS. Based on 5 cards the players can build creative strategies to dominate their opponents' cards. In the next main release EverdreamSoft is planning to integrate our engine in order to trust the positions given by the players. Their goal is to make the game more social by merging it with real world interactions. Players will meet to play in real world for conquering physical territories, for exchanging cards and for card fights.

3.2 Main Usage Scenarios

In order to keep things simple, we have defined two basic scenarios in our target game. The most basic one is when one user is trying to create a territory in the game using their geo-position. From the point of view of our trust engine, this is seen as a user-system interaction.

The second possible scenario is when one user plays against another user in the same physical location. This is seen as an interaction between two users. Bellow, we give a more detailed explanation for each scenario.

Scenario1. Alice is a user who wants to create a new territory in London. In the game options, she selects the corresponding create territory option at this position. She is now transparently interacting with the trust engine. The client application (mobile phone) sends a *post message* to the system (server) with the pertinent information about this interaction.

With the information contained in the message, the history of actions of Alice (position, time and other parameters) is updated on the server.

The server returns the trust value for Alice or whether it is possible to be where she claims to be (it is not possible to be in London and within 30 min in Paris). These calculations/observations can be made by taking into account the current context, the previous ones and the trust values that other users and the system have for this user.

Scenario2. Alice wants to play/fight with another user in the same location/area; after selecting the appropriate option in the game, Alice will get a list of the nearest users to interact with. When Alice selects another user within the list (Bob), and Bob accepts the request, both users send a *review request* to the server. The *review request* means that both users' history is updated with the context of this last interaction (time, location, opponent, etc.).

Due to this interaction each user is transparently rating the other user and updating their local trust value (their opinion) for this user. For instance, if Alice is fighting with Bob in the same physical location, both users should have the same position, and the positions should be physically possible compared to their last one. If every parameter of the context is as expected, each user will transparently update the trust on their opponent, if something is wrong, the trust on the opponent will be decreased.

With this review process we keep track of the last interactions for each user and a local trust value for all the users.

It is important to notice that the whole process is completely transparent to the player. Users cannot see the different trust value or the rates given by the trust engine. However, when accepting to play against other user posting the location, the users are claiming to be at the same place. Typically the community of trustworthy users will not play against not trustworthy users. Therefore, we expect to have an isolated group of cheaters and a big community of fair players, as we can see nowadays in our target game Moonga.

3.3 System Definitions

User Interaction. When one or two users interact with the game, the context of the interaction (position and time) is sent to our trust engine. The trust engine calculates the trust value of the user, rates the interaction and updates the tag (the user information and history). Figure 1 shows two users interacting with our system.

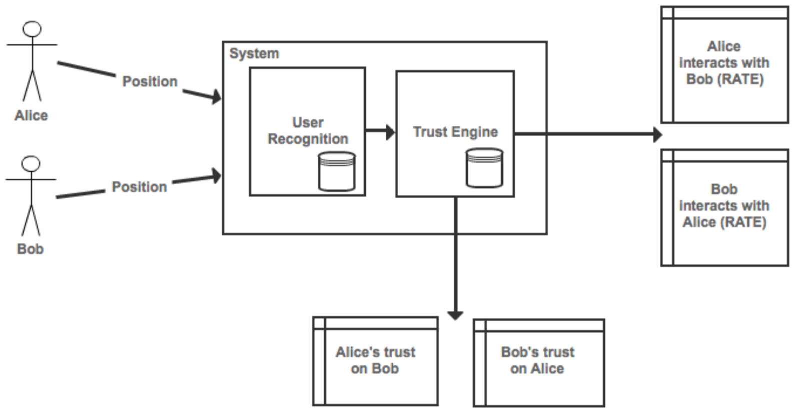


Fig. 1. Users interacting with the system

User. We keep track of the context (time and location) of the last interactions for each user to be able to evaluate his trustworthiness.

A user in the system has his own trust table. Each user keeps a local trust value (opinion) based on former interactions with other users. We consider the system as a special user who also has its own local trust table for all the interactions that the users do. This mechanism is similar to a human community, meaning that each individual keeps an opinion of other individuals based on the previous interactions.

Trust Value. In our model we use two important parameters:

- Local trust: Direct trust that one user has in another user, based only on former direct interactions between them.
- Global trust: The average of the local trust value of all the users in the system when asking for a given user.

MinTrustValue. The minimum trust a user can have. It should be big enough, bigger than the *maxTrustValue* (in absolute). It is difficult to become trusted after a few positive actions but easier to become untrusted after the same amount of negative actions. Initially the *minTrustValue* has been set to -70.

MaxTrustValue. The maximum trust a user can have. It should not be too high in order to avoid trusty users with a high trust value become malevolent. Initially its value is set to 5.

Rate. The rate of the interaction is a value between 0 and *maxRate* given by the trust engine to each user action. This value is helpful to evaluate the interaction (*user-system* or *userA-userB*) giving a rate or trust value to it. In this value the trust engine takes into account the different parameters involved (interactions made from a trusty source, same position as other user, etc). For instance, the rate is bigger if both users have the same position or if the previous interactions were positive, etc.

History. The history keeps track of the last *N* interactions with the system or other users sorted in reverse chronological order (recent events are in the top, the old ones are at the bottom). This field stores information related to the interaction, the different users, the positions and the rate given by the trust engine among others.

4 Trust Engine Model

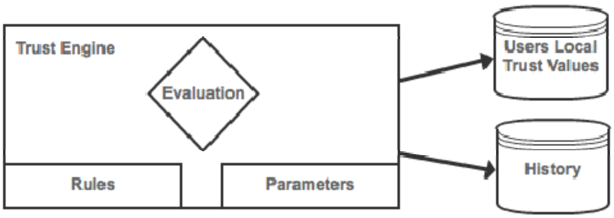


Fig. 2. Trust Engine Architecture

4.1 Trust Engine Architecture

The architecture of the trust engine's model has been kept as simple as possible. As shown in Figure 2 the Trust Engine consists in a set of parameters and rules and two databases with the entire local trust values and the history of the different users.

The rules and parameters are based on the different context and actions. For instance, if the position is possible when comparing to the history of positions, the trust calculation will be positive. The parameters reflect the values necessaries to make the decisions, such as the limit when a user becomes untrustworthy.

The evaluation component is in charge of handling the interaction, using the rules, parameters and trust model to make the proper decision and update the data and out-puts.

4.2 Trust Model

Our model seeks to be close to the human concept of trust. In human communities, when an individual wants to know if another individual is trustworthy, the usual way of proceeding is to ask his friends or other individuals that have already interacted with this particular individual. We adopted a similar strategy; for each user, we store a trust value of other users, based on former interactions. We call this value *local trust value*, which in essence is the opinion that a user has on another user based on previous interactions.

So, when we want to know about the trustworthiness of a user, we only need to ask all the other users with whom he has interacted, including the system. We call this value *global trust value*.

We want a model similar to the human way of thinking in which recent events are in mind. It is difficult to become truly trustworthy. It is necessary to have several positive consecutive actions to achieve it. However once someone is trustworthy after a few disappointments it is easy to be untrustworthy.

In our model, trust increases linearly towards a limit and decreases exponentially. This means that several consecutive positive actions are needed in order to increase the trust value and become trustworthy. On the other hand, as the decrement is exponential, little mistakes at the beginning are forgivable, but if the consequences of the actions are negative, the value decreases exponentially and it is very difficult to become trustworthy again. In our point of view this behaviours, reflect the human way of thinking in managing trust. Perhaps misunderstanding how the game works or small cheats at the beginning are forgivable but if the behaviour persists, the trust on this individual will decrease quickly and he will never be trustworthy again.

As an example, consider a situation where to co-workers go to buy coffee every day. It is always the same person who pays but he is reimbursed by his colleague when they return to the office.

The one who pays, trusts that his partner will pay him the coffee later. If one day his partner does not pay, as he trusts his partner, he will think that is a mistake or misunderstanding. But if the behaviour of the partner continues like that, at the end, he will stop paying his colleague's coffee because he will not trust him anymore.

It is important to take into account the fact that we need to fix a maximum and a minimum trust value that a user can achieve. Typically, the maximum value should not be too high in order to avoid that a user which has been acting properly for a long time, suddenly becomes malevolent and tries to subvert the system. However for the minimum value, we should set a big negative value so a user that has been cheating during all the interactions should not be easily trustworthy again.

As we previously commented, the trust engine rates each interaction, taking into account the context and the different parameters of the interaction. For instance it has more weight if the interaction is with a trustworthy user or if the interaction has been done with a trusty source, like a NFC reader in a shop or if the last interactions were also positive.

4.3 Trust Engine Model

In Figure 3 we can see the basic behaviour of the trust engine. The process starts with an interaction, that could be one user alone posting his position or two users who want to play against each other. All the important contextual information is transmitted to the trust engine; after evaluating the different inputs and the previous history the trust engine updates the corresponding trust values and rates the interaction following specific rules and decisions of this model.

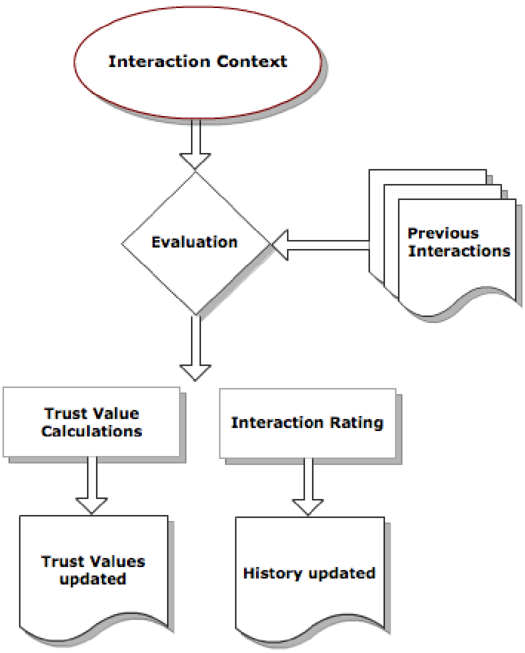


Fig. 3. Trust Engine Model

5 Conclusions and Future Work

Mobile technologies and smartphones with location capabilities are widely popular nowadays as well as applications and games with location-based features. For social location-based applications, like games with physical features based on users' mobile location it is important to trust the position sent by the users. All the similar solutions that we have found are basically focused on using external infrastructures, which is unconceivable for a social game spread all over the world.

In this paper, we have presented our approach to solve the problem of trusting the position given by a user's mobile phone in a game environment. We have explained our model of trust engine, which is able to generate and manage transparent trust links between the users in order to determinate their trustworthiness. This trust model uses the user history, and interactions to rate each interaction in the game and calculate a local trust value for each user and a rate for each interaction. The trust engine, combined with the appropriate security measures when posting the position in the game, is a powerful tool for avoiding cheaters.

TrusPos is a very good first filter, which will be combined with the appropriate security methods to obtain the position. Additionally, a management tool will be provided, so a human operator can see the last suspicious interactions, trust updates, and make the decision whether to ban or put in quarantine suspicious users and actions.

After a review of the current related approaches, we can conclude that there is not yet a popular solution to this problem and that our model could be an interesting solution. We have presented an overview of our model, our trust model, the behaviour and architecture.

Currently, this model is in deployment and testing phase. We have started the development of the trust engine and we are constantly working on and improving it. In the future we will finish all the functionalities presented. Simulations will be made to test the proper behaviour of our trust engine. A simulator is under development in order to help us test different situations in a location-based game.

Acknowledgments. The work of this paper was supported by the Commission for Technology and Innovation - CTI under contract no 14249.1.

References

1. Foursquare, `https://foursquare.com/about` (accessed on May 2013)
2. Ingress. The game. NianticLabs@Google, `http://www.ingress.com` (accessed on May 2013)
3. Ingress mock your location, `http://decodeingress.me/2012/11/23/ingress-chea-mock-your-location` (accessed on May 2013)
4. Deriaz, M., Seigneur, J.-M.: Trust and Security in Spatial Messaging: FoxyTag, the Speed Camera Case Study. In: Proceedings of the 3rd International Conference on Privacy, Security and Trust. ACM (2006)
5. Deriaz, M.: Trust without Truth. In: Etalle, S., Marsch, S. (eds.) Trust Management. IFIP, vol. 238, pp. 31–45. Springer, Boston (2007)

6. Lenders, V., Koukoumidis, E., Zhang, P., Martonosi, M.: Location-based Trust for Mobile User-generated Content: Applications, Challenges and Implementations. In: Proc. HotMobile 2008 (2008)
7. Saroiu, S., Wolman, A.: Enabling new mobile applications with location proofs. In: Proc. HotMobile 2009 (2009)
8. Luo, W., Hengartner, U.: VeriPlace: A Privacy-Aware Location Proof Architecture. In: Proc. of ACM SIGSPATIAL GIS (2010)
9. Gilbert, P., Cox, L.P., Jung, J., Wetherall, D.: Toward Trustworthy Mobile Sensing. In: Proc. HotMobile 2010 (2010)
10. Shen, Z., Wu, X., Zhan, J.: Trust management for mobile agent system based on trusted computing platforms. In: 2010 IEEE International Wireless Communications, Networking and Information Security (WCNIS) (2010)
11. Dua, A., Bulusu, N., Feng, W.-C., Hu, W.: Towards trustworthy participatory sensing. In: HotSec 2009: Proc. of the Usenix Workshop on Hot Topics in Security (2009)
12. Trusted computing group - trusted platform module specifications, `http://www.trustedcomputinggroup.org/developers/trused_platform_module/specifications` (accessed on May 2013)
13. eBay, `http://www.ebay.com` (accessed on May 2013)
14. Capkun, S., Hubaux, J.-P.: Secure positioning in wireless networks. IEEE Journal on Selected Areas in Communications 24(2), 221–232 (2006)
15. Hwang, J., He, T., Kim, Y.: Detecting Phantom Nodes in Wireless Sensor Networks. In: 26th IEEE International Conference on Computer Communications, INFOCOM 2007, May 6-12, pp. 2391–2395. IEEE (2007)
16. Alfaro, J.G., Barbeau, M., Kranakis, E.: Secure Localization of Nodes in Wireless Sensor Networks with Limited Number of Truth Tellers. In: Seventh Annual Communication Networks and Services Research Conference, CNSR 2009, May 11-13, pp. 86–93 (2009)
17. Govindan, K., Mohapatra, P.: Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey. IEEE Communications Surveys & Tutorials 14(2), 279–298 (2012)
18. Viriyasitavat, W., Martin, A.: A Survey of Trust in Workflows and Relevant Contexts. IEEE Communications Surveys & Tutorials 14(3), 911–940 (2012)
19. Zhang, P., Durresi, A., Barolli, L.: Survey of Trust Management on Various Networks. In: 2011 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), June 30-July 2, pp. 219–226 (2011)
20. Moonga the Game. EverdreamSoft, `http://www.moonga.com` (accessed on May 2013)

Agile Software Development Processes for Mobile Systems: Accomplishment, Evidence and Evolution

Luis Corral, Alberto Sillitti, and Giancarlo Succi

Free University of Bozen-Bolzano
Piazza Domenicani 3, 39100 Bolzano-Bozen, Italy
luis.corral@stud-inf.unibz.it,
{alberto.sillitti,giancarlo.succi}@unibz.it

Abstract. Mobile software applications have to cope with a particular execution environment that includes limited resources, high autonomy requirements, market regulations, and many other constraints. To provide a software development process that responds to these challenges, several methodologies proposed the adoption of Agile practices; however, it is not clear how a software development process would help to solve all the issues present in the mobile domain. Moreover, the rapid evolution of the mobile environment questions several of the premises upon which the proposed methodologies were designed. In this paper, we present a review on Agile software development processes for mobile applications and their implementations, with the objective of knowing the contribution of Agile methods to address the needs of the mobile software in a real production environment. In addition, we aim to put up to date the discussion about what are the best practices that facilitate the creation of high quality software products in the current mobile domain.

Keywords: Agile, Development, Mobile, Process, Quality.

1 Introduction

Smart mobile devices like cellphones and tablets are a key target for software products and services, benefiting from the high impact of ubiquitous computing and the growing capabilities of this kind of terminals. The quantity of software applications for smart devices grows very quickly, along with the features and computing power offered by the mobile equipment. Mobile-specific software applications, commonly known as "apps" bear the big challenge of performing satisfactorily in a heterogeneous and resource-limited environment that demands high availability, efficient performance and short response time, while delivering value to the end user. In addition, apps should be developed quickly and should keep a low price to succeed in a market that comprises millions of users and hundreds of thousands of products. All these factors build up an environment that is competitive, complex and fault prone, posing the question of how to develop software products able to succeed in the mobile domain [1].

In an attempt to answer this question, scientific literature on Software Engineering introduces different frameworks for conducting mobile software projects. Even

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 90–106, 2013.

though diverse approaches are proposed, a majority of these research works show a convergent approach based on the Agile home ground themes. Nonetheless, they do not explain how mobile-specific conditions may influence the selection of a software development process, or why a software development process would help to solve the issues present in the mobile domain. Additionally, there is a lack of evidence about how the proposed methodologies work in a real setting, since the follow up literature tends to be limited and does not provide elements to evaluate their utilization. Finally, the rapid evolution and the current status of the mobile environment challenges several of the premises upon which the proposed methodologies were designed.

In this paper, we present a review on how different Agile-based frameworks (abbreviated: *Agile*) claimed to suit the needs of the mobile environment, establishing as research question: "What is the contribution of Agile methods to address the needs of the mobile software product in a real setting?" To sketch a strategy to solve it, we introduce discussion in four areas: (a) Suitability of Agile to fit the needs of the mobile business environment; (b) Suitability of Agile to fit the needs of the mobile operational environment; (c) Adoption of Agile frameworks and evidence about their use in mobile projects; and (d) The rise of new conditions that challenge some of the premises upon which the proposed frameworks were designed.

The rest of the paper is organized as follows: Section 2 revisits the need of having ad-hoc development processes for the mobile product, Section 3 presents a summary of published Agile methodologies for conducting mobile software projects, Section 4 performs a high-level comparative analysis among the presented methodologies; Section 5 introduces discussion about their real adoption and contribution, as well as the evolution of the mobile domain; Section 6 outlines further work on this research track; to close, Section 7 provides a summary and draws conclusions.

2 Mobile Software Development

Mobile software development should comply with clear goals and practices in order to be successful, however, this kind of software bears several limitations not present in desktop computing that make the mobile ecosystem a particular environment. For instance, wireless communication problems (availability, variability, intermittence), mobility issues (autonomy, localization), the variety of platforms and technologies, the limited capabilities of terminal devices (low power supplies, small-sized user interfaces), and strict time-to-market requirements [2].

The universe of constraints that exist on mobile environments can be classified in two types: evolving and inherent [3]. Evolving constraints include current limitations that will be solved in the future by the evolution in technology, resources, bandwidth, coverage, etc., (e.g., slow processors or intermittent networks). Inherent constraints are those intrinsic to mobile platforms, since they are part of the operational condition of the system and will not be solved in the near future (e.g., a small screen or a limited power source). These constraints should be taken into account by developers to determine the practices to relieve them as means of non-functional requirements. In addition, mobile software products have a particular business model [4] that foresees high competition, short time to market, and large distribution app stores.

Those factors require a link with the development practices that help to produce software able to compete in such an environment [5, 6]. For instance, the fast-paced mobile market sets the need of having lightweight processes that facilitate the change and the adoption of new technology or emerging trends. An effective development strategy should be strong enough to consider the quality drivers of the mobile ecosystem, the expectations of the end user, and the conditions set by application market; at the same time, it should be flexible to adapt to the advancements of the enabling technologies, and to cope with the market's competitiveness and the evolution.

To provide an answer to the discussed challenges it has been proposed that Agile practices are the framework of choice for the development of software products for mobile devices. In 2003 it was introduced the discussion on the suitability of Agile for the fulfillment of the objectives of the mobile software development [7]. Later, it was shown a thorough mapping between Agile home ground themes with several development traits observed in mobile software [8]. This mapping permitted to outline why Agile is a competent solution for implementing development processes in the mobile domain, based on characteristics like collaboration in small teams, reduction of development times, management of ongoing changes of requirements, coping with the variety of target platforms, and the assumption a small-sized, non-critical end product.

3 Agile Development Processes for Mobile Applications

Our research question pursues to know the contribution of Agile to address the needs of the mobile software in a real environment. To have a better understanding on how Agile practices can be implemented in a mobile project, we surveyed three major digital libraries (IEEE Xplore, ACM DL and ScienceDirect) searching for research articles that present a comprehensive mobile software development framework based upon Agile principles. To make the selection of the mobile development frameworks, we established as criteria: 1) frameworks should be specific and mainly applied for mobile applications; 2) they should focus on the use of Agile methods; 3) frameworks should have been put in practice preferably in at least one study case, and 4) frameworks and case studies must have been published in an international journal or conference proceedings. As result, we found on five Agile approaches: Mobile-D [9], MASAM [10], Hybrid Methodology [11], Scrum [12], and Scrum Lean Six Sigma [13], sorted by publication date (Figure 1).

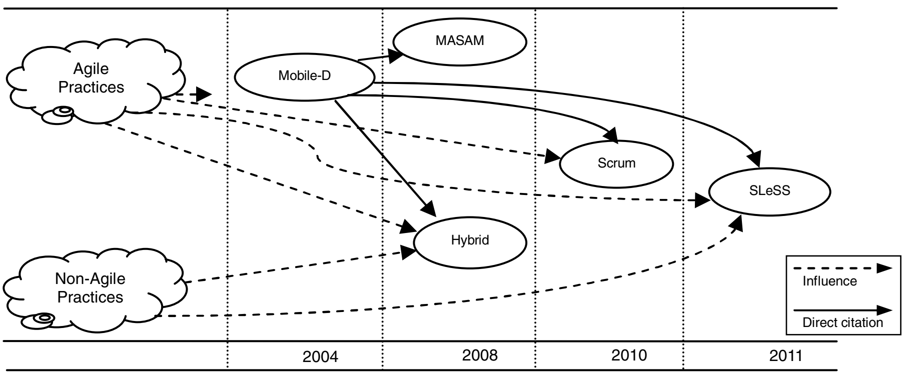


Fig. 1. Evolution of Agile Development Methodologies for Mobile Software

Once we identify the body of knowledge, our next question concerns on analyzing in a detailed way how each methodology undertakes each phase of a generic software development process to identify development needs and opportunities still open for improvement. Then, to determine their contribution in a production environment, we looked for instances (research papers, technical reports) that cite these methodologies to illustrate how to utilize them to develop a real-world mobile product. Citation count was calculated from the number reported in the digital libraries when available, and it was complemented from Google Scholar removing duplicated instances.

3.1 Mobile-D

Mobile-D was the first attempt to incorporate Agile for the development of mobile applications. Mobile-D was introduced in 2004 by Abrahamsson et al. [9] as a development methodology inspired on Extreme Programming, Crystal Methodologies and Rational Unified Process (RUP). It is recommended to be used by a small, co-located team, working in a short development cycle. It is structured in five phases (Figure 2), sequentially arranged following a generic software development process. Each phase implies a Sprint, covering from stakeholder identification through system test and delivery. The execution of a single phase involves typically around three days for planning, producing, wrapping and delivering the associated work products. In spite of its sequential organization, Mobile-D encourages iterations, after which a functional product is created. Actual Agile activities within the methodology include: Test-Driven Development, Continuous Integration, Pair Programming, etc.

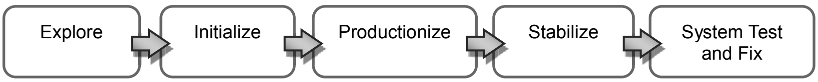


Fig. 2. Phases of Mobile-D Software Development Process (Adapted from [9])

Mobile-D has been put in practice in several projects at the ENERGI (Industry-Driven Experimental Software Engineering Initiative) laboratory of the VTT, the technical research center of Finland, these project involved mobile phone extensions of database systems. Furthermore, it was exercised in a case study on a major software development project on software security for a large customer (F-Secure) in a project to develop a mobile security application. In general, Mobile-D is the most influential methodology in the field, being cited by 17 articles including case studies, though not all of them referring to mobile development. Other 16 examples of use in real projects are showcased in the Mobile-D website.

3.2 MASAM

MASAM (Mobile Application Software development based on Agile Methodology) was proposed by Jeong, Lee and Shin [10] and it is based on Extreme Programming, Agile Unified Process, RUP and The Software and Systems Process Engineering Meta-model. It provides a series of principles from which different development processes can be defined according to the context of an Agile software development company.

The structure and detailed implementation of MASAM show a strong tie with Mobile-D, and only introduces slight variations, for example a project management and follow up tool harnessed on the Eclipse Process Framework. MASAM follows a software life cycle based on the Agile approach underlining the importance of interaction among participants, communication with the customer, Extreme development practices and continuous deliveries. After these principles, MASAM proposes a simple life cycle conformed by 4 phases (Figure 3).

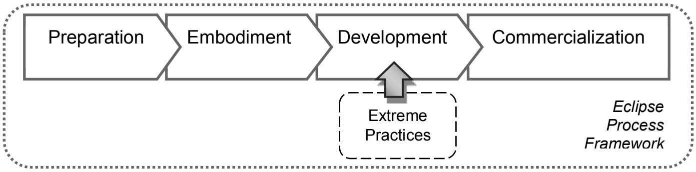


Fig. 3. Stages of MASAM (Adapted from [10])

First, a Preparation Phase defines a summary and a first notion of the product, and assigns roles and responsibilities. The Embodiment Phase, focused on understanding user's needs, defines the architecture of the software product. The next phase, Product Developing Phase, benefits from traditional Agile principles to furnish an iterative development sequence supported by Extreme Programming. The implementation of the software product is carried out through Test-Driven Development, Pair Programming, etc. with iterative testing activities. Finally, a Commercialization Phase concentrates on product launching and product selling activities.

MASAM claims to be supportive for small companies focused on the development of mobile applications, nonetheless, authors do not present a case study of an actual implementation of this methodology in a real-world project to appreciate its results. It is cited by 3 papers, none of which is an experience report of its utilization.

3.3 Hybrid Methodology Design Process

Rahimian and Ramsin [11] promoted the adoption of Agile and plan-based methodologies as a framework for the development of mobile software. They identified specific requirements and activities based on characteristics expected on tasks and intermediate products through the development process. To structure their methodology, it is proposed to baseline a generic software development lifecycle and customize it with a merge between Agile and principles of Methodology Engineering and New Product Development. The outcome is a "Hybrid Methodology Design Process" organized in steps that cover from the generation of the idea until the release of the product (Figure 4).

The principles present in this methodology are: Agility, Market Consciousness, Software Product Line Support, Architecture-Based Development, Support for Reusability, Review and Learning Sessions, among others. It defines a of a top-down

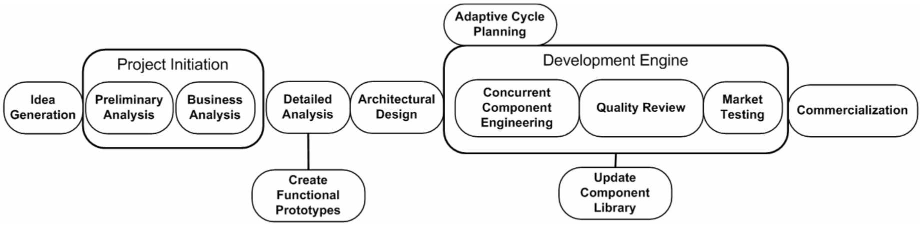


Fig. 4. Phases of the Hybrid Methodology Design [11]

iterative-incremental process that considers prioritizing the requirements, building an "Iterative Design Engine" (that designs, models integrates and review components) and finish with market testing and commercialization. It is remarkable the inclusion of market-awareness principles, aimed to conduct an efficient commercialization of the resulting product, addressing a well identified need on mobile software development.

Unfortunately, up to the date of the underlying work, published material on Hybrid Methodology Design and its 9 citations do not include an experimental setup that tests this methodology in the development of a real-world, mobile software product.

3.4 Scrum

Scharff and Verma [12] published a study that covers the use of Scrum for the development of mobile applications in a scholar setting. Scrum is an iterative and incremental framework commonly used in combination with other Agile practices. It uses iterations of fixed duration (typically one to four weeks) called Sprints. At the beginning of each Sprint, during the Sprint planning, the team commits to complete a certain number of tasks established from the "Product Backlog" and documents them in a "Sprint Backlog". After this, the Scrum team decides how much work they will commit to complete in the next Sprint.

Through the iteration, daily meetings are carried out at the beginning of the day to keep track of the progress and setting the goals for the working day. By the end of the Sprint, a functional product is delivered, and the pending features (i.e., the new Product Backlog), are processed in a next iteration (Figure 5). For project tracking, it is used a "Burn-down Chart", that is a relationship between the pending work and the time. With this, it is possible to relate the Product Backlog with the project's timeline, for follow-up or progress estimation purposes.

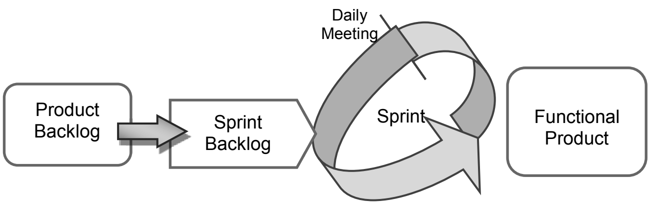


Fig. 5. Scrum Iteration (Adapted from [12])

Authors propose to use Scrum for mobile software development since phone applications are generally simple and activity-centered, dedicated to accomplish a very restricted number of actions, focused on user-experience, and can be developed by small teams in short periods. As a case study, authors defined a working model to be used in a classroom setting, establishing Sprint iterations of two weeks. Scrum teams are composed by groups of students, a certified Scrum Master facilitates the work, and a real-world client provides the requirements and validates the end product. The project consists on a mobile application for restaurants in Senegal that allows waiters to manage orders and bills more efficiently. This research paper includes one scholar case study and has been cited by 4 articles, including 1 case study not related to mobile software.

3.5 Scrum Lean Six Sigma

Scrum Lean Six Sigma (SLeSS) is an integration approach of Scrum and Lean Six Sigma proposed by Cunha et al. [13] for the development of embedded software for mobile phones. This philosophy enables the achievement of performance and quality goals, while progressively improving the development processes in a statistically-controlled basis. SLeSS picks up from the Scrum methodology already explained, but pursuing a combination of the effort and consistent deliveries of the Scrum Sprints with the continuous process analysis and improvement model represented by the 5-phase DMAIC methodology (Define, Measure, Analyze, Improve and Control). The Sprint Backlog is used not only to establish the objectives of the next iteration, but it is also carefully examined for statistic-based process improvement purposes (Figure 6). The implementation of SLeSS foresees an incremental approach, in which an Agile philosophy like Scrum is adapted to coexist with a planned-based methodology like Lean Six Sigma (LSS).

The case study offered involves a real-world project, developed in 6 months by a R&D laboratory. The product is an embedded software snapshot to be shipped with a cell phone, while the customer is the cell phone manufacturer. Each team consists of

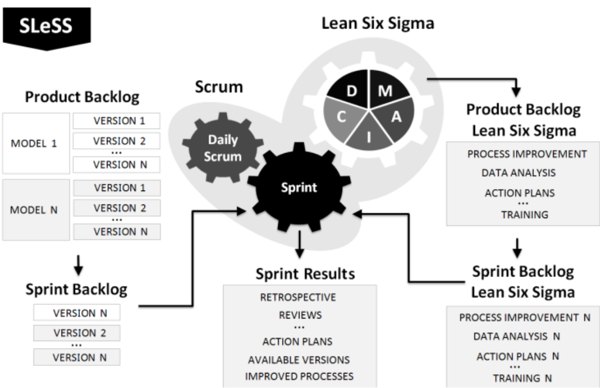


Fig. 6. Structure of the Implementation of Scrum Lean Six Sigma [13]

12 developers. SLeSS led to improvements the in process capability (a value that is translated into a reduction of defects per million of opportunities), and achieved a reduction in the working hours required to ship a functional product. In addition to this case study, SLeSS is cited by 1 work that is not a report of its utilization.

3.6 Other Relevant Works

Not all the mobile-specific software development methodologies are based on Agile methods. We can mention RaPiD7, which is based on Agile principles but focused on the authoring documentation of mobile projects rather than in actual product development [14]. Also, the Mobile Development Process Spiral was proposed to utilize a usability-driven-model to integrate mobile-specific usability matters into existing application development processes. This methodology is not Agile-based, but it supports iterations to ensure that requirements are addressed and validated [15]. Finally, the Intel Mobile Application Development Framework is an enterprise-oriented effort that focuses on evaluating the suitability of mobile applications to generate business value within a company. It prescribes guidance documentation, enabling technologies and supporting resources for conducting projects that adhere better to the organization's standards and best practices [16].

Another group of works report design practices or implementation guidelines for mobile products. They do not describe a comprehensive software development framework, but they provide development guidelines, design methodologies and best practices for mobile software that can be applied following any development process [17-21]; some of them are published online[1,2,3]. In several of these works, the mobile environment constraints (e.g., memory consumption, energy awareness, user interfaces) are considered for the hands-on implementation of suitable mobile-specific products.

Similarly, a product-oriented family of works concentrates on proposing software metrics specific to mobile applications. These metrics typically consider characteristics particular to mobile applications, and promote the customization of general-purpose product quality standards to suit the needs of mobile environments [22-25], including an attempt to create a relationship between quality factors and some of the development guidelines and best practices already mentioned [26]. Nevertheless, these papers are completely product-focused and do not show the link between the quality goals of the mobile product and the processes that have to be conducted to develop an application that satisfies these expectations.

4 Comparative Analysis

We are interested to study how each methodology addresses the different steps of the software development process. To perform a fair, practical analysis and comparison

[1] http://www.unifiedtestinginitiative.org/files/uti_best_practices_v1_final.pdf

[2] http://developer.android.com/design/index.html

[3] http://www.w3.org/TR/mwabp/

of the reviewed methodologies, we took as baseline a generic software development life cycle that consists of analysis, design, implementation, test and deployment phases, adding a category on measurement and continuous improvement. We surveyed the reviewed methods against this baseline, pointing out the way (i.e., an activity or principle) in which each methodology addresses each stage.

4.1 Analysis

The Analysis phase is well represented in all the surveyed methodologies: All of them count on a methodological way to explore the working setting, identify the customer and initialize the work settings. Mobile-D, MASAM and Hybrid conduct the assignment of roles and responsibilities in this early phase, focusing on creating self-organized teams and promoting people's interaction and close communication between roles and customers. In light of the Agile approach, requirements gathering and planning tasks are modest, concentrating on establishing an initial set of ideas that welcome and respond to change, rather than going through strict planning. Scrum-oriented methodologies propose an iteration-based planning effort that is updated and customized depending on the results achieved by previous iterations.

4.2 Design

Design efforts are evenly distributed through the five approaches. Once the system analysis is completed, different design tasks take place. By definition, Agile requires design to be flexible but carefully implemented, since good design facilitates quick reaction and reduces the impact performing eventual changes. Architectural design is underlined by Mobile-D, MASAM and Hybrid Methodology, proposing adequate guidance for applying incremental and product line design. MASAM, for example, supports a tool to translate design models into code. Scrum-based methodologies use the design phase to establish a perspective to process the existing Product Backlog. An additional point considered by Six Sigma is the identification of attributes Critical to Quality (CTQ) that represent what the customer cares most about. The CTQs are measured at this phase to establish a process capability and they will be measured in further phases to determine the impact of the resulting product on the customer's goals.

4.3 Implementation

Implementation phases under Agile approaches usually strive for frequently delivering working software products following a policy of intense collaboration among teams and close communication with the customer. Mobile-D and MASAM introduce typical Extreme Programming techniques. Hybrid Design performs the implementation activities within the so-called Development Engine for the creation of reusable software components, using Test Driven Development. Scrum and Scrum Lean Six Sigma consider the hands-on implementation of the work Sprint by the Scrum teams using any Agile implementation technique. Six Sigma in particular suggests carrying out an improvement phase to apply a solution that addresses the issues found in the Measure phase; nevertheless, the implementation phase does not prescribe specific activities and can be executed in any Agile way.

4.4 Testing

For the conduction of testing activities, Mobile-D, MASAM and Hybrid Design merge several tasks of the implementation phase to accommodate test-based development. With this, for every feature implemented, a test effort (e.g., a test case or test scenario) will exist automatically. Scrum methodology indicates that test efforts should exist in every Sprint, however, in the analyzed study, testing is not explicitly mentioned, so it is not possible to know what test efforts were conducted. Scrum Lean Six Sigma, on the other hand, indicates the existence of comprehensive test activities as part of the validation of the improvement phase.

4.5 Deployment

As mentioned in the implementation phase, Agile focuses on providing frequent deliveries of functional software versions. Based on this philosophy, for Mobile-D and Scrum, the deployment of a new software product means the delivery of a production-level snapshot. Scrum Lean Six Sigma establishes the deployment phase as a piloted solution that involves the utilization of the software product in a controlled setting with the purpose of observing its utilization. The piloted solution is accompanied by a new measure phase that gathers the necessary data for understanding the impact of the implemented solution on the process capability, and to maintain it under control. MASAM and Hybrid Design feature a commercialization task, which means preparing the product to work in a specific context, for example satisfying the policies of a country; this phase can be extended to ensure that the application remains in compliance with the policies given by the software distribution channels.

4.6 Measurement and Improvement

Measurement is a very important step in gaining understanding on what happens through the development process. This means that each methodology should offer a way to collect relevant information, set it down in metrics, study such metrics, and draw conclusions. Mobile-D and Hybrid Design recommend the implementation of project metrics and data collection practices, as well as celebrating learning sessions for process improvement. Scrum recommends "Sprint Retrospectives" as exercises to review the results of the finished effort, and identify lessons learned and opportunities of improvement for the next Sprint. For Scrum Lean Six Sigma, being a statistically-based methodology, Measure and Control phases are driven by data collection plans, which first establish the process capability and then measure the impact of the piloted solution. Finally, it is important to note that MASAM does not prescribe any effort related to measurement and continuous improvement.

5 Discussion: Accomplishment, Evidence and Evolution

The reviewed methodologies provide different sets of practices that aim to achieve fully functional mobile products on short development cycles. When available, case studies illustrate the fitness of those methods for assisting the development and delivery of a real project. The surveyed methodologies show convergent approaches based on the Agile home ground themes, nonetheless, they leave open how to address the environment-specific circumstances to create more suitable end products. Also, the review shows a lack of supporting evidence about the applicability of the methodologies in a non-scholar setting. Furthermore, there have been significant changes in the conditions through the span of time upon which the mobile environment the surveyed methodologies were developed. For instance, in 2004 concepts like location-based services, app stores, iOS, Android and other current key terms were emerging concepts or did not exist at all.

While the reviewed Agile-based frameworks deem to suit the needs of the mobile environment, we need further analysis to validate this claim. To outline the strategy to solve this question, we introduced a discussion [27] in four thematic areas, covered in the rest of this section.

A. Are Agile methods the best fit for the needs of the mobile business environment?

Agile methodologies establish their core values in the possibility of focusing on the working product, remaining close to the client, and responding efficiently to the change. Implementing Agile allows adapting processes and practices to the unsteady needs of the mobile domain. Mobile apps should be developed quickly and keeping up a low price to be successful in a market of millions of potential users with an offer of millions of products. Agile embraces effectively this business model to understand the market, structure the product and release it short time frames.

None of the reviewed frameworks question the suitability of Agile for the development of mobile apps. Instead, they focus on profiting from their advantages, and when necessary, authors fulfill their shortcomings incorporating directions provided by non-Agile development methodologies, for instance using plan-based methodologies or statistical quality control.

B. Are Agile methods addressing the needs of the mobile operative environment?

The reviewed Agile methodologies claim effectiveness on addressing the needs and constraints of mobile software products. However, it is hard to argue for a direct effect on the end product at the level of abstraction on which they are presented in their corresponding papers, mainly process-oriented. Considering the constraints of the mobile environment, it is not clear as to what problems can be injected if general-purpose software development processes are used, or why a software development process would help resolve these issues. For instance, if we consider a scenario in which one has to design an application to work in an environment with poor and intermittent connectivity, what development process (Mobile-D, Scrum, etc.) should be used? Why this choice would be an important factor in the success of the final outcome? Such questions are generally left open at the level covered by development frameworks.

Instead, an implementation-oriented approach can be found on the range of articles that focus on design practices that concern on solving, from a practitioner approach, the environment conditions that affect the mobile product, including interface design, usability guidelines, human-computer interaction and others. Regarding the mobile environment itself (efficiency, resource utilization, memory and CPU consumption, network utilization and battery usage) development guidelines are proposed at different levels (architectural, design, coding). Another family of works introduces evaluation metrics and other quantitative indicators that can aid the developer to assess the level of accomplishment and fitness of the end product for the execution environment from a practical point of view.

In summary, we can suggest that Agile collaborates to create a development lifecycle appropriate to meet the business needs of the mobile environment, design guidelines and best practices furnish mobile-specific development practices that aid to implement a product that is pertinent for this domain, and software quality metrics stand in the middle to monitor the activities and supply the data that support the measurement tasks required by the high level development process (Figure 7).

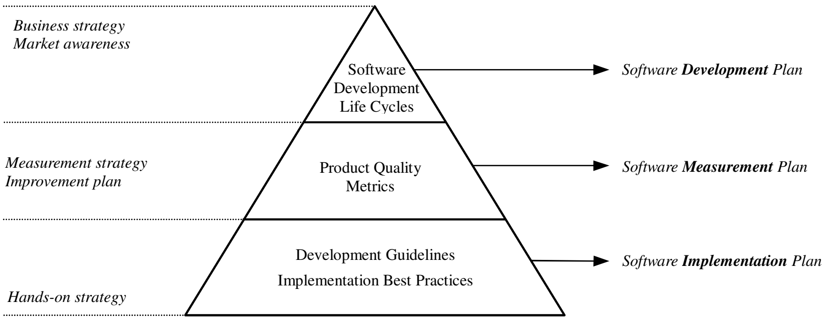


Fig. 7. Organization of Software Lifecycles, Product Assurance and Development Guidelines

C. Is there enough evidence about the real use of the proposed Agile methodologies?

It is not realistic to claim that the frameworks presented are applicable in a real environment without supplying empirical evidence. Besides Mobile-D, the rest of the Agile frameworks cannot argue to have major support on their adoption and implementation in real production environments. Mobile-D keeps records about software projects developed using this methodology, including works carried out both in research and large industry settings (ENERGI, F-Secure, Nokia, Philips) [28, 29]. Scrum and SLeSS present only one case study as part of the validation of their work, but citing literature does not show further implementations. MASAM and Hybrid Methodology do not show a case study, and are not cited by any other report. This situation challenges seriously their feasibility and potential success in a real setting.

Table 1 shows a summary of the documented instances of implementation of each methodology, including the case studies presented as part of the research paper in

Table 1. Agile-based Mobile Software Development Processes and their Implementations

Methodology	Year	Case Studies	Cited by
Mobile-D	2004	16	17
MASAM	2008	0	3
Hybrid	2008	0	9
Scrum	2010	1	4
SLeSS	2011	1	1

which they are introduced (citation count was taken from the digital libraries when available, otherwise from Google Scholar removing duplicated instances).

We expanded this analysis with a review of additional field studies (i.e., development surveys,[4,5,6] attempting to identify substantiation of the usage of a methodology or other development practices in true mobile software projects. However, the reviewed studies focus their interest on analyzing the operating system of choice, software development kits, type of applications produced, multiplatform development and other topics. Although they suggest a clear trend on shortening the development cycle and broaden the impact of the end product, further work is required to unveil the utilization of a consistent development methodology [30].

D. Have Agile methods kept up with the evolution of the Mobile environment?

When Agile methods were considered as the best fit for mobile development, the mobile business and development environment were different to the current one. Table 2 shows the mapping between the Agile home ground themes and the characteristics of the mobile software, made available in 2005.

Table 2. Mapping of Agile Ground Themes and Mobile Software Development Traits [8]

Ideal Agile Characteristic	Mobile Software Development
High environment volatility	Dynamic environment: hundreds of new mobile phones published each year.
Small development teams	Majority of mobile software is developed in micro or SME companies or development teams.
Identifiable customer	Potentially unlimited number of end-users
Object-oriented development	Java and C++ are mainly used.
Non-safety critical Software	Majority of existing mobile software is for entertainment purposes. Mobile terminals are not reliable.
Application level software	Mobile applications are stand-alone applications
Small systems	Size of mobile applications varies, but generally they are less than 10,000 lines of code.
Short development cycles	Generally mobile applications and services can be developed within 1-6 month time frame.

---

[4] http://www.infoq.com/news/2011/05/A-Survey-on-Mobile-Development

[5] http://www.gqs.ufsc.br/wp-content/uploads/2011/12/GQS-Workingpaper-002-2011-E-v10.pdf

[6] http://www.comp.nus.edu.sg/~damithch/df/device-fragmentation.htm

A decade of evolution on the mobile domain (software, hardware and business models) has brought significant advancements; therefore the current applicability of this mapping is controversial and invites to conduct an up-to-date discussion. To name only some of the differences of the current status of the mobile domain, we identify:

- While hundreds of new mobile models are still released each year, mobile developers also have well settled operating platforms (e.g., iOS or Android), that have solid software development kits (SDK) and APIs that facilitate the interaction with new device models.
- Mobile software is still developed by small teams and small-medium enterprises, but currently it is also part of major developments that involve large corporate teams.
- Nowadays, mobile applications spans not only in stand-alone applications but also interacting with other systems, collaboration tools, using heavily network and hardware resources, etc. This also implies that the mobile software product is not anymore small by definition.
- The range of applications deployed on cellular telephones now includes healthcare monitors[7], mobile banking[8] or earthquake alerts[9] that are required to meet strict standards to enter into service and cannot be categorized as non-critical software.

Finally, we observe that some Agile-based mobile software development frameworks try to enhance their methods by adapting practices from plan-based methodologies, for example, project documentation, traceability records, and other considerations. For instance, the latest Agile proposal, SLeSS, is an approach to relieve several short-comings from Agile by applying statistically-based quality control. This represents a complex merge of two different viewpoints: light-weight development practices (Scrum) and heavy quality control methodologies (Six Sigma). Following this way, other recent development methodologies have completely relegated the Agile approach [15, 16]. These examples reflect an identified decline on considering Agile as a silver bullet [31] and instead promoting a savvy strategy to decide when to use which practice evaluating its advantages and disadvantages, to decide how to apply and complement the Agile approach.

6 Future Work

To provide a more robust answer to our research question, there is a clear need of conducting evidence-based research that unveils what mobile development practices are actually used, in order to have a factual picture on how development teams are

7 http://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm263340.htm

8 http://www.sans.org/reading_room/whitepapers/ecommerce/security-mobile-banking-payments_34062

9 http://www.economist.com/blogs/americasview/2012/04/earthquake-warnings-mexico-city

conducting and managing mobile software projects. These insights may be gained as means of industrial surveys, interviews with mobile software managers and other empirical studies, and from the proactive discussion with the scientific community.

To this end, we will design and deploy an experiment that will supply the empirical data required to study the effectiveness of the Agile approach in a real industrial setting that implements mobile projects following this philosophy. The analysis will include process analysis, interviews with developers and gathering ad-hoc metrics [32, 33] in a co-located team of 25 members of different levels of seniority and experience [34]. The data collection plan will be executed during the implementation of an Agile-conducted mobile project intended to be delivered to the customer in 3 months. After the data collection, we will conduct an analysis to determine the level of accomplishment on product quality and customer satisfaction; finally we will look for correlations between these indicators and the selection and implementation of the Agile methodology.

7 Summary and Conclusions

The published development models specific for mobile applications agreed to identify Agile as the best fit to conduct a software project in a mobile context. They show convergent and complementary approaches based on the Agile home ground themes that can be selected depending on the organization's nature and the project's conditions and goals. At the level of abstraction presented in the development methodologies, Agile methods seem to bring value helping to keep up with the mobile business trends and marketability, while other mobile specific constraints like resource limitation are more likely to be managed via development guidelines.

The development methodologies reviewed in this paper call attention on the necessity of adapting the processes and practices to the evolving needs of mobile software, based on assumptions that must be reviewed as they may be currently obsolete, for instance, the fact that the mobile software product is small and non-critical. To solve this issue, current Agile-based mobile software development processes can be improved, updating practices, assumptions and recommendations to the current reality of the mobile domain, and also by adapting practices from plan-based methodologies. In other scope, current scientific documentation does not show a clear link between the proposed methodologies and their utilization in a production setting, and development surveys pay little attention on software development frameworks. Finally, the association between the Agile ground themes and the characteristics of the mobile software upon which these methodologies were built has changed significantly, urging the need of conducting evidence-based research to have an up-to-date discussion of such relationship.

The evolution of mobile platforms from being a simple communication tool toward becoming the primary end-user computing equipment requires researchers and practitioners to understand the context of the mobile domain to create the best strategies to develop mobile software. Under the premise of "high quality processes deliver high quality products", mobile Software Engineering still faces an extensive work load to determine what are the best processes and practices that facilitate the creation of high quality, successful mobile software products.

References

1. Zhifang, L., Xiaopeng, G., Xiang, L.: Adaptive random testing of mobile application. In: Proc. of the 2nd Int. Conf. on Computer Engineering and Technology (2010)
2. Rahimian, V., Habibi, J.: Performance evaluation of mobile software systems: Challenges for a software engineer. In: Proc. of the 5th Intl. Conf. on Electrical Engineering, Computing Science and Automatic Control (2008)
3. Spataru, A.C.: Agile Development methods for mobile applications. School of Informatics. University of Edinburgh, UK (2010)
4. Yamakami, T.: Business model engineering analysis on mobile client-side software platform strategies. In: Proc. of the 7th Intl. Conf. on Mobile Business, pp. 59–64 (2008)
5. Glissmann, S., Smolnik, S., Schierholz, R., Kolbe, L., Brenner, W.: Proposition of an m-Business procedure model for the development of mobile user interfaces. In: Proc. of the Intl. Conf. on Mobile Business (2005)
6. Hammershoj, A., Sapuppo, A., Tadayoni, R.: Challenges for mobile application development. In: Proc. 14th Int. Conf. on Intelligence in Next Generation Networks (2010)
7. Abrahamsson, P., Warsta, J., Siponen, M., Ronkainen, J.: New directions on Agile methods: A comparative analysis. In: Proc. of the Int. Conf. on Software Engineering (2003)
8. Abrahamsson, P.: Mobile software development: the business opportunity of today. In: Proc. of the Intl. Conf. on Software Development, pp. 20–23 (2005)
9. Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Koskela, J., Kyllönen, P., Salo, O.: Mobile-D: An Agile approach for mobile application development. In: Proceedings of OOPSLA 2004 (2004)
10. Jeong, Y.J., Lee, J.H., Shin, G.S.: Development process of mobile application SW based on Agile methodology. In: Proc. of the 10th Intl. Conf. on Advanced Communication Technology, pp. 362–366 (2008)
11. Rahimian, V., Ramsin, R.: Designing an Agile methodology for mobile software development: A hybrid method engineering approach. In: Proc. of the Second Intl. Conf. on Research Challenges in Information Science (2008)
12. Scharff, C., Verma, R.: Scrum to support mobile application development projects in a just-in-time learning context. In: Proc. of the ICSE Workshop on Cooperative and Human Aspects of Software Engineering, pp. 25–31 (2010)
13. Cunha, T., Dantas, V., Andrade, R.: SLeSS: A Scrum and Lean Six Sigma integration approach for the development of sofware customization for mobile phones. In: Proc. of the 25th Brazilian Symposium on Software Engineering, pp. 283–292 (2011)
14. Kylmäkoski, R.: Efficient authoring of software documentation using RaPiD7. In: Proc. of the 25th Intl. Conf. on Software Engineering, ICSE 2003, pp. 255–261 (2003)
15. Nosseir, A., Flood, D., Harrison, R., Ibrahim, O.: Mobile development process spiral. In: Proc. of the 7th Int. Conf. on Computer Engineering & Systems, pp. 281–286 (2012)
16. Doolittle, J., Moohan, A., Simpson, J., Soanes, I.; Building a mobile application development framework. Intel Whitepaper (2012), `http://communities.intel.com/docs/DOC-19555` (retrieved January 20, 2013)
17. Häkkilä, J., Mäntyjärvi, J.: Developing design guidelines for context-aware mobile applications. In: 3rd Intl. Conf. on Mobile Technology, Applications and Systems (2006)
18. Sá, M., Carriço, L.: Lessons from early stages design of mobile applications. In: Proc. of the Int. Conf. on Human Computer Interaction with Mobile Devices and Services (2008)
19. Ayob, N.Z., Hussin, R.C., Dahlan, H.M.: Three layers design guideline for mobile application. In: Proc. of the Int. Conf. on Inf. Management and Engineering, pp. 427–431 (2009)

20. La, H.J., Lee, H.J., Kim, S.D.: An Efficiency-centric design methodology for mobile application architectures. In: Proc. of 7th Int. Conf. on Wireless and Mobile Computing, Networking and Communications, pp. 272–279 (2011)
21. Kim, W.-Y., Park, S.-G.: The 4-Tier Design Pattern for the Development of an Android Application. In: Kim, T.-h., Adeli, H., Slezak, D., Sandnes, F.E., Song, X., Chung, K.-i., Arnett, K.P. (eds.) FGIT 2011. LNCS, vol. 7105, pp. 196–203. Springer, Heidelberg (2011)
22. Spriestersbach, A., Springer, T.: Quality attributes in mobile web application development. In: Bomarius, F., Iida, H. (eds.) PROFES 2004. LNCS, vol. 3009, pp. 120–130. Springer, Heidelberg (2004)
23. Ryan, C., Rossi, P.: Software, performance and resource utilisation metrics for context-aware mobile applications. In: Proc of the 11th IEEE Int Software Metrics Symposium (2005)
24. Dantas, V., Marinho, F., Costa, A., Andrade, R.: Testing requirements for mobile applications. In: Proc. of the 24th International Symposium on Computer and Information Sciences, pp. 555–560 (2009)
25. Mantoro, T.: Metrics evaluation for context-aware computing. In: Proc. of The 7th Int. Conf. on Advances in Mobile Computing & Multimedia (2009)
26. Marinho, E.H., Resende, R.F.: Quality factors in development best practices for mobile applications. In: Murgante, B., Gervasi, O., Misra, S., Nedjah, N., Rocha, A.M.A.C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2012, Part IV. LNCS, vol. 7336, pp. 632–645. Springer, Heidelberg (2012)
27. Corral, L., Sillitti, A., Succi, G.: Software Development Processes for Mobile Systems: Is Agile Really Taking Over the Business? In: Proc. of the 1st International Workshop on Mobile-Enabled Systems (MOBS 2013) at ICSE 2013, pp. 19–24 (2013)
28. Abrahamsson, P.: Agile software development of mobile information systems. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 1–4. Springer, Heidelberg (2007)
29. Hedberg, H., Iisakka, J.: Technical reviews in Agile development: Case Mobile-D. In: Proc. of the 6th Intl. Conf. on Quality Software, pp. 347–353 (2006)
30. Marko, K.: Application development in the age of mobility. Information Week Magazine, No. 1341 (August 20, 2012), ISSN 8750-6874
31. Janes, A., Succi, G.: The dark side of Agile software development. In: Proc. of Onward! 2012 (SPLASH 2012), pp. 215–228 (2012)
32. Corral, L., Sillitti, A., Succi, G.: Using a Partially Instantiated GQM to Measure the Quality of Mobile Apps. In: 25rd Int. Conf. on Software Engineering and Knowledge Engineering (2013)
33. Sillitti, A., Janes, A., Succi, G., Vernazza, T.: Measures for Mobile Users: an Architecture. Journal of Systems Architecture 50(7), 393–405 (2004)
34. Coman, I., Sillitti, A., Succi, G.: A Case-study on Using an Automated In-process Software Engineering Measurement and Analysis System in an Industrial Environment. In: 31st International Conference on Software Engineering (2009)

Design and Development Guidelines for Real-Time, Geospatial Mobile Applications: Lessons from 'MarineTraffic'

Dimitrios Zissis, Dimitrios Lekkas, and Panayiotis Koutsabasis

University of the Aegean, Department of Product and Systems Design Engineering,
Syros, GR-84100, Greece
{dzissis,dlek,kgp}@aegean.gr

Abstract. The development of real-time, geospatial mobile applications poses particular challenges regarding their interaction design and technical implementation. In this paper, we present insights into the design and development of the mobile version of MarineTraffic (marinetraffic.com), which is an open, community-based project that provides real-time geospatial information about vessel movements and port traffic. During the technical development and deployment of MarineTraffic mobile, we identified a number of 8 principles and 25 guidelines that had to be followed in order to improve the user experience and tackle technical issues. We discuss these guidelines with respect to implementation examples and experiences from the specific mobile applications and we suggest that these guidelines have to be kept in mind for the design and development of similar mobile applications.

Keywords: Mobile application design guidelines, mobile interaction design, mobile application development, MarineTraffic, application design.

1 Introduction

Currently an overall of 10% of global website views are from mobile devices (smart phones and tablets). On Christmas day 2012, more than 50% of online activity came from mobile devices [1]. It is interesting though that, according to this study, the vast majority of mobile activity, measured at 98%, was from native applications. Only 2% of mobile activity was conducted through the mobile web. According to the NPD group survey, 37% of consumers who used to access content on their PCs switched to their tablets and smartphones, with the top two activities being web browsing and Facebook [2].

When web sites rest mostly on textual and image content, do not involve many network queries and make limited use of interactive technologies, then their conformance to web usability (e.g. Research-based Web Design and Usability Guidelines [3] and accessibility (Web Content Accessibility Guidelines [4]) guidelines can suffice for a satisfactory mobile user experience. This is not the case for more sophisticated web designs that involve: real-time geospatial information visualization,

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 107–120, 2013.

manipulation and navigation on geographic maps, layered information presentation and user content uploads and ratings. These websites are almost completely inoperable via mobile browsers, as differences in compatibility hugely deter their interactivity often making it impossible to perform even basic map functionality. These incompatibility issues are added on to typical mobile devices' limitations in CPU, memory, network availability and screen size.

The design and development of native mobile applications is complex and challenging [5]. It concerns developing for a number of different operating systems and devices with various characteristics, in a diverse set of programming languages. In such a fragmented field, designers and developers have to comprehend the restrictions of the technology (e.g. small screen, memory, CPU) and the particular technical characteristics of mobile operating systems (e.g. iOS, Android, Windows phone, etc.) [6]. In addition, mobile devices are characterized by a unique synthesis of interaction affordances that can actually transform the user experience when compared to desktop platforms including: gesture-based, multi-touch interaction with digital content; location awareness and subsequent service and content adaptation; advanced sensing capabilities (with embedded devices like: accelerometer, gyroscope, GPS, camera etc.); multimedia (photos, sound, video) capturing and sharing [7–9]. Therefore, mobile application development requires a whole new way of thinking in respect to interaction design and HCI, as well as to software development.

In this paper we present the case of the design and development of the mobile version of MarineTraffic (marinetraffic.com), which is an example of a real-time, geospatial, community-based web service that allows users to view vessel information, positions, routes and port traffic in real-time. Our work and experience with MarineTraffic mobile has yielded a number of respective principles and guidelines (8 principles and 25 guidelines). We discuss these guidelines with respect to how we implemented them in terms of the specific mobile applications and we suggest that these guidelines have to be kept in mind for the design and development of similar mobile applications.

The paper is structured as follows: Section 2 briefly presents the MarineTraffic service, outlining its basic characteristics that have to be transferred to its mobile counterparts. Section 3 presents the design and development of MarineTraffic mobile applications based on a number of principal development goals that gradually transformed into a set of design and development guidelines that were followed throughout the process; in this section we view these guidelines in terms of examples and experiences from the mobile applications developed. Finally, section 4 presents the discussion of our work with respect to related work and outlines its potential contribution for mobile design and development of similar applications.

2 Marine Traffic Web Application

Marine Traffic is part of an open, community-based project that provides real-time information to the public, about ship movements and port traffic, mainly across the coast-lines of many countries around the world. The initial data collection is based on the Automatic Identification System (AIS). As from December 2004, the International Maritime Organization (IMO) requires all vessels over 299GT (Gross Tonnage) to carry an AIS transponder on board, which transmits their position, speed and course, among some other static information, such as vessel's name, dimensions and voyage details. This information is broadcasted through public radio frequencies and is accessible to any interested party with the required equipment. Using an AIS receiver and dedicated software we are able to store this information in a database and then provide it to the general public through marinetraffic.com.

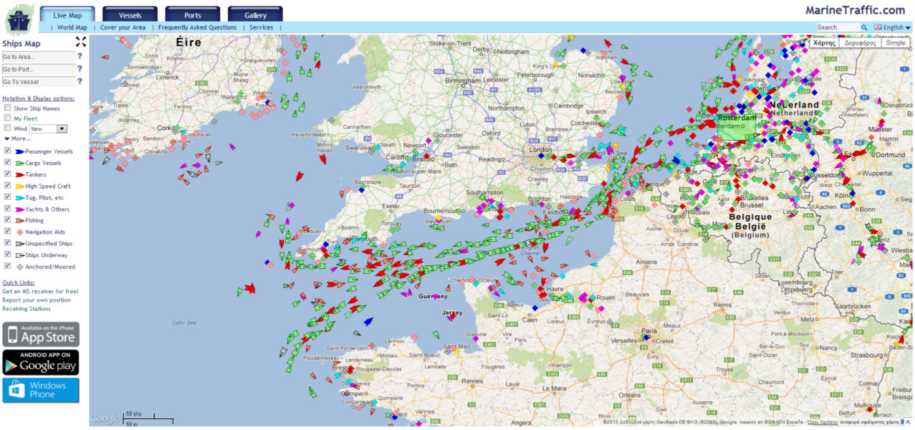


Fig. 1. Aspect of Marine Traffic Web Application (http://marinetraffic.com)

The MarineTraffic Web application is an AJAX (Asynchronous JavaScript & XML) map mashup, that overlays vessel and traffic information on an interactive map. MarineTraffic makes use of a base map, a geocoder, an additional layer of presentation information and a Web interface. In general, Web mashups are mostly realized by web pages that leverage script languages such as JavaScript, which enables better user interactivity by locally executed functions and the dynamic loading of data from web services [10].

MarineTraffic provides a number of services including: real time vessel tracking, vessel search, vessel track history, direct navigation to vessel, port or area, distance calculation, vessel photographs, user ratings of these photos and additional vessel related information. MarineTraffic also provides a number of personalized services such as fleet management, vessel related notifications and specific area tracking. MarineTraffic heavily rests on network availability to display and update vessel movements and port traffic. In addition, MarineTraffic is a community-based (or crowd-sourcing) application that also rests on users who can contribute to the service infrastructure by adding nodes of data collection (according to instructions provided from the website), as well as content by uploading and rating ships' photos.

MarineTraffic is a highly popular service, with approximately 14 million visitors per month and 50 million page views. From these, approximately 1.5 million visits originate from mobile browsers (mobile phones and tablets). Before the design and development of MarineTraffic mobile applications, a large number of users had reported that viewing MarineTraffic from specific mobile web browsers offered a degraded experience (actually in many cases the user interface was completely unusable). We have to note that there are many thousands of users that access MarineTraffic at sea via their mobile devices. For example, it is typical user behavior to take photos of vessels at sea and post them to MarineTraffic, or request specific vessel information including its current coordinates, heading, speed, port arrival time and related information.

Poor user experience over mobile devices is mostly due to JavaScript compatibility issues with the AJAX map mashup. Unfortunately, websites containing AJAX map mashups are often dysfunctional when viewed via mobile browsers. Map gestures and interactions are interpreted inconsistently, mostly due to differences in JavaScript implementations)[11][12], buttons are pressed with difficulty, and the user experience is hampered by the narrow layout making it almost impossible to perform even basic map functionality[13]. Many map providers only offer a degraded version of their map for a specific subset of browsers if any. In many occasions maps have been completely unavailable on specific mobile browsers [14][13].

It was soon obvious that native mobile applications of MarineTraffic had to be built, so that mobile users would not be excluded from the service. Although time-consuming and sometimes a repetitive task, the development of native mobile applications not only makes it possible to overcome many of the interaction issues that deteriorate the user experience through the mobile web browser, but provides the development teams with the ability to program at a lower level, thus increasing application execution speed and performance, while gaining access to a wide array of device specific functionality.

3 Marine Traffic Mobile

3.1 General Design Goals

The general design goals of Marine Traffic mobile (MT mobile) were to:

1. Make use of the most popular functions of MarineTraffic web service; this selection was largely based on reviewing the navigation paths users commonly took while using the online service.
2. Get rid of traditional desktop-oriented design elements like buttons and menus. Instead, make use of established user interface design elements of mobile platforms.
3. Exploit mobile device sensing capabilities like: location awareness (GPS), accelerometer, gyroscope, and camera available on the mobile platform to offer a location-aware (context-aware) service.
4. Exploit gesture-based interaction affordances of mobile devices by making a careful selection of appropriate gestures through the conduction of simple tests for finding/remembering gestures suggested by Saffer [15].

3.2 Overview of the Marine Traffic Mobile Application

MT mobile is currently available on the Android, IOS and Windows Phone mobile platforms (Figures 2, 3, 4). These applications attempt to imitate the look and feel of the MarineTraffic web service, customized for improved interactivity and performance on each mobile platform. The basic functions available for MT mobile include:

- The ability to track live vessel positions, reported by more than 80,000 vessels per day on an interactive map. The application provides worldwide coverage of more than 3,000 ports and a significant number of open-sea areas.
- Search for current conditions in ports, vessel details, historical data and estimated time of arrivals. Port arrivals and departures are recorded in real-time.
- Search for any vessel in range but also historical data (last known positions, last port visited, track etc).
- Browse the Photo gallery with nearly 1,000,000 pictures of vessels, ports and lighthouses. Users can upload photos and rate available photos.

At the first user interaction with MT mobile, the JavaScript map is loaded and vessel information is retrieved via an asynchronous XML web request. During this first load a user is prompted to permit MT mobile to access his location so as to enable a location aware experience (load map on the user's location); however users may opt out at any time. Vessel icons are pinned onto the map at the geographic coordinates of the vessel and icon coloring is selected according to vessel type. To avoid making continuous web requests, MT mobile retrieves vessel positions for a larger map segment than visible (depending on the available device screen dimensions). Additional requests are made only when the user navigates out of this larger map segment. A user can interact with the map using most common multi-touch screen map interactions and with any element on the map by tapping on it. During interaction with the map, the user can request a "refresh" of vessel positions by taping the refresh icon at any time. MT mobile provides a clustered view of vessel positions when users selected zoom level is small, in order to avoid clutter; this also occurs in highly active areas such as popular ports.

When a user taps on a vessel icon, basic vessel information is presented with a link to additional vessel information. If a user selects to view the additional details a panorama page (a page panning more than one screen where a user can swipe left/right to view as a whole) is presented containing vessel history, vessel details but also vessel photos. A user can swipe left/right between vessel pages containing detailed, information, vessel track history and photos. At any time, a user can select to return to the original map or search for an additional vessel. A user can search for any vessel or port (XML web request) by tapping on the application bar available across the bottom of the screen. A user is provided with a list of matches for his query and can select any item from this list. This list is only fully populated when a user scrolls beyond viewable items. When a user selects to view a vessel or a port, he/she is presented with details, images and relative information and from there can select to load map on this location. The basic MT mobile functions, user gestures and technical issues are listed in Table 1.

Table 1. Overview of basic MT mobile functions, gestures and technical issues

Function	Gesture	Technical Issues
OnLoad	n/a	Vessel and Map Data Retrieval
Navigate	Traditional map gestures	Incremental Map and Vessel Information Retrieval
Go To Position	Tap (option on application bar)	Enable location services on application and loads map on users position(context aware)
Zoom In/ Zoom Out	Pinch Open/Close	Clustered View
Examine Vessel	Tap, Swipe	PushPin annotation containing vessel basic details. Option to visit vessel detailed page
Refresh	Tap	Incremental Vessel Postion Retrieval
Search Ves-sel/Port	Tap	Generate Response . Go to details and option-al position on map
Customize Map	Swipe (Option Page ToggleS-witch)	User can customize application (show satellite imagery, vessel positions, vessel types etc)

MT mobile displays real-time data on an interactive map, which generates a number of particular requirements for a mobile application. The service has a heavy network dependency as users tend to often load many different views requiring the constant delivery of additional information. This service needs to continuously provide users with accurate data on constantly changing vessel positions. As network delays and drops are common in mobile application Internet access and the application needs to remain responsive even during these conditions, it was necessary to find the correct balance between data freshness, network consumption, and interface responsiveness. We needed to pay special attention to map loading issues and perform all other functions on asynchronous background threads providing users with incremental results. Thus, vessel loading and other processing were performed in the background. In addition, results had to be presented in an incremental fashion and are prioritized. We also exploited local mobile storage to cache static vessel information: users tend to re-view vessel and ports they are interested in, thus this information is cached locally.

During the design of the mobile versions of MT special interest was paid to providing personalized services for mobile users. A number of services were developed, including a "what's near me" service, that shows information of ports and vessels close to them and a "where is my ship" service, providing passengers with the ability to closely track a vessels position. Specifically for vessel navigators MT provides service for the calculation of the closest point of approach (CPA) and time to closest point of approach (TCPA). In addition, users can contribute information to these services, by taking photos and providing ratings (thus supporting crouwdsourcing functions in a similar way to [16]).

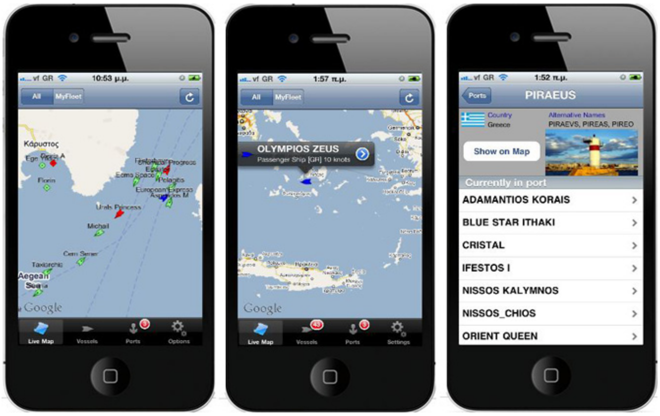


Fig. 2. MarineTraffic on Iphone Platform

4 Design and Development Guidelines

We organized the aforementioned requirements of MT mobile to the following 8 principles and 25 guidelines for design and development.

4.1 Principle 1. Promote Application Responsiveness

Users expect an application to remain responsive at all times. Although a tolerable waiting time for web users seems to be approximately 2 seconds [17], this is unacceptable delay to thin-client applications [18]. Response delay less than 1 second is noticeable to annoying and more than 1 second is potential threat to mobile user experience [18], [19]. Thus, smartphone apps strive for crisp response of less than 150 ms [20]. Responsiveness is closely related to performance: performance limitations are inevitable on mobile devices but these must not limit application responsiveness. Guidelines:

1.1. Use background threads: To the best extent possible, keep all processing that will affect the user interface on background threads e.g. MT mobile asynchronously attempts to retrieve vessel information updates.
1.2. Display results incrementally: The main idea is to show the most important pieces of information first and improve on quality as resources become available; e.g. use multi-resolution encoding and make progressive transmission and computations [6]. This is done in various aspects of MT mobile, specifically when loading long lists and image thumbnails only the “in-view” elements are originally loaded.
1.3. Make all non-UI related function calls asynchronously: An asynchronous call ensures that your active execution thread never blocks for a significant amount of time.

1.4. Use animations with care: Although animations are engaging, these can quickly make the UI unresponsive. This issue generated a lot of debate in the design phase of MT mobile, as a balance was sought between engaging animations and application responsiveness.
1.5. Identify and correct memory leaks: Mobile applications have a finite amount of memory. Memory leaks are one of the major causes of memory exhaustion problems. Take care to flush memory of unnecessary data e.g. MT disregards all information after a page is navigated from.

4.2 Principle 2. Content First

Mobile applications require a minimalistic approach about navigation that emphasizes on providing content first. This is also documented in a number of more general mobile design guidelines, for example the two out of 6 user-centered design guidelines proposed by Cerejo [21] are: '1. Minimize navigation' and '2. Prioritize content'. Guidelines:

2.1. Present the most important information from the user perspective: This requires prior knowledge of what information is important for users and requires information prioritizing. User-centered methods like card sorting [22] can be of use in this. MT mobile only loads information on user request.
2.2. Provide affordances for user comprehension with appropriate use of media for presenting content: Even when text content is well-written and understandable by users, it takes much space and is not engaging, so we need to try to provide visual designs of textual information to afford (in the sense of affordances proposed by Norman [23] for user comprehension. For example, the icon used for representing a vessel on the map, affords user comprehension about: type (passenger, cargo, tanker, etc.), size, bearing, status (moving or stopped) and projected course orientation.
2.3. Promote highly user-rated content. This is particularly important for community-based applications that rest on crown-sourcing and can dynamically select content based on user ratings. When the user is viewing a vessel, MT mobile dynamically loads the most popular vessel photo as a background image for the information page; this is also done on the port page (Figure 3). Also, users can browse all vessel photos sorted by their ratings.

4.3 Principle 3. Promote Robustness

Robustness is important for user experience and has been referred to as important usability principles in popular HCI textbooks at the early days of Graphical User Interfaces (e.g. by Dix et al [24]). Incorrect memory & CPU usage, network and bandwidth changes can often occur and crash a mobile application. Develop your application so that it can deal with all kind of exceptions. Mobile application crashes must be avoided at all costs. Guidelines:

3.1. Ensure system recovery from errors. This is achieved through exhaustive exception handling: uncaught exceptions can make your application to crash abruptly.
3.2. Provide meaningful user feedback in cases of errors. Ensure that the user is informed with understandable messages in cases of errors/exceptions and that he is provided with appropriate options.
3.3. Monitor network and bandwidth changes: To adjust processing accordingly. Before any network related query MT has to pass a number of validations so that no web exception is thrown that could potentially crash the application.

4.4 Principle 4. Respect User Expectations

Many mobile users are familiar to some extent to typical user interface / interaction conventions and responses, before making use of any particular mobile application. Therefore, a user expects from any mobile application to respond to ways he or she is accustomed to. Guidelines:

4.1. Make use of device (hard) buttons consistently: for example, a back or home button on the device must respond in the way that the user is accustomed to.
4.2. Provide timely feedback: for example progress indicators and appropriate messages when the application delays to respond.
4.3. Maintain the flow of the application throughout the interaction. For example, avoid breaking the application interaction experience by requesting the user to visit external webpages, to download and view files, etc.
4.4. A user has expectations about the look and feel of a native application for the particular device and OS. In respect to this principle MT mobile has followed a separate design approach for each platform (iOS, Android, Windows Phone) making use of particular design guidelines provided by each vendor. For example, MT mobile for Windows phone makes use of windows metro design and Bing maps in contrary to MT mobile for android that is built on Google maps.

4.5 Principle 5. Expect and Design for Latency

Latency is the time it takes for a single data transaction to occur and it is measured in milliseconds (ms). In a CNET article, Marissa Mayer, Vice President of Search Product and User Experience, Google, said that when the Google Maps home page was "put on a diet" (shrunk from 100K to about 70K to 80K), traffic was up 10% the first week and grew 25% more in the following three weeks [25] [26] Network delays and drops are common in mobile application Internet access. Guidelines:

5.1. Isolate UI operations from the network to your best effort: To avoid delays or crashes due to network loading data.
5.2. Only load image thumbnails and let user select to view larger images: this requires design provisions at the web service side to provide mobile apps with appropriate images. MT mobile only loads thumbnail version of user

uploaded photos until a larger size is specifically requested by the user. To minize network dependency and data download size, MT mobile makes use of lazy load algorithms [27].

4.6 Principle 6. Balance Energy Consumption

Energy consumption is one of the most important concerns of mobile application development and can result to poor user experience. Guidelines:

6.1. Perform CPU power intensive tasks, only when necessary.
6.2. Use orientation changes with care: Orientation changes put a heavy performance burden on the device processor; if they are not necessary, do not support them.
6.3. Exploit local storage: in order to minimize network connections. MT exploits local storage to cache static vessel information; thus minimizing unnecessary information retrieval.

4.7 Principle 7. Exploit Mobile Interaction Techniques

Mobile users have different behavior patterns than desktop or laptop users, which are to some extent cultivated by their interaction with the mobile device. Understand and design for these. Each version of MT mobile (android, iPhone & windows phone) takes a platform specific approach to designing application interactions e.g. windows phone supports panorama and metro UI interactions. Guidelines:

7.1 Web design conventions can be replaced with mobile applications' counterparts: For example, replace zoom, menu bars and buttons with pinch, swipe and tap respectively (Table 1).
7.2 Make use of device sensing affordances to provide contextual interactions: E.g. make use of the camera to add a photo, or make use of the GPS to present information based on the user location.
7.3 Avoid using heavy input methods: Use techniques to make text entry more efficient, and require it only when necessary. When possible, use smart algorithms to help text input.

4.8 Principle 8. Carefully Design Application Navigability

When designing websites we develop sitemaps and allow user to move using links between various webpage of our websites. For each such page our browser makes a new request and old pages are discarded. In mobile applications such pages remain dormant in memory as these as stateful devices. Guidelines:

8.1. Do not create a new mobile page when you can simply return to the previous one that is dormant in memory. This is one of the principal reasons of memory leaks.
8.2. Design navigation paths so that the creation of new pages is minimized.

Fig. 3. MarineTraffic on Windows Phone. Port Information Page.

5 Discussion

5.1 Related Work

A number of guidelines have been proposed to assist practitioners in mobile application design and development. Most mobile OS vendors have issued design guidelines for mobile application design and development: Apple iOS Humane Interface Guidelines [28], Android Design Guidelines [29], MS Windows Phone User Experience Guidelines[30]. These guidelines refer to a large number of issues specific to the software platforms and are certainly the starting point of study for any designer/developer. These sets of guidelines continue to be updated mainly to provide conformance to new versions of the operating systems.

These guidelines mainly refer to general design principles and to specific user interface elements and functions like icons, widgets and notifications. They also tend to emphasize design over development: this is natural since their aim is not only to provide help to potential designers and programmers but also to show off good and appealing design practices, and if possible characterizing each platform. However, proposed guidelines do not refer to specific types of mobile applications – with the exception of recent guidelines published by Apple for routing apps (on December 17, 2012); this may be the beginning of issuing guidelines for specific types of mobile apps from industrial vendors.

A number of mobile app designers have also proposed relevant guidelines. Weevers [31] proposes seven guidelines for designing high-performance mobile user experiences: (1) Define UI brand signatures; (2) Focus the portfolio of products; (3) Identify the core user stories; (4) Optimize UI flows and elements; (5) Define UI scaling rules; (6) Use a performance dashboard; (7) Champion dedicated UI engineering skills. We find these guidelines really useful especially for organizing team work with a focus on the design aspects of the development lifecycle. In addition Weevers's guidelines are general and relevant to any type of mobile app that offers some kind of business service.

Matzner [32] presents 10 mistakes to be avoided during the design and development of mobile apps. All are important to consider, however we distinguish the following: ‘1. Don’t begin wireframes or designs without a flowmap’, ‘6. Don’t leave users hanging’, and ‘7. Don’t blindly copy style from other operating systems’, which are relevant to our proposed principles: ‘1. Promote Application Responsiveness’, ‘2. Content first’, ‘8. Carefully design application navigability’.

Cerejo [21] discusses a user-centred approach to Web design for mobile devices. He outlines an iterative lifecycle process and presents a number of important guidelines for design: ‘Simplify Navigation’, ‘Prioritize content’, ‘Minimize user input’, ‘Design for intermittent connectivity’, ‘Offer cross-channel consistency and integration’. From all available sets of guidelines this is probably one that is most relevant to our work, since that all these guidelines are relevant to our proposed set.

There are too few works on design and development guidelines for geospatial mobile apps with the characteristics of marine traffic. Samet et al [33] discuss the porting of the Web-based application of NewsStand, a mapping application that enables searching for spatially-referenced news information, to a native App. They identify the issues of: how to compensate for the absence of a hovering action; the implementation of an intuitive mechanism for undo; how to integrate an interaction restriction to one hand coupled with use of the thumb as the pointing mechanism; and the formulation of navigation shortcuts to avoid excessive network traffic. In our work we have not met the first two issues, and we have provided guidelines for the latter two in our best attempt to exploit mobile device affordances.

The focus of our approach is quite different with respect to these works since that: (a) we refer to a specific type of mobile apps: real-time, geospatial mobile applications; and (b) we attempt to connect design and development issues, since that any product design process requires an appreciation of the way the product will be built. Our approach is largely based on our experience with the design and development of the mobile applications of MarineTraffic. The characteristics of MarineTraffic called for a demanding approach to application design and development that sought to find the fine balance between application responsiveness, data freshness while providing a valued mobile experience. Our proposal consists of a number of 8 principles and

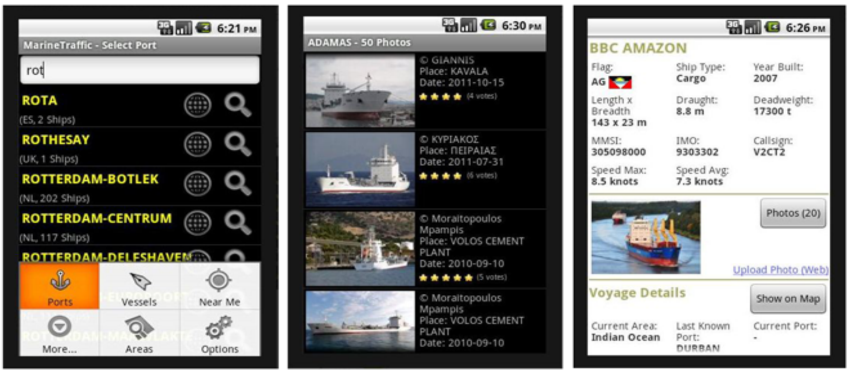


Fig. 4. MarineTraffic android version

25 guidelines that collectively attempt to define an overarching approach to designing such applications. We also provide examples of applying these principles and guidelines for MT mobile.

5.2 Outline and Potential Contribution

Mobile devices present a whole new ballgame for which web designers, but even traditional programmers, are not well prepared. Designers and developers from a web background need to conceive the differences in design created by network delays, network costs and screen size. The interface has to remain responsive regardless of the information retrieval or network latency. But most importantly mobile devices present an array of functions that can increase the user experience in methods not available on traditional PCS. Even for traditional application developers, mobile application development is a challenging task. Traditional software developers need to apprehend the complexity of memory management and CPU restrictions to maintain application responsiveness [6]. Without special programming considerations, such applications will exhibit unacceptable user responsiveness when resources are slow.

The design and development of real-time, geospatial, community-based mobile applications has to cope with particular challenges: (a) the ineffectiveness of current web design guidelines for usability and accessibility for this particular type of applications; (b) the technical issues for mobile application development that stem out of the requirement of constant geospatial information update and support for user-generated content over the Internet; (c) the exploitation of interaction affordances of current mobile devices to maximize the user experience.

The paper makes two distinct contributions to the field. First, we provide an overview of the complexities and perplexities of developing interactive real-time, geospatial, community-based mobile applications, as documented for the case of MT mobile. Second, we propose a number of design principles and development guidelines that stem out of the experience of MT mobile. Our work strongly couples design and implementation issues and we envisage that this orientation can be useful for other practitioners.

References

[1] MixPanel, Celebrating Christmas with Data (2012)

[2] N. Group, Connected Intelligence Application and Convergence Report (2013)

[3] U.S. Department of Health & Human Services, Research-Based Web Design and Usability Guidelines (2006), `http://www.usability.gov/guidelines/index.html` (accessed: February 15, 2013)

[4] W3C, Web Content Accessibility Guidelines (WCAG) 2.0. W3C Recommendation 11 December 2008 (2008)

[5] Charland, B., Leroux, A.: Mobile Application Development: Web vs. Native. Communications of the ACM 5(54), 49–53 (2011)

[6] Forman, G.H.: Wanted: Programming Support for Ensuring Responsiveness Despite Resource Variability and Volatility (1998)

[7] Ferreira, D., Kostakos, V., Dey, A.K.: Lessons Learned from Large-Scale User Studies: Using Android Market as a Source of Data. IJMHCI 4(3), 16 (2012)

[8] Corey, G.: Nine Ways To Murder Your Battery (These Are Only Some Of The Ways). In: Battcon 2010 (2010)

[9] Reddy, M., Mun, S., Burke, M., Estrin, J., Hansen, D., Srivastava, M.: Using Mobile Phones to Determine Transportation Modes. ACM Transactions on Sensor Networks 6(2) (2010)
[10] Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.): WISE 2008. LNCS, vol. 5175. Springer, Heidelberg (2008)
[11] Google, Google Maps API, Google Developers (2012), https://developers.google.com/maps/faq (accessed: January 18, 2013)
[12] Sensa Developer Forum, PR3: Google Map does not respond on Map control click events in mobile Browsers, Sencha Touch 2.x Forums / Sencha Touch 2.x: Bugs (2012), http://www.sencha.com (accessed: January 18, 2013)
[13] Blagdon, J.: Google enabling Maps access for Windows Phone after uproar. TheVerge (2013), http://www.theverge.com/ (accessed: January 18, 2013)
[14] Kunito, G.: Issues for mobile Ajax for cellular users, Hikari-no-oka, Yokosuka, Kanagawa, 239-0807 Japan (2012)
[15] Saffer, D.: Designing Gestural Interfaces. O'Reilly Media (2008)
[16] Shah, S., Bao, F., Lu, C.-T., Chen, I.-R.: CROWDSAFE: Crowd Sourcing of Crime Incidents and Safe Routing on Mobile Devices, pp. 521 – 524 (2011)
[17] Forrester Research, Inc., eCommerce Website performace Today, An Updated Look At Consumer Reaction To A Poor Online Shopping Experience (2009)
[18] Tolia, N., Andersen, D.G., Satyanarayanan, M.: Quantifying Interactive User Experience on Thin Clients. Computer 39(3), 46–52 (2006)
[19] Abolfazli, S., Sanaei, Z., Shiraz, M., Keshavarz, H., Gani, A.: An Evolutionary Study of Rich Mobile Applications. CoRR (2012)
[20] Saeid Abolfazli, A.G., Sanaei, Z., Shiraz, M., Keshavarz, H.: An Evolutionary Study of Rich Mobile Applications. CoRR, vol. 1206.6222 (2012)
[21] Cerejo, L.: A User-Centered Approach To Web Design For Mobile Devices. Smashing Magazine (2012), http://mobile.smashingmagazine.com (accessed: March 15, 2012)
[22] Hudson, W.: Card Sorting. In: The Encyclopedia of Human-Computer Interaction, 2nd edn. Interaction Design Foundation (2012)
[23] Norman, D.: The Design of Everyday Things (1988)
[24] Dix, A., Finlay, J., Abowd, G.D., Beale, R.: Human Computer Interaction, p. 832. Prentice Hall (2003)
[25] Nokia Siemens Networks, Latency- The impact of latency on application performance (2009)
[26] Farber, D.: Google's Marissa Mayer: Speed wins. CNET News.com (2006), http://www.zdnet.com/ (accessed: February 09, 2013)
[27] Cameron, R.: Pro Windows Phone 7 Development. Apress, Berkeley (2011)
[28] Apple, iOS Humane Interface Guidelines (2008)
[29] Google Inc., Android Design Guidelines
[30] Microsoft Corporation, Microsoft Windows Phone (2012)
[31] Weevers, I.: Seven Guidelines For Designing High-Performance Mobile User Experiences. Smashing Magazine (2011), http://uxdesign.smashingmagazine.com/ (accessed: March 15, 2013)
[32] Matzner, R.: Designing a Mobile App? Don't Make These 10 Mistakes. Mashable.com (2012), http://mashable.com/ (accessed: March 15, 2013)
[33] Samet, H., Adelfio, M.D., Fruin, B.C., Lieberman, M.D., Teitler, B.E.: Porting a web-based mapping application to a smartphone app. In: Proc. of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2011, p. 525 (2011)

Mobile Cloud Gaming: Issues and Challenges

Omar Soliman[1], Abdelmounaam Rezgui[1], Hamdy Soliman[1], and Najib Manea[2]

[1] Department of Computer Science & Engineering, New Mexico Tech
Socorro, NM, 87801, USA
{oms,hss}@nmt.edu, rezgui@cs.nmt.edu
[2] University of New Mexico, Los Lunas, NM 87031, USA
nmanea@unm.edu

Abstract. Recent developments in mobile, cloud, and graphics processing technologies have enabled *mobile cloud gaming*, a gaming model where players use mobile devices to play graphics-intensive games that run remotely on cloud servers. This delivery paradigm is called *Gaming as a Service* (GaaS). GaaS is used to stream computer games across the Internet. It gives rise to various technical, legal, and ethical issues. In this paper, we present the current state of the art in GaaS along with open issues and research challenges.

Keywords: GPU service, cloud gaming, on-demand gaming.

1 Introduction

Gaming as a Service (GaaS) is a relatively new concept in the field of computer science and cloud computing. It involves providing video games on-demand to consumers through the use of cloud technologies. One benefit of cloud technology is the transfer of computation from a relatively weak user device (or thin client) to more powerful cloud servers. Even when a user's device is appropriately powerful, GaaS can reduce power consumption and provide other cloud services (e.g., storage).

1.1 Background

The effects of Moore's Law [9] have led to successively cheaper and more powerful computers in a cyclic fashion since the mid-20th century. Since the advent of the ARPANET and its evolution, the Internet has grown at a similar rate [21]. As Moore's Law allowed computer access to a greater population, demand for Internet service also grew. Dial-up access evolved into broadband to serve more customers who now have more powerful machines that can run more compute-intensive applications. Meanwhile, technical progress led to smaller and more portable computers, which needed portable Internet. This was answered through the development of wireless Internet access technologies. These mobile devices continued to shrink and have become more powerful contributing in millions of users migrating from traditional computing platforms to "smart" phones and

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 121–128, 2013.

tablets. In 2012, more than 800 million such devices were sold and demand is expected to exceed 1 billion in 2013 [18]. These devices now provide advanced multi-media capability, three dimensional graphics, multiple cores, and large touch-screens [2]. Their capabilities have allowed anytime, anywhere usage of traditional desktop applications, including games. To serve the Internet demand of such devices, standards such as IEEE 802.11ad and LTE-Advanced are being developed and deployed [1,13].

Although mobile smart devices and their Internet capabilities have greatly evolved, their abilities are still outmatched by similar progress in their desktop counter-parts. Meanwhile, the rise of cloud computing has enabled computational resource consolidation in order to provide data and processing outsourcing on a massive scale [4]. The effect is to provide economies of scale in terms of computation and cost to unrelated users. Cloud technology amenities are provided through virtualization services. These services include Infrastructure, Platform, and Software as a Service. They provide computational infrastructure (IaaS), middleware and runtime environments (PaaS), or application and data hosting (SaaS).

Users need terminal devices (thin clients) to access cloud services. The ubiquity of mobile smart devices allows widespread cloud access to a large portion of the population. Overall, this has led to users likely carrying mobile touch-screen computers with persistent cloud connection, functioning as a phone, Web browser, gaming device, etc. According to [14], a significant portion of device applications consists of games. For instance, Apple's AppStore has 18% mobile games. Additionally, gaming users are shifting towards mobile devices and away from traditional consoles. In summary, the groundwork has been laid to provide a large user base for cloud services in order to deliver game content anytime, anywhere, on mobile devices without being limited by the device's capabilities or game requirements.

1.2 Video Gaming

Single-player video games started spreading with personal computer adoption in the 1980s [10]. With the growth of the Internet and computer networking technologies, video games increasingly incorporated multiplayer features. As mobile devices evolved to become more useful, mobile games also came into existence. Recently, the growth of cloud computing has accelerated the rise of GaaS. Video games are an important and growing segment in the entertainment industry. In April 2008, the Grand Theft Auto IV video game earned $310 million in 24 hours globally (more than the highest earning movie at $60 million and book at $220 million combined) [20]. The video game industry is predicted to reach $82 billion in size by 2017 [5], and mobile gaming presently accounts for $9 billion [19]. Even in a depressed economy, the industry grew 16% [16], while the largest game publisher Electronic Arts netted 40% of its revenue from online content [7]. This indicates that not only are video games a significant market force, but also that they are heavily involved in online infrastructure.

The major uses of gaming related cloud technologies include social gaming, massive multiplayer online role playing games (MMORPGs) and auxiliary services. Social games are largely based on social interactions between users, especially in the context of a social-enabling platforms (e.g., Facebook). These games involve communication and large-scale user participation, and may include aspects such as cooperation and competition. They may also include performance tracking as well as virtual incentives (e.g., achievements/trophies). MMORPGs are online-hosted video games with a massive user base. The major example is War of Warcraft, which has peaked at 1 million concurrent users from a potential base of 10 million subscribers [6]. The scale of these games requires major computational power to coordinate user interaction and its results but still relies on user devices to render graphics and provide much computation. Auxiliary cloud services (such as Steam and XBOX Live) provide IaaS level cloud services such as limited storage and networking infrastructure. They are only meant to augment the functionalities of user devices with networking, storage, and server access.

2 Gaming as a Service

GaaS can be condensed as *any-device* gaming. This relates to the ubiquity of graphical screen devices present in a user's vicinity. As discussed earlier, users maintain mobile smart devices which are growing in power and functionality, able to utilize the Internet and execute desktop applications, including video games. The greatest shortcomings of these devices however are graphics processing and battery life. Efficient graphics processing requires dedicated GPUs and puts a heavy drain on battery. However, with the shift to mobile gaming, users demand the quality of dedicated GPU gaming with the convenience of mobility. Pre-cloud approaches included creating lower-quality games or constructing desktop-replacement laptops. GaaS aims to provide a solution for high-quality games on *any* device. Utilizing the cloud to provide on-demand games to users can be thought of as a form of SaaS. From an administrative point of view, this is more akin to Platform as a Service, with the game's application data fulfilling the SaaS designation. In general, PaaS requires a cloud infrastructure and a middleware for executing application code. In the case of GaaS, the specific middleware may be vendor-dependent. However, the infrastructure will likely emphasize GPU-centric capabilities. GaaS generally requires broadband access for speed, compression techniques for bandwidth, and encryption for access control. Immediate benefits to the user include re-use of existing hardware, no game downloads, and no software maintenance.

A general model of GaaS can be described in terms of three components: the client, the interface, and the renderer:

– The client consists of the user and their thin client. The user is a consumer who wishes to play a game on their device (thin client). The thin client allows the user to access the GaaS interface and receive audio/video from the renderer. It also provides the user with I/O capability as well as Internet access.

- The interface is a mechanism that authorizes users and receives commands from them. An example would be an application that runs on the user's thin client. Through the interface, the user can login to the service, select games to play, or send input to be processed. The interface also separately controls the renderer.
- The renderer is the processing center of the GaaS service. It interprets user commands, creates frames of game data, and transmits them to the user.

In summary, GaaS provides GPU-centric SaaS which allows the users of smart devices to demand any game anywhere anytime. A major benefit of GaaS is that it allows client outsourcing of complex graphical calculations from a consumer device to the cloud infrastructure. This reduces the user-side computation and, as a result, the total energy required to execute game-related code on a client device. Further energy savings are realized as additional users share GaaS cloud GPUs where virtualization technology (e.g., NVIDIA VGX Hypervisor[1]) can maximize user density, resulting in more effective utilization of GPU cycles. To illustrate, a single GaaS-capable NVIDIA VGX K2 unit requires 38 watts per cloud user[2] whereas a comparable single-user NVIDIA GTX 690 consumer unit requires 300 watts to operate[3]. In this case, GaaS can reduce the overall graphics power consumption by 87%. In addition to power saving, users gain all the benefits of traditional cloud services and save the time necessary to contend with offline games.

2.1 GaaS Providers

GamingAnywhere is the first open-source GaaS development platform[4]. It allows researchers to test ideas on a GaaS testbed, GaaS providers to develop services on it and users to construct their own personal gaming clouds from desktops. CloudUnion[5] and OnLive[6] are GaaS companies that provide interfaces through downloadable applications and incorporate social aspects into the environment. Gaikai[7] delivers GaaS through Web browsers and needs no application download. G-cluster[8], the pioneer of GaaS, specializes in delivering GaaS over IPTV and mobile phone networks.

While it does not offer GaaS directly, NVIDIA makes the hardware necessary to enable GaaS. It offers a repackaged version of the traditional GPU-centric IaaS to GaaS companies. It has developed virtual GPU technology that allows 36 concurrent gaming streams on a single server. This service reduces the barrier

[1] `http://www.nvidia.com/object/cloud-gaming-benefits.html`
[2] `http://www.nvidia.com/object/grid-boards.html`
[3] `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-690/specifications`
[4] `http://www.gaminganywhere.org`
[5] `http://www.cloudunion.cn`
[6] `http://www.onlive.com`
[7] `http://www.gaikai.com`
[8] `http://www.g-cluster.com`

to entry for new GaaS companies that would otherwise be incapable of acquiring such specialized hardware.

2.2 GaaS Issues

The main concerns of GaaS include user responsiveness, stream quality, service quality, and operating cost:

- *User responsiveness* relates to the visible lag of the delivered video frames or audio as well as the perceived latency in the response to user commands. This is caused by network latency and processing delay. Network latency is the result of the user's access network. It can be addressed through gaining access to higher quality connections or deploying more cloud locations by the service provider. Audio/Video lag can be reduced through the use of better GPU systems although it may be related to network quality as well. User responsiveness is especially crucial because many video games depend on extremely quick action performances from the user and may be the single greatest barrier to widespread replacement of gaming devices with GaaS. However, this latency requirement may not exist for all game types (e.g., turn-based games).
- *Video quality* is a balance of many factors. It is limited by the graphical resolution of the user's device, network bandwidth, and load on the cloud GPUs. Bandwidth can be addressed through the use of compression algorithms. A recent example is when Netflix began utilizing a codec optimization technology to reduce its video streaming bandwidth by 50% [15]. Optimizations such as this and the development of new standards will allow higher quality video with fewer resources for GaaS service. Video quality is important because it provides direct data of the game's current state to the user and serves as a major selling point for GaaS.
- *Service quality* is the overall quality of the GaaS service. This includes game selection space, storage capacity, processing speed, and interface. Game selection is partly non-technical although it relates to the compatibility of the middleware with games originally developed for non-cloud platforms. Storage involves the saved game data of users as well as their settings and paused game sessions. Processing speed involves the loading/buffering times of playing a cloud game. Interface involves the attractiveness and ease of use of the service as well as its compatibility with the controls of the user's mobile thin client. Service quality is important because it determines the level of participation in a particular GaaS service.
- *Operating cost* is a combination of equipment cost and operational costs. Equipment must be maintained and replaced when outdated. Operating cost (such as power and bandwidth) can be optimized through the selection of the cloud's physical location but is a trade-off between low billing rates and user responsiveness. Operating cost is important because it determines the profitability of a GaaS company and the level of any charges users may face.

2.3 Related Legal Issues

Many legal issues arise from the use of GaaS [11]. As more interest in GaaS becomes generated, patents may be formed specifically for this domain, resulting in restrictions in evolution and growth. Ownership also becomes a concern. Since game code resides and is rendered on the cloud, the concept of owning a game may give way to purchasing access to a game for a set amount of time. Also of importance are guaranteed service levels and pricing schemes. Other issues include the requirement of a large number of users to participate in the service, not only to provide multiplayer potential but also to offset the large costs associated with providing a GaaS service. On a beneficial note, piracy and hacking may be missing in GaaS services since no code is hosted on the user's device; only streamed video of the game is delivered.

3 Current GaaS Research

The field of GaaS is currently the focus of significant research. We illustrate that research through some representative recent work. In [8], the authors present a GaaS system that permits the simultaneous support of multiple devices of varying resolutions from varying focal points. This is accomplished through the modularization of the cloud structure into service (DSP), rendering (DRS), and streaming (EQS) systems. The DSP receives user input and interacts with the DRS to model a scene. The EQS then renders the image and streams to the client device. Multiple views are achieved through iteratively creating cameras and varying viewports and binding them to render windows. Performance is maintained through the use of per-thread task queues, highly optimized schedulers, and shared GPU resources. In addition, multiple views are packed as a continuous stream of traffic using the H.264 codec. Tests achieved 15 fps for 8 simultaneous streams on a single GPU.

In [17], the authors present a framework that provides a virtual interface for remote hardware. In this sense, it is a form of middleware for GaaS. It is composed of three components: a Kusanagi plug-in, lobby server, and MPEG-4 client. The plug-in provides interactivity from the client to server and video from server to client. The lobby server acts as both a portal and resource manager. The client is composed of a full multimedia player with networking and interactivity. Tests indicated that the system was appropriately adjusting video bitrate to respond to network congestion and that 92% of test users felt positively about the system.

In [3], the authors examined the suitability of the current network infrastructure in the United States in providing GaaS. The authors measured network conditions of 2500 users of the Amazon EC2 service using BitTorrent and geographical location filtering. They found that the median latency to less than 70% of them was 80 ms or better which they argue is the maximum permissible latency for an acceptable GaaS experience. In addition, 10% of users would be incapable of properly utilizing GaaS due to extreme latency. The authors then evaluate the best way to reduce the overall latency for the country by comparing latency vs. region based data center creation strategies. They found that a prohibitive number of additional centers are required to achieve 90% population coverage. An additional strategy of deploying GPU-equipped CDN edge servers results in a 28% increase of user coverage. Overall, the authors conclude that the existing US cloud infrastructure is currently not suitable for providing GaaS.

In [12], the authors examined the evolution of the software business model from traditional packaged format to SaaS and found that it has forced developers to transform into low-margin, high user formats. Specifically, they perform a case study on G-cluster, the pioneer of GaaS. They examined G-cluster in 2005 and found that it reached 15,000 households by running game source-code modified for their infrastructure. This service was then offered to middle-men who, in turn, sold it to consumers. By 2010, G-cluster had evolved to provide full-fledged GaaS to network operators of IPTV totaling 3 million households. They also released a SDK for game developers wishing to submit their own compatible games. G-cluster found that the main audience for GaaS was moderate users who were passionate enough to pay for games but casual enough to not want to spend money on their own GPU devices. Overall, the authors recommend that GaaS companies provide a minimum of PaaS level in order to remain competitive.

4 Conclusion

GaaS is an emerging cloud-based computing paradigm that is enabling mobile gaming. It is the result of consumer demand for anytime, anywhere gaming on low-power, Internet equipped mobile devices. GaaS is game-specific SaaS that utilizes GPU centric hardware. The game is processed and rendered in the cloud and streamed to the user's device. This allows the use of low-end, mobile devices to "play" high-end games. It also reduces the cost and time necessary to play games by distributing these over the entire user base. This is a relatively young field and many developments are being produced as more attention is given to this emerging service. In the future, GaaS will likely become an important facet of the cloud computing portfolio and may lead to related GPU services such as on-demand visualization or encryption cracking.

References

1. WiGig Alliance. Defining the Future of Multi-Gigabit Wireless Communications (2011), http://wirelessgigabitalliance.org/?getfile=2132
2. ASUS. Eee Pad Transformer Prime TF201, https://www.asus.com/Tablets_Mobile/Eee_Pad_Transformer_Prime_TF201
3. Choy, S., Wong, B., Simon, G., Rosenberg, C.: The Brewing Storm in Cloud Gaming: A Measurement Study on Cloud to End-user Latency. In: 2012 11th Annual Workshop on Network and Systems Support for Games (NetGames), pp. 1–6 (2012)
4. Cohen, R.: The Cloud Hits the Mainstream: More Than Half of U.S. Businesses Now Use Cloud Computing. Technical report, Forbes (2013)

5. Gaudiosi, J.: New Reports Forecast Global Video Game Industry Will Reach $82 Billion by 2017. Technical report, Forbes LLC (2012)
6. Grubb, J.: World of Warcraft: Mists of Pandaria Sees 1 Million Concurrent Players in China. Technical report, VentureBeat (2012)
7. Hing, D.: EA Admits Disappointment With Warfighter Sales. Technical report, Bit-Tech (2012)
8. Kim, S.-S., Cho, C.: Multiscreen-based Gaming Services using Multi-view Rendering with Different Resolutions. In: Proc. of the Second International Conference on Mobile Services, Resources, and Users (MOBILITY 2012), pp. 34–37 (2012)
9. Moore, G.E.: Cramming More Components onto Integrated Circuits. Electronics Magazine, 38(8) (1965)
10. Moreno, C., Tizon, N., Preda, M.: Mobile Cloud Convergence in GaaS: A Business Model Proposition. In: 2012 45th Hawaii International Conference on System Science (HICSS), pp. 1344–1352 (2012)
11. Nicholson, J., Leavitt, J.: To the Cloud! Anticipating the Legal Issues in Cloud-Based Gaming. Technical report (2011)
12. Ojala, A., Tyrvainen, P.: Developing Cloud Business Models: A Case Study on Cloud Gaming. IEEE Software 28(4), 42–47 (2011)
13. Parkvall, S., Dahlman, E., Furuskar, A., Jading, Y., Olsson, M., Wanstedt, S., Zangi, K.: LTE-Advanced - Evolving LTE towards IMT-Advanced. In: IEEE 68th Vehicular Technology Conference, VTC 2008-Fall, pp. 1–5 (2008)
14. Robinson, J.: The Evolution of the Gaming Industry into the Pockets of the Consumer. Technical report, Univ. of Southampton, UK (2012)
15. Roettgers, J.: EyeIO: Netflixs Secret Weapon Against Bandwidth Caps? Technical report, GigaOM (2012)
16. Takahashi, D.: Video Game Industry to Hit $70 Billion by 2015, But Growth Will Slow. Technical report, VentureBeat (2010)
17. Tizon, N., Moreno, C., Cernea, M., Preda, M.: MPEG-4-based Adaptive Remote Rendering for Video Games. In: Proceedings of the 16th International Conference on 3D Web Technology, Web3D 2011, pp. 45–50. ACM, New York (2011)
18. van der Meulen, R.: Gartner says 821 million smart devices will be purchased worldwide in 2012; sales to rise to 1.2 billion in 2013. Technical report, Gartner (2012)
19. Watson, P.: Mobile Gaming Market Grows to $9 Billion as OBJE Readies Debut Gaming App. Technical report, BusinessWire (2013)
20. Mark Whiting. GTA4 Launch Sales Break Two World Records (2008), `http://www.1up.com/news/gta4-launch-sales-break-world`
21. Zhang, G.-Q., Zhang, G.-Q., Yang, Q.-F., Cheng, S.-Q., Zhou, T.: Evolution of the Internet and its Cores. New Journal of Physics 10 (2008)

Circlebook: Visual Display of Friend Proximity

Gianni Fenu and Lucio Davide Spano

Department of Computer Science, University of Cagliari,
Via Ospedale 72 Cagliari, Italy
{fenu,davide.spano}@unica.it

Abstract. In this paper we introduce Circlebook, a novel technique for visualizing an ego-network according to a distance function, using a radial layout. We apply such technique in order to support social network users (e.g. Facebook) in inspecting the level of interaction with their friends. In addition, we propose a set of control techniques that exploit this visualization, such as the filtering of contents created by the user's friends, and the end-user editing of the distance values. In addition, we detail the implementation of a prototype for both the visualization and the filtering techniques in a mobile setting. We evaluated the usability of the proposed approach through a user study, comparing our prototype with the current Facebook wall-post visualization. The experimental results shows that the user are immediately proficient with the visualization and that it can be successfully exploited for controlling the content filtering.

Keywords: Radial layout, mobile user interface, social networks, content filtering, ego network, visualization.

1 Introduction

The wide availability of internet-connected mobile devices is pushing the market of social network applications, since users are engaged in creating new contents in a ubiquitous manner. The number of users that produce and consume contents through a mobile device in a social network has officially overtaken the number of desktop users, as has been officially reported in [3].

In a similar fashion, the number of active users is increasing and, considering that each user has about 100 friends on average [11], this means that users are already overwhelmed by the content created by their friends.

However, in [4] the authors demonstrated that Facebook users actually interact with a small subset of their declared friends. Therefore, the social network users may usually skim many contents they are not interested in even in their wall or homepage.

In order to ease the management of the different friend subsets, social networks like Facebook, Google Plus and Twitter provide a "friends list" mechanism, which enables grouping and differentiating friends. The user can usually select one of the different groups directly in their homepage, filtering the contents that were created by friend not belonging to the current group.

Unfortunately, they are considered tedious to create and maintain when the network grows, and the situation is even worst if we consider a mobile setting.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 129–142, 2013.

Different solutions for automatically creating such subsets are provided by research on graph analysis. For instance in [6], the authors demonstrate that it is possible to identify user's social circles exploiting both the network structure information and the user profile information. This solution has the advantage of discharging the user from subset creation task but, from the user's point of view, it is not possible to understand which criteria are used by the system in order to create them.

Therefore, it is important to provide the users with means for:

1. identifying and controlling the subset of closer friends, according to the number of interaction that occurred between the user and her friends
2. understanding the relation between the subset, which can be created also with the support of the underling application, and the contents that are shown or filtered in the application interface. This point is particularly relevant in mobile settings, where it is crucial to provide meaningful information to the user.

In this paper, we propose a radial layout visualization for friends in a social network, which we demonstrate to be particularly effective in mobile settings. Such visualization exploits a distance concept, which can be exploited by the system for content filtering and that can be interpreted by the user as the criteria for selecting contents. The user can exploit the same visualization for analyzing the system's internal representation of her social network (inspection). In addition, the same visualization allow her to change such representation (control), communicating that it is different from what was expected by the user.

The paper is organized as follows: after a brief discussion of the related work, we present the visualization and its features, discussing an implementation prototype. After that, we discuss the results of a user study, showing that users are immediately proficient using such visualization and that it improves current mobile social-network interfaces. Finally, we go into conclusions and future work.

2 Related Work

The exploitation of radial layouts for representing big amounts of data has a long history, surveyed in [2]. The authors categorized the different layouts and visualization design patterns (polar plot, space filling and ring). According to this classification, our work uses a star pattern, included in the polar plot category, which has been proved to be useful for a wide variety of applications, from the display of queries [6] to networks [14] etc.

In a mobile setting, a radial visualization is efficient in space usage, especially for browsing and interacting with trees and hierarchical structures. For instance, in [5] the authors propose an edgeless visualization of a tree using a 2D space filling technique, while in [11] is discussed an expandable-table interactive visualization for hierarchical structures.

In this paper, we apply this layout to the visualization of an ego-network, which is the network that includes a user and her friends, enabling the inspection and control for the filtering of contents created by friends.

Some work has been done in the area of the visualization of communication patterns among mobile users of social networks. For instance, in [8] the authors demonstrated the benefits of an inspection of the different social interactions mediated by the mobile phone in a daily basis. Besides the representation of expected interaction trends, the users were able also to identify interaction gaps between them and some other friends. We created a visualization that is able to summarize the different interactions through the "lifetime" of the user in Facebook.

In the field of social recommender systems, such kind of transparent exposition of the mechanism for selecting the proposed contents increases the perceived recommendation quality and the overall system satisfaction [7]. We apply the same principles in the context of content filtering.

3 Friend Visualization

In this section, we discuss the visualization technique exploited in Circlebook. The visualization is based on a modified radial layout [12], where graph nodes are positioned using a set of concentric circles, each one representing an hop between a given node and the one that appears at the center of the visualization. Such kind of layout does not consider edge weights: the position of a node is calculated according to its depth with respect to the central node.

We took inspiration from such visualization since it has two main advantages: the first is the possibility to focus on a central node, which in this case is the current user. The second one is the possibility to provide an immediate feeling of the distance through the concentric circles in the background.

Differently from [12], in our case we cannot use the depth as parameter for positioning the nodes on the various circles. Indeed, from a social graph point of view, all nodes in a user's friend list are directly connected with her. Instead, we need to consider some weight on the edges.

Such weights depend on two different factors. The first one is directly related to type of network we are modeling and, consequently, to the concept of distance we consider. Here we consider a social network and we aim to model the "interaction distance", which means that we want to find the friends that exchange information with the current user, separating them from the ones that are included in the fried's list, but do not communicate with our user often. It is worth pointing out that the distance concept can be applied also to networks that are different from the one considered in this paper (e.g. article topics, videos etc.).

The second factor purpose is twofold. The first one is to control the graph layout, distributing the different nodes on the radial representation, while the second one is to implement a discretization step from the continuous value of the distance to the discrete values of the distance levels, which are represented as circles.

The two factors are implemented by two different functions: one for obtaining an interaction distance between the user and a given friend, and one for laying-out the graph. We detail how we selected such functions in the next section.

The general idea of the visualization is positioning each one the user's friends in one of the concentric circles according to the edge weight. This means that the visualization do not show the exact distance value, but a friend is positioned on a given circle if the distance value is contained in a range, which depends on the number of distance levels we consider.

Considering such kind of layout for visualizing the interaction level between the user and her friends, we have evidence from the study in [4] that people communicate regularly with a small subset of friends. This grounds the main assumption for the effectiveness of the circular layout in this case: from an interaction point of view, the number of close friends is much lower than those far. The latter concept can be applied to the layout, assuming that on average the inner circles are less populated than the outer ones, justifying the usage of circles with a lower perimeter for the closest friends and with a higher perimeter for the others.

The resulting layout for friend visualization is depicted in Figure 1. Each square icon represents a friend of the user that is in the center of the visualization. The positioning of an icon in the different circles depends on the two aforementioned functions. As it is possible to see in Figure 1, the inner circles are less crowed than the outer ones. We used 10 levels of distance between the user and her friends.

Our visualization can be mapped on the one proposed in [12], which exploited the node's depth for selecting the circle, simply adding a number of fake nodes between the considered user and a given friend corresponding to the distance level (minus one). This mapping is not practical from an implementation point of view, since in the worst case it may raise the number of nodes by a factor of ten.

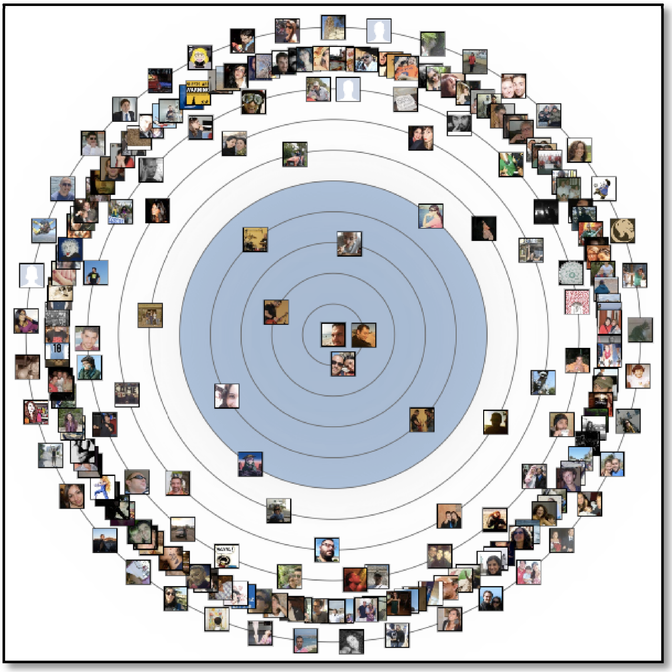


Fig. 1. Circlebook visualization

3.1 Distance and Layout Functions

As we already mentioned in the previous section, the circular friend visualization is based on the composition of two functions: a distance function and a layout function.

The first one calculates a continuous distance between the central node (the Facebook user in our case) and the directly connected ones.

In this paper, we consider as case study the Facebook social graph. We defined a distance function that is based on the number of interactions between the user and each one of her friends. In particular, for each friend of our user, we considered as interactions the following events:

1. The friend comments one of the user's posts on her wall
2. The user comments one of the friend's posts on her wall
3. The friend likes one of the user's post on her wall
4. The user likes one of the friend's post on her wall

In our prototype implementation, for each user we considered (see the evaluation section), we had to count all such events in order to have the total number of interactions, accessing the user's Facebook data through the Open Graph Api [2]. Therefore, the number of interaction considered in this paper for each friend is simply the sum of previously listed events.

After this counting step, we normalized the distance value by the maximum value of interactions with a single friend. Such sum gives us a value between 0 and 1 that is higher for friends that communicate with our user very often, and lower for the others. The actual value of our distance function has been calculated as shown in equation 1, where $count(u, f)$ represents the interaction count between the user u and the friend f and $max(u, F)$ the highest number of interactions of the considered user with any one of her friends (represented by the set F).

$$distance(u, f) = \left(1 - \frac{count(u, f)}{\max(u, F)}\right) \cdot 100 \tag{1}$$

We do not consider private messages in the distance calculation. This has a practical rather than theoretical motivation: the real Facebook users we contacted for the testing did not allowed our application (which was built on top of the Facebook Open Graph Api [2]) to access the private messages for obvious privacy issues.

It is worth pointing out that it is possible to use more precise definitions for estimating the distance (such as the edge centrality [7]) but, as we discuss more in detail in the evaluation section, this measure is already satisfactory for the users. A study on the best candidate for this function is beyond the scope of this paper.

The second function we use in the visualization implements the discretization step for grouping the users in different circles, given a number of distance levels. The only assumption is having values normalized in an interval between 0 and 100, where 0 corresponds to the minimum distance from the center and 100 to the maximum. As already mentioned in the previous section, the function can be selected according to the distance function distribution.

In our case, we empirically established that for friend visualization a quadratic function distributes the nodes fairly among the levels, while a linear function leaves too many empty circles. A resulting distribution example is shown in Figure 1, using ten distance levels.

3.2 Visualization Features

The visualization shown in Figure 1 provides the user with an overview of his friend distribution. However, considering the limited screen space in mobile devices, it does not allow to recognize immediately all friends, in particular the ones that are located in "crowded" circles.

In order to mitigate this problem, the user can zoom and pan the view. The zoom function, which is associated to the usual pinch gesture, enables the exploitation of different levels of detail, showing the friend's name below his icon.

The resulting visualizations are shown in Figure 2: the left part shows a zoomed-in view, where the friends' icons have also a label with the friend's name (different level of detail). The right part shows the possibility to pan the radial visualization, positioning the viewport in a point different from the center.

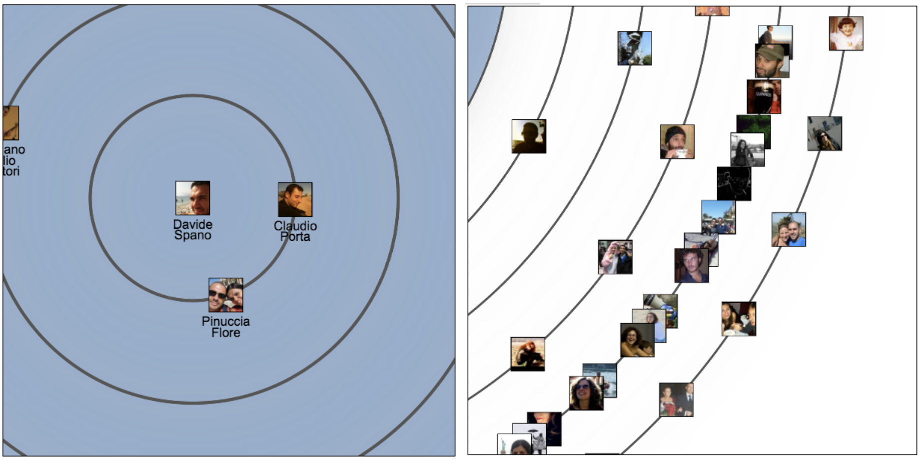


Fig. 2. Interaction with the Circlebook visualization. Zoom in (left part) and pan (right part).

In addition to these classic features, the visualization is provided also with a functionality for highlighting only one among the friends. This enables the user to immediately find a particular node on the graph, and it can be used for implementing an interactive search by node in the visualization, as discussed in the next section.

Figure 3 shows how the Circlebook visualization highlights one of specific friend: her icon and the one representing the user are completely opaque, while all the icons representing the other friends are shown in transparency. In this way it is possible to locate immediately the only icon which is opaque in a position different from the center of the circles.

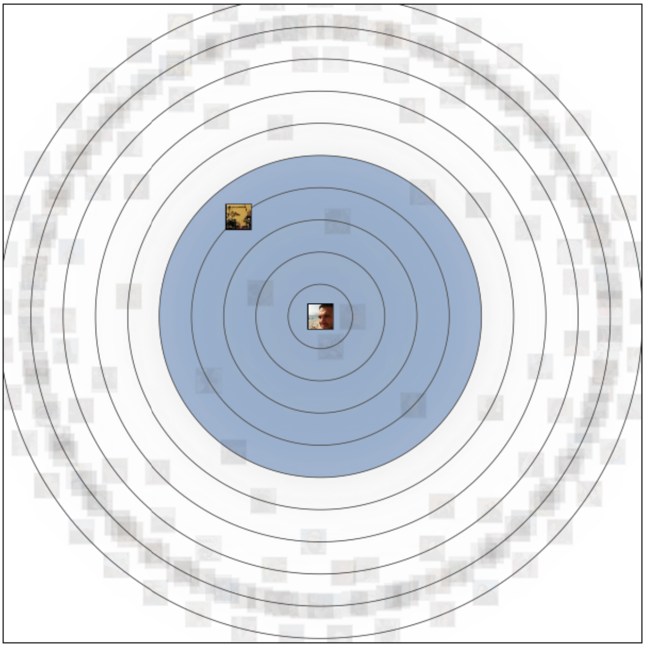


Fig. 3. Circlebook friend highlighting

4 Application to Content Filtering

In this section, we describe an application prototype that exploits the proposed visualization for controlling a content filter through the inspection of the interaction levels. The application is an enhanced Facebook wall-posts visualizer, which enables the filtering of content produced by friends considered distant by the user. The visualization is shown in Figure 4.

In addition to the normal post visualization in chronological order (the rounded rectangles with “Like” and “Comment” links), the application shows the position of intermediate filtered contents with a scissor button, which indicates also the number of filtered posts. It is possible to see two of these buttons in the left part Figure 4, one above the post (one content filtered) and one below (7 posts filtered).

Such buttons can be pressed by the user, in order to show the contents that have been filtered by the application, which are shown using transparencies for differentiating it from the other contents.

The user can inspect why some of the contents have been filtered by the application pressing the gear button on the top-left part of the title bar. Once the user presses this button, the application shows the presentation in Figure 5, which exploits the visualization proposed in this paper.

The content created by the friends included in the sky-blue circle are shown in the Figure 4 presentation, while the ones that are outside are filtered.

Fig. 4. Circlebook wall-post visualization

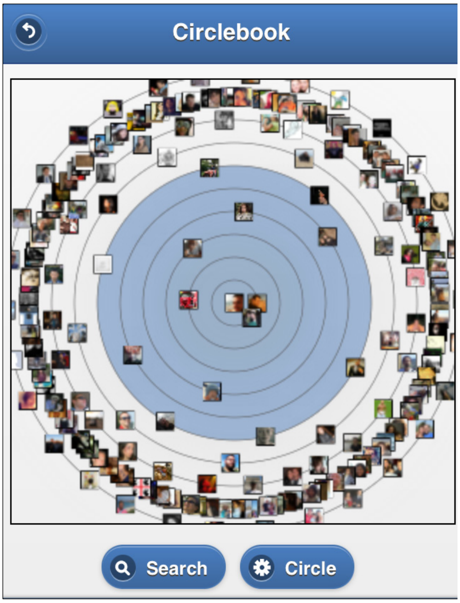


Fig. 5. Inspecting friend distances in Circlebook

The user is able to control the dimension of the circle pressing the Circle button in Figure 5. After that, the application shows the presentation in and changing the value of a slider, shown in Figure 6: in the left part the slider has an higher value and the light blue circle has a bigger diameter, while in the right part the slider and the circle diameter are smaller. After changing the filtering circle dimension, the effects are visible on the post presentation (Figure 4): if it has been increased, there are less filtered wall posts and vice-versa.

In addition to the control on the level of distance exploited by the filter, the user can also inspect and control the distance between him and a particular friend. Indeed, it may happen that the user is not happy with the distance calculated automatically by the application support and she wants to modify it.

Supporting such feature implies that:

1. the user has the possibility to inspect the view and find the position of a particular friend
2. the user has the possibility to change the friend's position if she think that the distance calculated by the application is different from the real one.

The inspection part is supported through a search by name feature: the user presses the Search button in Figure 5. Then, she selects the friend's name through a search filter bar.

At this point, the application highlights the selected friend, showing the presentation in Figure 7 and increasing the value of alpha value for the icons that do not represent the selected friend. In this way, the user can immediately locate the selected friend and see interaction distance.

Fig. 6. Changing the filtering distance level

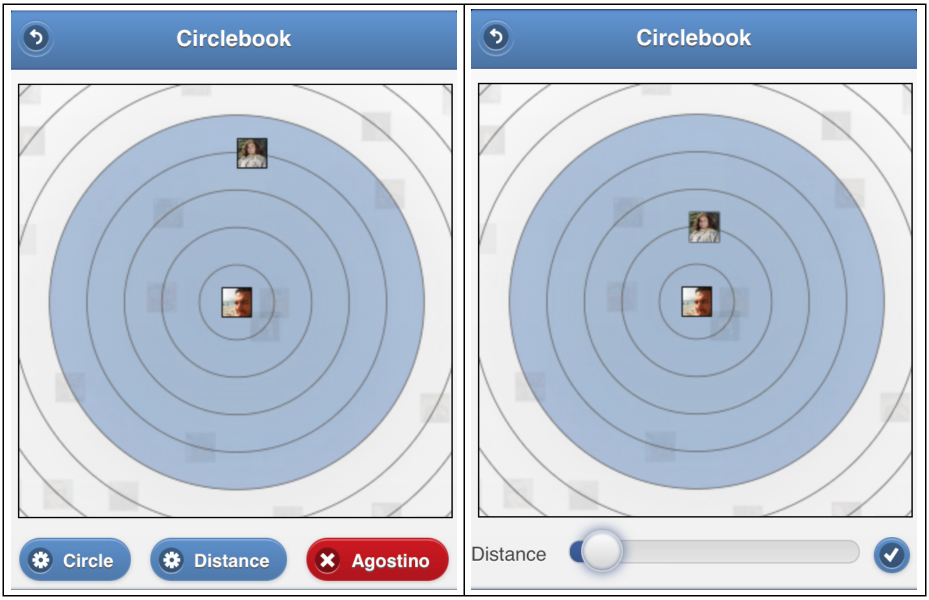


Fig. 7. Changing a friend's distance

The user may now decide to change the value of this distance, which can be done pressing the *Distance* button in Figure 7, left part and moving the selected friend's icon from one circle to another through a slider, in the same way she can modify the dimension of the sky-blue circle in Figure 6. The result of changing such value is shown in Figure 7, right part: the user has decreased the value of the slider and the icon of the selected friend is now closer to the center. It is possible to de-select the highlighted friend simply pressing the red button in Figure 7.

In summary, the application provides the user with means for inspecting the distance at a high level and also for particular friends. The filter control is based on a general cut-off rule based on a distance level completely customizable for the user. In addition she can also modify the value of the distance representation provided by the system for particular friends, offering also a fine-grained control on each one of them.

5 Evaluation

In order to evaluate the usability of the friend visualization in the proposed setting, we performed a user study. The aim of the test is twofold. On the one hand, we want to establish whether the users prefer the current visualization of the wall posts (considered as the baseline) or the enhanced one proposed in this paper.

We recall here that Facebook allows filtering contents created by a certain friend in the wall post, in both its desktop and mobile versions. In addition, it exploits a ranking algorithm for contents, which visualizes first the wall-posts that are considered more important, weighting the affinity between the user and the friend that created the content. In addition, it considers also the content type and a time-decay function [5]. The user can select to visualize the wall posts in chronological order or according to their importance. We compare the two approaches in order to understand if providing means for inspecting and controlling the contents have a positive impact on the user's experience.

On the other hand, we want to demonstrate that such inspection and control features are supported adequately by the visualization we propose in this paper, measuring the effort required for completing the filter inspection and control tasks we defined for the test.

5.1 Test Organization

The test consisted of five parts. In the first one, the users read a small introduction about the application they were going to evaluate and about which kind of personal data was required. After that, they filled a small demographic questionnaire.

The second part was dedicated to an evaluation of the current filtering support for the Facebook wall-posts. In particular, we recalled the possibility to filter contents that were created by a specific friend, and the possibility of showing only the contents created by a given friend list, as we discussed in the previous section. Obviously, for participating to the experiment, the user should be a Facebook user and, as we better detail in the next section, we selected user that were experienced with this social network.

This evaluation was performed filling the Software Usability Software Usability Scale (SUS) test [1], which is considered one of the most effective post-study questionnaire in literature [10].

We did not ask explicitly to use the current Facebook interface before filling this test, since all the user were already familiar with it, but it was available for re-experimenting its features if needed.

In the third part, the user had to allow our application using their Facebook data in order to calculate the distances as described in section 3. This means that each user authenticated with her Facebook credential and the following parts of the test were performed exploiting the user's real data and friend list. Consequently, the calculated distances and the friend visualization were based on the current user's Facebook real interactions.

The fourth part was dedicated to performing a set of tasks with Circlebook. The tasks were performed through different smartphones (different versions of the Apple iPhone, Samsung Galaxy S2 and S3 etc.) since the prototype is a web based application and the user accessed it through their own personal device. The only requirement was having a screen resolution of at least 320 x 400 pixels.

In particular, the task to accomplish were four:

1. Explore the post and the radial friend visualization, establishing who are the closest friends (inspection task)
2. Modify the dimension of the filtering distance level, in order to include the ones that they want to follow (control task)
3. Find a friend in the circular visualization (inspection task)
4. Modify the distance of a friend that is supposed to be closer or more distant (control task)

At the end of each task, we wanted to measure the task performance effort that was perceived by the users. In order to do that, we asked each user to answer the the well-known Subjective Mental Effort Question (SMEQ) [13].

The last part of the test was dedicated to filling again the SUS questionnaire, but this time the users evaluated the overall usability of Circlebook.

5.2 Test Results

Nineteen users participated to the study, 11 male and 8 female, aged between 21 and 43 ($\bar{x} = 34.5,\ SD = 5.28$), 6 have a high-school degree, 4 a bachelor, 7 a master and 2 a PhD. One of them has been using Facebook since three years, 10 have been using it since 4 years, while 8 since five or more years. Eighteen participants use a mobile device daily, while only one uses it weekly. The users were all experienced in both using Facebook and using mobile devices.

In order to establish the best content filtering support, we performed a within-subject comparison of the SUS scores for the current Facebook visualization and the one proposed in this paper, in order to establish if there is a statistically relevant difference between the two designs.

In this case, it is impossible to alternate the starting condition for mitigating the carryover effect. Indeed, all testers needed to be Facebook users and then they used its visualization in advance.

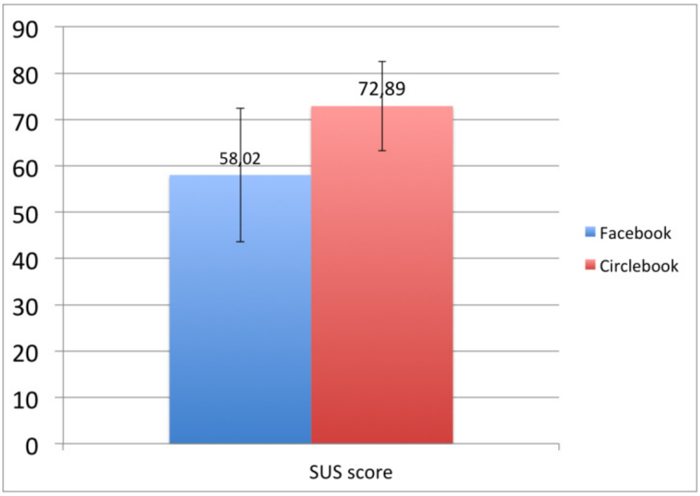


Fig. 8. Facebook and Circlebook SUS score comparison. The caps shows the standard deviation of the respective scores.

The analysis of the SUS rating is shown in Figure 8. The current Facebook visualization SUS rating in a [0,100] interval is 58.02, SD 14.47, while Circlebook was rated 72.89, SD 9.62. The score difference is statistically relevant: we performed a paired t-test and we obtained 0.0019 as p-value. Therefore, we are 95% confident that the overall usability score difference is between 8.62 and 21.12 points. considering such numerical results and according to [9], we can affirm that the perceived usability of Circlebook is higher. This confirms that the work discussed in this paper is well-grounded, since the features implemented through the friend visualization help the user in understanding *why* a content is there or not and, in addition, to modify the system's behavior if she is not completely satisfied.

After having demonstrated that the visualization is useful for filtering social network contents, we have to establish whether the effort required for manipulating the proposed friend visualization is adequate or not. In order to do this, we analyze the answers to the SMEQ [13] question, which were given at the end of each performed task. This allowed us to establish a confidence interval for a quantitative measure of the user's performance effort. In addition, it is also possible to provide a label for such quantitative measure according to [13], in order to have a natural language explanation of the effort level. Examples of such labels are "not very hard to do" or "very, very hard to do".

The post task test scores are in a [0,150] interval, where 0 is the minimum difficulty value, while 150 is the maximum. The test results are the following 1) $\bar{x} = 8{,}33\ \ SD = 8.51,$ 2) $\bar{x} = 5.27,\ SD = 6.23,$ 3) $\bar{x} = 5.27,\ SD = 6.23,$ 4) $\bar{x} = 16.67,$ $SD = 27.78$ and they are summarized in Figure 9.

Given such results, we can calculate for each task the upper bounds for the perceived difficulty scores that are the following, with a confidence interval of 95%: 1) 13.45, 2) 7.47, 3)7.47, 4) 19.57.

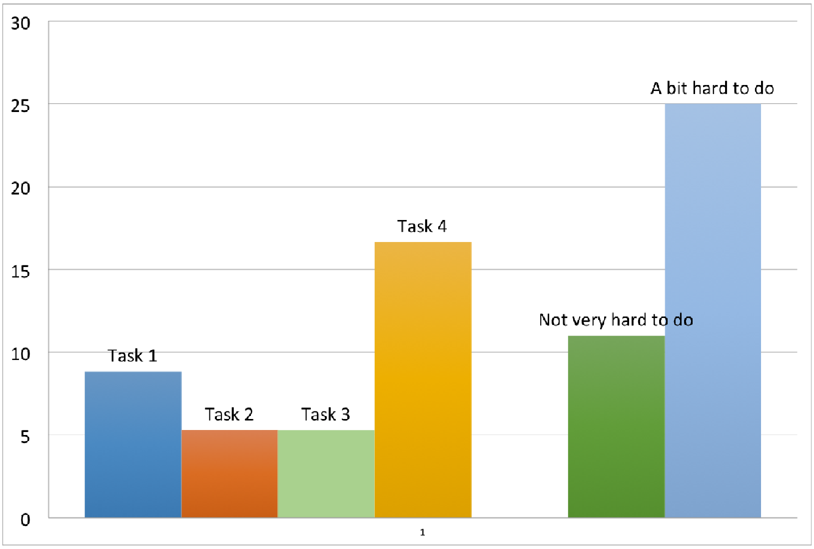


Fig. 9. SMEQ scores for the four user-study tasks (left part). The right part shows the values for the two closest score labeling in [13].

According to [13], the numerical value for a task "*not very hard to do*" is 12, while for a task "*a bit hard to do*" is 25. Therefore, the tasks 2 and 3 are easier than the not very hard to do level, while the tasks 1 and 4 are easier than a bit hard to do level.

Such experimental results confirms that users are able to understand and manipulate the proposed radial representation of their ego-network for inspecting and controlling the content filter, with an acceptable effort, according to the boundaries reported in [13].

In summary, we got evidence from the experiment that the users prefer Circlebook as a way to inspecting and controlling the content filtering in social networks with respect to the current Facebook interface. In addition, we have shown that the user did not find particular difficulties in understanding and executing the different tasks, therefore the proposed radial visualization can be considered effective for the proposed application.

6 Conclusion and Future Work

In this paper, we presented a novel visualization of the friend list that we exploited in a mobile device social network client called Circlebook.

The visualization provides a general idea of how many friends usually exchange information with the current user and it allows the user to control a content filter by both modifying the distance level for the inclusion and also the distance value between a given friend.

A user test demonstrated the proposed visualization is more usable than the one currently used in Facebook, and that both the inspection and control task have a low difficulty level.

In future work, we want to study the impact of using different distance and layout functions and to apply the visualization to recommender systems. In addition, we plan to perform a long-running testing phase, where we will ask users to exploit our Facebook client for a long amount of time, in order to evaluate the prototype usability in a “real life” setting for a day to day use of the social networks.

Acknowledgments. We gratefully acknowledge Sardinia Regional Government for the financial support (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1 “Avviso di chiamata per il finanziamento di Assegni di Ricerca”.

References

1. Brooke, J.: SUS: A “quick and dirty” usability scale. In: Jordan, P., et al. (eds.) Usability Evaluation in Industry, pp. 189–194. Taylor & Francis, London (1996)
2. Facebook: Facebook Open Graph, `https://developers.facebook.com/docs/opengraph/` (accessed: April 29, 2013)
3. Facebook: Fourth Quarter and Full Year 2012 Financial Summary, `http://investor.fb.com/releasedetail.cfm?ReleaseID=736911` (accessed: June 04, 2013)
4. Golder, S., et al.: Rhythms of Social Interaction: Messaging Within a Massive Online Network. In: Steinfield, C., et al. (eds.) Communities and Technologies 2007, pp. 41–66. Springer, London (2007)
5. Kincaid, J.: EdgeRank: The secret sauce that makes Facebook’s news feed tick. TechCrunch (April 2010)
6. McAuley, J.J., Leskovec, J.: Discovering Social Circles in Ego Networks. CoRR. abs/1210.8 (2012)
7. De Meo, P., et al.: A novel measure of edge centrality in social networks. Knowledge-Based Systems 30, 136–150 (2012)
8. Paasovaara, S., et al.: Perceptions of visualizing personal mobile communication patterns. In: Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia, pp. 14:1–14:10. ACM, New York (2010)
9. Sauro, J.: A Practical Guide to the System Usability Scale. Create Space, Denver (2011)
10. Tullis, T.S., Stetson, J.N.: A comparison of questionnaires for assessing website usability. In: Usability Professional Association Conference (2004)
11. Ugander, J., et al.: The Anatomy of the Facebook Social Graph. CoRR. abs/1111.4 (2011)
12. Yee, K.-P., et al.: Animated exploration of dynamic graphs with radial layout. In: IEEE Symposium on Information Visualization, INFOVIS 2001, pp. 43–50 (2001)
13. Zijlstra, R., Van Doorn, L.: The Construction of a Scale to Measure Subjective Effort, Delft, Netherlands (1985)

Feel the World: A Mobile Framework for Participatory Sensing

Theophilos Phokas[1], Hariton Efstathiades[2],
George Pallis[2], and Marios D. Dikaiakos[2]

[1] Department of Computer Science, University of California, Los Angeles
[2] Department of Computer Science, University of Cyprus
tphokas@gmail.com,
{cs07eh2,gpallis,mdd}@cs.ucy.ac.cy

Abstract. Nowadays, smartphones have almost replaced basic feature phones all over the world. Today's smartphones come equipped with an increasing set of embedded sensors, computational and communication resources. All these gave developers the ability to design and implement a wide variety of applications in the domains of healthcare, social networking, safety, environmental monitoring and transportation. This paper presents a novel middleware platform, called Feel the World (FTW) which provides third party programmers, with little phone programming experience, the ability to develop applications that enable people to sense, visualize and share information about the world they live in.

Keywords: Participatory sensing, mobile applications, mobile data collection.

1 Introduction

Technological advances in sensing, computation, storage and communication have brought closer than ever a global sensing device that enables a great number of novel applications that weren't available in the past. Small inexpensive sensors, low power processing but mostly the fact that today's smartphones come equipped with an increasing set of embedded sensors, computational and communication resources, gave developers the ability to design and implement a wide range of applications. Analysts estimate that 5 billion people worldwide use mobile phones[1] - more than half the world's population.

While the real numbers are debatable, it is clear that the proliferation of smartphones equipped with cutting-edge sensing technology and high-end processors opens new horizon in participatory sensing having a tremendous impact on our society. In this context, smartphones represent an ideal computing platform to develop urban sensing applications across a wide variety of domains, such as social networking, environmental monitoring, healthcare and transportation [7,10]. Therefore, a key challenge in realizing the potential of participatory

[1] http://www.physorg.com/news185467439.html

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 143–156, 2013.

sensing is to ensure the ease of developing and deploying such applications, so that the developer does not have to reinvent the wheel.

To address this challenge, we present a novel middleware platform, called Feel the World (FTW). FTW provides to third party programmers, with little phone programming experience, the ability to develop applications that enable people to sense, visualize and share information about the world they live in. FTW framework abstracts implementation details, like service and thread implementations, in order to provide a convenient way to handle sensors both embedded and external. Even though capturing sensor data is quite easy using the provided SDKs, developing a continuous sensing application that takes into account network traffic and the resources of the device (e.g., battery, processor utilization) on which the application is running is not a trivial task. Although several frameworks have been developed in order to address this challenge (e.g. MyExperience [9], Jigsaw [8], Funf [1], SociableSense [11], ODK Sensors [3]), FTW differs from the existing platforms since it provides a dynamic control of sensors (sampling rate) based on the specific technical specifications and the current resources state (battery level, number of running applications) of the mobile device. In addition, FTW provides a mechanism that determines when to upload the collected data to the server based on the priority of data, Wi-Fi availability and other configuration parameters of the mobile device. The key contributions of our work can be summarized as follows:

- A novel open source framework for developing people-centric sensing android applications. Through FTW, developers would be able to exploit and configure all the embedded sensors of mobile phone as well as external sensors. The general configuration properties of FTW are the sampling rate, the duration of each data collection, the priority of data and the running environment (background/foreground). Additionally, developers can specify whether or not the data will be uploaded on a server and how often this will take place. FTW framework can be downloaded from `http:/grid.ucy.ac.cy/FTW/`.
- We evaluate the proposed framework demonstrating its expressivity, the utilization of resources and the ease of use in developing a people-centric application using its built-in features.

The rest of this paper is structured as follows. Section 2 gives a brief overview of relevant frameworks. Sections 3 presents the system architecture of the proposed framework. Section 4 demonstrates the FTW framework. Section 5 gives an insight to the work that will follow on the framework and concludes the paper.

2 Related Work

Participatory sensing enables collection of environmental sensory data by ordinary citizens, through devices such as mobile phones, without requiring any pre-installed infrastructure. Despite the radical increase of mobile applications mainly due to the popularity of smartphones and app stores, there are still only a limited number of applications for participatory sensing purposes [4]. The reason

Table 1. Mobile Frameworks for Participatory Sensing

	Jigsaw [8]	MyExperience [9]	SensLoc [6]	SociableSense [11]	Funf [1]	ODK [3]	FTW
`Embedded Sensors`	microphone accelerometer GPS	GPS Bluetooth	WiFi accelerometer GPS	microphone accelerometer Bluetooth	all	all	all
`External Sensors`	no	no	no	no	no	yes	yes
`Task Inference`	yes	yes	yes	yes	no	no	no
`Send data to server`	no	yes	no	yes	yes	yes	yes
`Dynamic Sampling`	no	no	no	yes	no	no	yes
`Decision module`	no	no	no	yes	no	no	yes

resides to the programming challenges of implementing such applications and especially due to resource constraints that prevent adoption of these applications in under-resourced environments. This main constraint reveals the challenges that participatory sensing still has to face: (1) Continuous sensing by making mobile applications more efficient and (2) creating mobile applications easier by leveraging the complexity of building such applications. Many frameworks and platforms have been proposed addressing the aforementioned areas. Some related work focus on specific sensors and tasks trying to achieve maximum efficiency, while other ones succeed a more general approach of the problem trying to ease the development of participatory sensing using all sensors.

Jigsaw [8], SociableSense [11], SensLoc [6] MyExperience [9], ODK Sensors [3] and Funf [1] are some indicative recent frameworks that enable mobile phones to collect sensor data important to a users daily life. The Jigsaw [8] mobile sensing platform balances the performance requirements of the applications and the resource demands of sensing continuously on the smartphone. SociableSense [11] is a framework that captures user behavior in office environments, while providing the users with a quantitative measure of their sociability with their colleagues. Novel energy efficient mechanisms that are adaptive with respect to the changing context and optimized according to the requirements of users collecting the sensor data have been implemented. SensLoc [6] is a platform that provides location context to applications by using a combination of acceleration, Wi-Fi, and GPS sensors to find semantic places, detect user movements, and track travel paths. MyExperience [9] is a framework for collecting data and capturing user's feedback. The innovative idea behind MyExperience is that it does not only collects quantitative data, but also qualitative such as user's opinion for the sensed event. Moreover, MyExperience provides an easy configuration mechanism throughout an XML-based configuration file. A key characteristic of all these frameworks is that they are focused on specific sensors and sensing tasks (like location queries, activity inference, etc). The ODK Sensors framework [3] provides a single sensing interface for both built-in and external sensors, hiding many details involved in developing sensing applications. The ODK Sensors framework provides an abstraction on which to develop and deploy user-level device drivers on Android smartphones.

The work most similar to ours is Funf [1], which is an open sensing framework developed by MIT Media Lab[2] for creating participatory sensing applications that takes account all of phone's embedded sensors. The core concept is to provide an open source, reusable set of functionalities, enabling the collection, uploading, and configuration of a wide range of data signals accessible via mobile phones. Funf contains reusable and expandable code and abstracts data and sensors through probe data structure.

The proposed FTW framework differs from existing ones in several aspects. Contrary to SensLoc, MyExperience and SociableSense, FTW does not focuses on specific sensors and task inference. Instead, FTW, ODK Sensors and Funf frameworks take into account all the phone's sensors. ODK Sensors framework aims at packaging software so that non-technical users can access external sensors from a locked mobile device running a stock version of Android operating system. As opposed to our work, Funf framework and ODK Sensors framework do not provide any mechanism for dynamic control for the process of sensing. Additionally, Funf and ODK Sensors framework do not support any mechanism that will decide when to upload the accumulated data to the server with respect to the priority of data, WiFi availability and programmer's configuration. Table 1 summarizes the key characteristics of the existing frameworks.

3 System Architecture

3.1 Overview

FTW uses the Android SDK and Java Runtime environment in order to develop the services that provides at the client-side and at the server-side respectively. The client-side component allows the collection of values from sensors embedded on the smartphone or from external sensors (the platform of Sensaris senspods[3], which is running under a RTOS-based kernel, has been implemented) connected to the device via Bluetooth connectivity. Collected values can be aggregated and/or transformed locally on the client and uploaded to the server-side component in real-time or at a later moment. The server-side component gathers, processes and visualizes the retrieved values. Android OS has been selected as the target platform for FTW framework because it is open source, has more features unlocked compared to the rival operating systems (background services) and supports connectivity with more external sensors. Our high level goal is to minimize effort for the developer. This includes both developers who are using

[2] Funf. `http://funf.media.mit.edu/index.html`
[3] Sensaris. `http://www.sensaris.com/`

the FTW API and Android library to build apps, as well as developers extending the FTW framework at the lower levels. Below, we list the design goals for FTW:

– be expandable ensuring the ease of developing applications so both experienced and inexperienced users will be able to use it effectively to create applications for participatory sensing, providing a high-degree of isolation between applications and sensor-specific code;
– support all the embedded sensors of mobile phones, facilitating the integration of new sensors into applications without requiring modifications to the OS configuration;
– provide a background service for monitoring the collection of data in order to ensure that the collection process doesn't affect user's experience. This service will dynamically configure sensors based on phone's technical characteristics and the phone's current resources state;
– provide functions which automatically determine when the data should be uploaded to the server.

3.2 FTW Data Types

FTW has a common interface for each unique data type. Specifically, each type of sensor data collected by the system is encapsulated as a conceptual "IData" object. IData gives an homogenous representation of sensor data (generic sensor data, location data, WiFi data) in the framework. Figure 1 depicts the IData interface. The abstraction for implementing and operating sensors is captured by BaseSensor class (see Figure 2). Using the modular BaseSensor architecture, it is easy to add new sensors to the system, or swap existing sensors with an improved version. All sensors support a common set of behaviors, and each one defines a set of configuration parameters that control it, and the format of its output. Sensors can be configured locally on the device or remotely through the server. Figure 3 depicts the class diagram of FTW implementation. It is an automatically generated diagram which shows the dependencies between the classes. BaseSensor class represents an abstract sensor. It contains the android service implementation and the code which is responsible for the communication with the modules of the FTW, such as communication service and monitoring service. Inheriting BaseSensor class, we can create our own sensor instances for sampling sensor data. Currently, 16 different types of sensor data are collected. Specifically, FTW software framework includes all the embedded sensors of the phone as well as external environmental sensors from the Sensaris senspods platform.

3.3 Data Formats

Sensor data are saved by default as CSV values and stored locally in the phone. Each write is made on a different file in order to prevent complete data loss in case of file corruption and to allow periodic data upload to the back-end server. Since many users do not have a mobile data service plan, the system

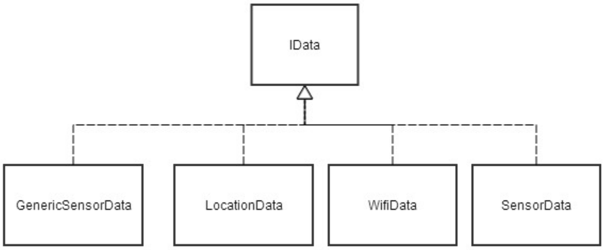


Fig. 1. IData interface

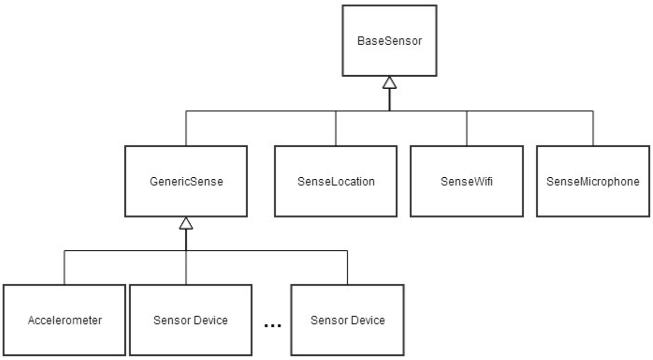


Fig. 2. BaseSensor Class

has been designed by default in the absence of network access and the phone accumulates the collected CSV files locally. When server connection is established (for example a participant connects to Wi-Fi), the system attempts to upload files in order to be further processed.

3.4 FTW Components

FTW is a two-sided application with the client-side running on the mobile device and the server-side running on the server. On server-side we provide a simple Application Programming Interface (API) for easy communication with the client-side of the framework. More specifically, two modules are provided: Messaging manager and File transfer manager, which both aim to leverage the complexity of creating the receiving on server-side. On client-side our main purpose is to provide a modular framework for creating new sensing modules by abstracting the way we handle data. Figure 4 depicts the architecture of the FTW middleware platform. In the following paragraphs the modules of the proposed FTW framework are described.

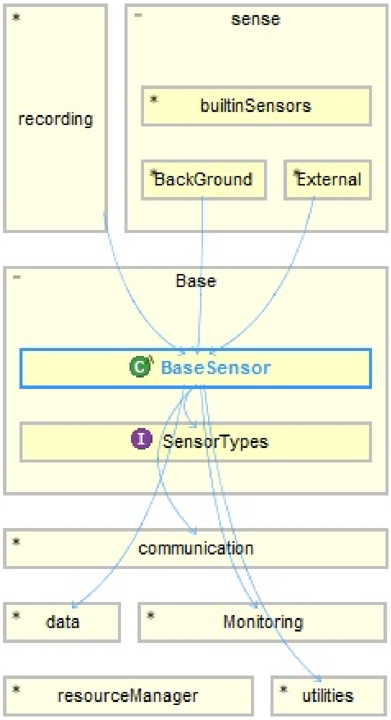


Fig. 3. FTW Class Diagram

Client-Side Data Layer

- Foreground Sensing. An abstract implementation that enables developer to create an instance for a foreground sensor configuring the sampling rate, duration and priority of data. The collection of data is made in the foreground and stops when the application is closed.
- Background Sensing. An abstract implementation that enables developer to create an instance for a background sensor configuring the sampling rate, duration and priority of data. The specific module is implemented as an Android service and the sampling is continuing even when the foreground application is closed.
- External Sensing. This module leverages the complexity of integrating an external sensor in the framework. FTW provides a mechanism to connect and pair the external sensor with the mobile device.
- Resources Service. This module runs on the background and collects data which are related with the resources of the mobile device. Collected data can be categorized in the following two categories: 1) static data, which rarely or never change (processor clock, ram) and 2) dynamic data, which are changing over time (battery level, running processes).

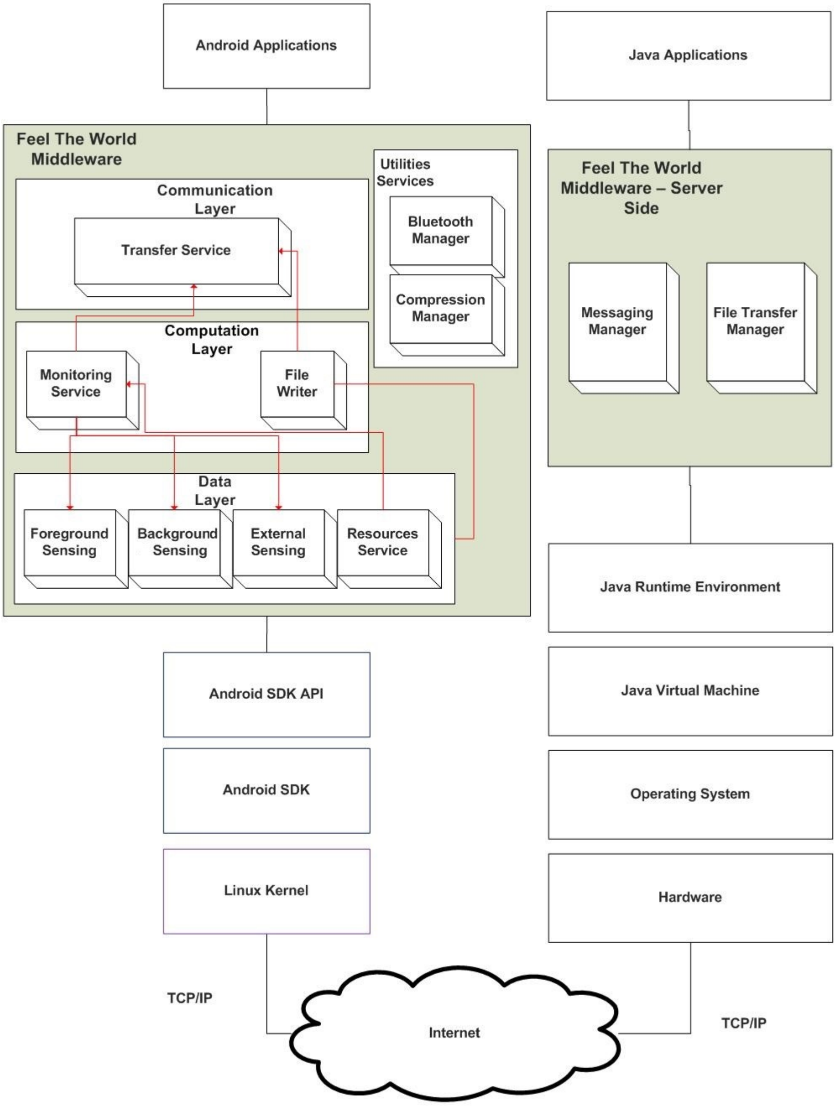


Fig. 4. FTW System Architecture

Computation Layer

- Monitoring Service. This module is responsible for monitoring the collection of sensor data. Monitoring service is able to dynamically configure sensors based on phone's technical characteristics and the phone's current resources state. Using this module, the developer can set specific rules that affect the sampling of sensor data. For example, when battery is below 20% then the sensors will stop sampling. Monitoring service includes a decision mechanism

which decide when to upload the accumulated data to the server with respect to the information received (e.g., priority of data, Wi-Fi availability and programmer's configuration). The decision module processes all information received and generates an action plan when to upload the data to the server.
- File Writer. This module writes data to CSV files.

Communication Layer

- Transfer Service. This module is responsible for data communication with the back-end server. It includes rules whether the data should be sent to the server. By default, when a participant connects to Wi-Fi, the system attempts to upload files.

Utilities Services

- Bluetooth Manager. This module provides a useful API for discovering and pairing nearby Bluetooth devices from the application. It also provides the capability of exchanging data between the mobile and the Bluetooth device.
- Compression Manager. This module compresses data using a widely known compression approach, the Deflate algorithm [5]. The Deflate algorithm divides the compressed data in variable-length blocks using a sliding window that can be up to 32 kilobytes large. Several compression algorithms have been tested [2]. Specifically, we tested the following compression techniques: a) Huffman coding, b) a combination of Huffman and Run-Length coding, c) ZIP compression, and d) Deflate compression. Deflate compression technique achieved the best compression ratio for both small and large length of strings. Huffman coding and Run-Length did not achieve data compression since these encodings are mainly used for compressing images. According to [12], experimental results have also shown that Deflate is the most energy-efficient algorithm.

In the next section, we have developed a few example applications to demonstrate reuse, flexibility, and extensibility of the framework.

4 Evaluation

In this section, we demonstrate the FTW framework capabilities. Specifically, we focus on demonstrating its expressivity, the utilization of resources and the ease of use in developing a people-centric application. All experiments were conducted using Google Nexus-S mobile phone.

4.1 Expressivity

With the term "expressivity" we refer to the ability of framework to utilize as many sensors as possible and the provided freedom to the configuration of

Table 2. FTW Expressivity

Sensor	Background Sampling	Rate (samples/sec)	Duration (sec)	Monitoring	Log data	Send to server
`Accelerometer`	yes	600	60	yes	yes	yes
`Accelerometer`	no	100	30	no	no	no
`Light sensor`	yes	500	60	yes	yes	yes
`Microphone`	yes	500	60	no	yes	no
`WiFi`	yes	1000	-	no	no	no
`Location-GPS`	yes	2000	-	no	no	no
`Location Network`	yes	2000	-	yes	no	no

sensors. In order to demonstrate the expressivity of FTW we have instantiated several sensors with different characteristics and configurations. We can see from Table 2 where we present 7 indicative sensor instances, each with different characteristics. For example, we have two accelerometer instances running with different sampling rate and duration, one running in foreground and the other in background implemented as a service.

4.2 Resources Utilization

In order to understand the automatic monitoring service and the way it handles battery utilization, we created two similar applications. Both applications gather accelerometer data and have the following configuration: background service runtime environment, sampling rate 30 seconds and duration 5 seconds. Their only difference is that the first application utilizes automatic monitoring service whilst the second one does not. The monitoring service handles the sampling rate of the first application based on the current battery level. The monitoring service runs on the background on specific rate and each time it runs sets the sampling rate as follows:

$$(\% \, of \, used \, battery \, + \, 1) \, * \, initial \, sampling \, rate \tag{1}$$

For example, when battery reaches 80%, then monitoring service will set sampling rate at 1.2 * initial sampling rate. This means that as long as the battery extinguishes, the sampling rate is becoming slower and slower. Our purpose here is to understand the impact of monitoring service in the utilization of resources at the mobile devices. In order to address it, we consider two similar applications which are running for 4 hours; the phone is plugged in to the power supply. We observe that the battery consumption is the same for both applications. This was expected since battery level was 100%. Next, we detach the phone from power supply and we observe that the battery utilization of the application which uses the monitoring service of FTW is gradually under 4% while the second one has a battery utilization of 6%. From these two applications, we observe that monitoring service utilizes in an efficient way the resources of a mobile device since it dynamically adapts the sampling rate as well as the duration of sampling.

4.3 Ease of Use

In order to demonstrate the ease of use of FTW framework, we developed a people-centric application for gathering data, called FeelMe. FeelMe application can be used by scientists or self-trackers that are willing to gather and share their daily data, their environment, or even their location. The motivation of FeelMe is to provide citizens a wide range of crowdsourced city information including traffic flow, road reports, parking places, and even warnings about where the latest speed traps have been set up. The user can take a picture of a place or an area where a problem has been detected through the mobile device's camera and report it with a description and title. The external sensor devices, which are embedded with GPS, detect CO_2, HUV, motion, noise measurements and sent these data to the mobile device through Bluetooth connectivity, and then the data are sent to server through TCP. The server stores all these data and depict them in a live map.

On the left screen in Figure 5, the user can choose the sensor that he is willing to gather data. FeelMe provides a wide range of settings, such as sampling rate, duration of each sampling etc. On the right screen in Figure 5, the user can observe the stream of collected sensor data. Moreover, the application collects location data and displays them on a map as depicted in the left of Figure 6. Finally, the application displays the current resources of the phone (on the right

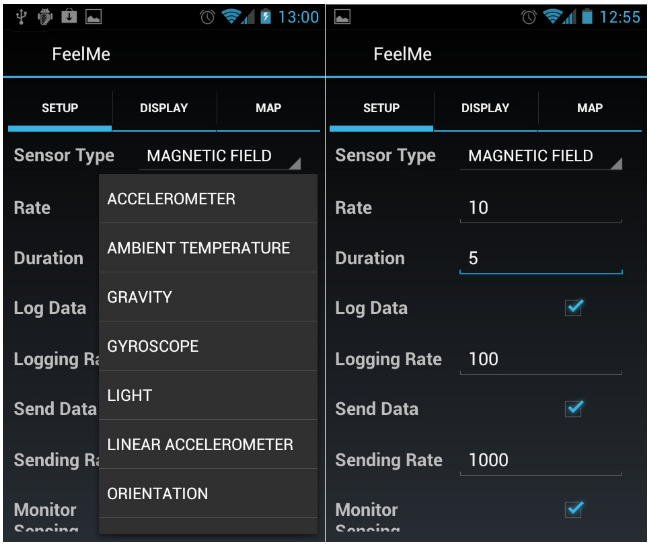


Fig. 5. Sample screenshots: FeelMe Application setup display

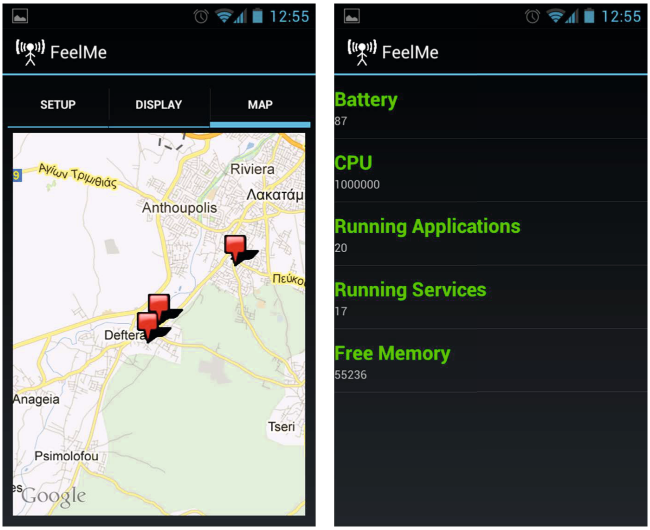


Fig. 6. Sample screenshots: (left) FeelMe Application map display, (right) FeelMe Application resources display

screen of Figure 6). For the development of this application we used the built-in features of FTW through its API. To sum up, the development of FeelMe application required about 320 lines of code.

Users who would like to participate in and contribute to *FeelMe* install the FTW runtime from `http:/grid.ucy.ac.cy/FTW/` on their mobile smartphones. Currently, FTW uses the Android SDK (Software Development Kit[4]).

5 Conclusion – Future Work

The great increase of smartphone users creates a need and opportunity to involve these devices in data gathering mechanisms that will provide projects with the needed data. Our focus in this work was to provide a framework that is scalable, expandable and maintainable. FTW ensures the ease of developing applications so both experienced and inexperienced users will be able to use it effectively to create applications for participatory sensing. Finally, an Android application has been built using the FTW framework, which makes use of many of its built-in features. The application can be used by citizens interested in collecting and exploring information related to the mobile device, its environment, and its user's behavior.

A key characteristic of the proposed framework is its extensibility. As future work, we plan to extend FTW framework so as to take into account tasks priority, security, and storage issues. In terms of tasks priority, the proposed

[4] `http://developer.android.com/sdk/index.html`

framework does not support any policy since it considers that all tasks have the same priority. However, some tasks are more time critical than other ones and therefore they should have higher priority. Regarding security issues, collected data contain sensitive private information. Although data are sent anonymously, no encryption/decryption protocol is used between sender-receiver in the existing framework. For the furure, we plan to extend FTW so as to allow storing and sending data in encrypted format for security purposes. In terms of storage aspects, in the current version of FTW framework the data are stored as CSV files. In the future, we plan to switch to SQL-lite in order to be able to index data and retrieve them more efficiently.

Acknowledgement. This work was supported by the third author's Startup Grant, funded by the University of Cyprus.

References

1. Aharony, N., Pan, W., Ip, C., Khayal, I., Pentland, A.: Social fmri: Investigating and shaping social mechanisms in the real world. Pervasive Mob. Comput. 7(6), 643–659 (2011)
2. Barr, K.C., Asanović, K.: Energy-aware lossless data compression. ACM Trans. Comput. Syst. 24, 250–291 (2006)
3. Brunette, W., Sodt, R., Chaudhri, R., Goel, M., Falcone, M., Van Orden, J., Borriello, G.: Open data kit sensors: a sensor integration framework for android at the application-level. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys 2012, pp. 351–364. ACM, New York (2012)
4. Das, T., Mohan, P., Padmanabhan, V.N., Ramjee, R., Sharma, A.: Prism: platform for remote sensing using smartphones. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys 2010, pp. 63–76. ACM, New York (2010)
5. Deutsch, P.: Deflate compressed data format specification version 1.3 (1996)
6. Kim, D.H., Kim, Y., Estrin, D., Srivastava, M.B.: Sensloc: sensing everyday places and paths using less energy. In: Proceedings of the 8th ACM Conference on SenSys, pp. 43–56. ACM, New York (2010)
7. Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musoles, M., Fodor, K., Ahn, G.-S.: The rise of people-centric sensing. In: IEEE Internet Computing, pp. 12:12–12:21 (July 2008)
8. Choudhury, T., Lu, H., Yang, J., Liu, Z., Lane, N.D., Campbell, A.T.: The jigsaw continuous sensing engine for mobile phone applications. In: Proceedings of the 8th ACM Conference on SenSys, MobiSys 2010, pp. 71–84. ACM, New York (2010)
9. Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B., Landay, J.A.: Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services, MobiSys 2007, pp. 57–70. ACM, New York (2007)

10. Lane, N.D.: Community-aware smartphone sensing systems. IEEE Internet Computing 16(3), 60–64 (2012)
11. Rachuri, K.K., Mascolo, C., Musolesi, M., Rentfrow, P.J.: Sociablesense: Exploring the tradeoffs of adaptive sampling and computation offloading for social sensing. In: Proceedings of the 17th MobiCom, Las Vegas, USA, pp. 71–84. ACM (2011)
12. Tsiftes, N., Dunkels, A., Voigt, T.: Efficient sensor network reprogramming through compression of executable modules. In: 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2008, pp. 359–367 (2008)

The Influential Factors of M-Government's Adoption in the Developing Countries

Vasileios Yfantis[1,*], Konstantina Vassilopoulou[2], Adamantia Pateli[3], and Abel Usoro[1]

[1] University of the West of Scotland, Department of Computing, Paisley, UK
Byfantis@yahoo.com, Abel.Usoro@uws.ac.uk
[2] Harokopio University, Department of Informatics & Telematics, Athens, Greece
kv@hua.gr
[3] Ionian University, Department of Informatics, Corfu, Greece
pateli@ionio.gr

Abstract. The application of E-government through the use of mobile devices is a challenge for all the countries. Although developing countries are slow in adopting the new technologies, most of their habitants are already familiar with the use of mobile devices. Despite that, the implementation of m-government is still a problem. Few scholars have researched the adoption of m-government by the developing countries and the influential factors that affect this adoption. The current research explores the influential factors of the m-government adoption by improving the Unified theory of acceptance and use of technology model. The improved UTAUT model derives from the additional elements of trust, context of use and human development index. The improved UTAUT contributes in the current m-government research by building a conceptual model that will be useful for scholars and policy makers so as to detect the potentiality of the developing countries to adopt m-government.

Keywords: m-government, Unified Theory of Acceptance and Use of Technology, human development index, trust, context of use.

1 Introduction

Governments around the globe utilize e-government services to improve citizens' interaction with the public administration. Recognizing the benefits e-government the government of Angola in 2005 implemented a portal, offering 157 public services online, 28 official forms, several useful documents and the opportunity to arrange appointments with the public servants [1]. Improved transparency in the public services and increased citizen feedback resulted in gaining the Technology in Government Award of 2007[1]. The government of Lebanon in 2008 developed a portal encouraging online transaction between taxpayers and the Lebanese Ministry of Finance [2]. Consequently citizens' visits to the tax offices shortened, i.e. from 1548 to 1,075 and VAT tax reduced from 11,344 to 7940 for the LTO type of tax [2].

* Corresponding author.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 157–171, 2013.

Successful case studies reveal peoples' demand for e-government services provision in the developing countries.

As citizens demand for better e-government services mobile device penetration acts as the main force to move from e-government to m-government. According to the ITU the global rate of mobile-cellular penetration stands at 96%; 128% for developed countries; and 89% for developing countries [5]. Growth was driven by developing countries, which accounted for more than 80% of the 660 million new mobile-cellular subscriptions added in 2011 [5]. For example, in Africa 151 million people or fifty seven percent of the whole population is subscribed to a mobile service provider. [3]. Mobile devices are very popular even among nations that have a high percentage of digital divide, e.g. South Africa mobile phone ownership reaches 84%, Nigeria 71% Kenya, Ghana and Botswana approximately 60% [4].

The significant findings of the mobile phone's ownership reveal that developing countries have potential dynamics to adopt mobile services. The International Communication Union defines m-government as "*the adoption of mobile technologies to support and enhance government performance and foster a more connected society*" [5]. Studies reveal a great interest in using mobile phones to interact with government and with other members of the community [6] leading to significant opportunities for both government and citizens, i.e. greater cost optimization through bypassing infrastructure [7], improved integration, communication and interaction, expanded service delivery and stronger digital equality [5], [8], [9]. The low quality of the infrastructure [10], a poor transportation system, high rate of criminality are among others significant reasons for choosing mobile phones as a means of communication with the government.

M-government is the next step from e -government which is already present in most developing countries. Unfortunately at the moment the adoption of m-government by developing countries is poor [11]. The main causes of the problem are the lack of technological infrastructure and the absence of the political will to promote transparency [12].

The motivation for this exploratory research derives from the fact that while mobile devices' usage has a tremendous influence upon peoples' lives and citizens welcome e-government services still m-government adoption lags behind. The use of the mobile devices for the transactions between the government and the citizens is a challenge due to the modern technology which is being used. In the developing countries where low access to information exists, m-government could be the means to set up the requirements for the transformation into the information society and the reduction of the digital divide. The Information Technology factors that support this transition should be explored by two different perspectives: The user's side and the mobile device's side. From the user's side it is important to research the behavioral and cognitive aspects that influence the human – mobile device's interaction towards the adoption of m-government activities [13]. Additionally, from the device's side it is important to find the technical aspects in terms of software and hardware that would affect the adoption of m-government. The scope of this paper is to build a conceptual model that takes into consideration both human and machine oriented aspects that will affect the transition to m-government. The research question in our case is *which are the factors that affect m –government adoption in the developing countries?*

The structure of this paper is divided into three parts: the theoretical framework, the discussion and the conclusion. The theoretical framework which is the main part of this research effort is trying to build a conceptual model by improving the Unified Theory of Acceptance and Use of Technology (UTAUT) model. It is explained in detailed how the model is constructed and how it could be improved so as to fit the unique characteristics of the developing countries. During the construction of the model, the three new elements of the model are being analyzed: Human development index, trust, context of use. Each addition of the element is supported by the appropriate arguments that show why this new element is important for the research and suitable for the developing countries. The second part of the research, discusses the findings of this academic effort and comments on the potential use of the new conceptual model. Finally, it refers to the contribution of the model for the citizens, the government and the scholars on the topic of e-government. The concluding part of the work focuses on the importance of the research and the future applications of the proposed conceptual model.

2 Theoretical Framework

2.1 The UTAUT Model

There are three aspects that we should take into consideration in our research: Developing countries, government, and mobile technology. The combination of these aspects triggers a mode of thought that a conceptual model should be formed so as the usage of the mobile technology will be a means to implement the transactions between the government and the citizens in the developing countries. We suggest that each of the three aspects is affected by a factor that moderates them. The three suggested factors as they are shown in the figure 4 are: Human development index, context of use and trust:

Human development [14] is a crucial factor because it is an index that could identify if the country is a developed or developing one. In this case, it empowers the research because the proposed model focuses only on the developing countries. The inclusion of a specific factor such as the human development index, which is used by the United Nations, will also make the conceptual model workable for international institutions that have already access to the human development data that United Nations publishes every year.

Context of Use is incorporated into the ISO 13407 standard on human-centered design [15] and "*consists of the users, tasks and equipment (hardware, software and materials), and the physical and social environments in which a product is used*." It is an important factor because it describes the environment of use. It helps since it addresses issues associated with the environment where the mobile device is used [16].

Every type of transaction between two or more parties has to be trust dependent [17] and be based on a set of rules in order to be applicable. Otherwise, the parties will not proceed with a transaction that will harm their interests and with no security measure than confirms the successful result of the transaction. During the transaction between the government and the citizens, trust is a factor that will help the citizens adjust themselves to the experience of dealing with the government. The more people trust the government, the more will repeat the administrative transaction again. Trust, as a factor of influence in the government will help us build a conceptual model that it is trusted by the citizens as well.

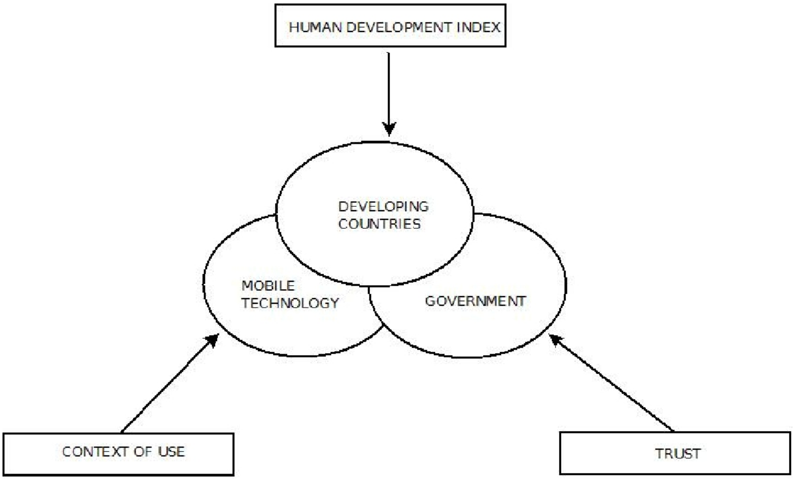


Fig. 1. Criteria for m-government adoption

The three factors are associated with the human computer interaction [18] and include both human and technology oriented knowledge. In order to detect the influence of each factor, we have to use a theoretical model of the Information systems theory [19] that will help us find the potentiality of the mobile technology acceptance from the developing countries in order to help the operation of the governmental services.

In the developing countries, identifying the user is very important because of the large number of disadvantaged populations [20], i.e. populations that suffer from poverty, lack of literacy or other limitations. Using UTAUT [21] as a basis model to formulate hypotheses will highlight the countries' needs in terms of infrastructure and improve the populations' adjustment in the adoption of modern technologies [22].

The most popular theoretical model that explores the adoption of the technology is the technology acceptance model [23]. TAM derived from the theory of reasoned action (TRA) [24], and theory of planned behavior (TPB) [25], while broadly accepted model, many researchers have tried to improve it. The unified theory of acceptance and use of technology [26] is an improvement of TAM. According to UTAUT, there are four determinants for the behavioral intention and use behavior. When referring to the behavioral intention (BI) [21], it means the intention of the user to use the technology. Use behavior (UB) [27] refers to the acceptance of technology. The four determinants are:

1. Performance expectancy (PE) [28]: The belief of a person that the system will help him improve the job performance.
2. Effort expectancy (EE): The expectation about the needed effort for the system's use.
3. Social influence (SI) [29]: The belief about how others perceive that the user should use the system.
4. Facilitating conditions (FC): The expectations regarding the infrastructure's existence to help the usage of the system.

The moderators of those determinants are: Gender, age, experience, and voluntariness of use. Gender has a psychological impact and age affects the attitudes. Experience is associated with the time that the user uses the system. Therefore, voluntariness of use defines the voluntary or the obligatory nature of the system's use (Figure 2) [26].

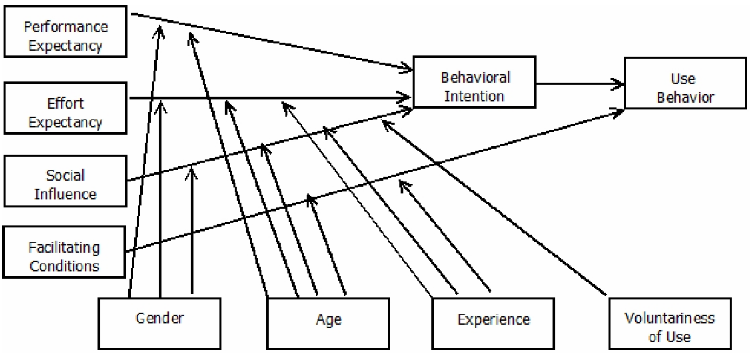


Fig. 2. Unified Theory of Acceptance and Use of Technology

Few if any studies have used UTAUT as a basis model for identifying adoption factors for m-government. Duraipandian and Rakesh in 2011 conducted an empirical research about the factors that influence the acceptance of e-government services in India [30]. The researchers used the UTAUT model as a tool in order to find the influential factors. Their customized version of UTAUT also included system expertise [31] as a determinant on the behavioral intention to use technology. Weerakkody et al in 2009 conducted an empirical research about the e-government adoption in Qatar [42]. They used the Information System theory of UTAUT by adding internet experience as a determinant for the performance expectancy (PE), effort expectancy (EE), social influence (SI), and the facilitating conditions (FC). On the other hand, Rokhman in 2011 explored the adoption of e-government in Indonesia. He used a combination of the innovation diffusion theory (ID) [32] and the perceived characteristics of innovating (PCI). The exploratory research in previous scholars that have researched the adoption of e-government in the developing countries proves that UTAUT as an extension of the TAM model has been used and has yielded important results (Table 1). In total, it seems that UTAUT is a workable model for the adoption of m-government in the developing countries and can be a suitable theory for our conceptual model.

Table 1. Previous works on m-government adoption

Authors	Theory	Sample size	Findings
Duraipandian et al. (2011)	UTAUT	170 persons - India	PE → BI, EE → BI, SI → BI, PR → BI, SE → PE, EE → PR
Weerakkody et al (2009)	UTAUT	1250 persons - Qatar	PE → BI, EE→ BI, SI → BI, FC → UB. BI → UB
Rokhman (2011)	Innovation diffusion / Perceived characteristics of innovation	751 persons - Indonesia	RA → P C → P
Abdelghaffar (2012)	TAM, Trust, Social Influence, Awareness, Culture, Internet Experience,	100 persons - Egypt	PU →I., C →I, SI → I, A→ I, FFI → I

2.2 The Influence of the Human Development Index

The definition «developing countries» has been criticized over the years from the non Western civilization such as the countries of the Latin America, that consider the definition of developed nation as a western civilization's cliché. Despite that disagreement, the definitions of developed and developing country usually refer to the power distance between two nations. The elements that empower a developed country are associated with its ability to maintain a high standard of living. United Nations uses the Human Development Index [33] for the measurement of the human development, which shows the development status of a country. Human Development Index consists of three different indexes:

1. Life expectancy index: Expected average years of life for a person [34].
2. Education index: The mix of the adult literacy rate and the number of students that are enrolled in educational institutes of different educational grades.
3. Income index: The Gross National Income per capita at purchasing power parity.

The United Nation's Human Development Index of 2011 rates most of the African countries in the grade of the lowest human development index. This fact proves that Africa is a suitable population for research and should be taken into consideration so as to explore the e-readiness of the continent for m-government. Table 2 below shows several African countries that are featured in the low grade of the Human Development Index.

Table 2. African countries with low Human Development Index

Global Ranking	Country	Global HDI
143	Kenya	0.509
144	São Tomé and Príncipe	0.509
148	Angola	0.486
150	Cameroon	0.482
151	Madagascar	0.480

Human development is an index that would be crucial for the acceptance of m-government in the developing countries because all of the three elements of the index are partially related to the use of the mobile devices. According to a research by Lauren Soelberg Treasure, the mobile phone usage is positively linked to life expectancy [35]. Lauren based on data that derive from the United Nations and the World Bank, found out in 2010 that "there is actually a positive correlation between the number of mobile phone subscriptions per 100 people and a country's average life expectancy". A factor which may positively affect the life expectancy is the usage of the mobile services for accessing m-health services. In the developing countries where the transportation infrastructure is poor and the rate of violence high, the distant medical advices could save lives of people who are unable to reach the local hospital or doctor's office. Most of the countries with low human development index would probably exploit the use of the m-government service for medical help because they are characterized by a high ownership of mobile devices and an instant need for the improvement of their health system. The remaining aspects of the human development index, education and income, are factors that influence the access to and the efficient use of technology. The gap between those who have access and use efficiently the technology and those who do not, it is called digital divide. The ownership of a mobile device for the access to the m-government system requires sufficient education level so as to become familiar with the mobile technology and sufficient financial resources so as to buy the device. In this case, it is common sense to realize that education and income as aspects of the human development index, will probably affect the potential adoption of an m-government system in the developing countries.

UTAUT is model examining factors that influence technology acceptance. UTAUT is a model which includes behavioral elements but does not take into consideration the human development that characterizes if a country is developing or not. Human development is expressed as mix of life expectancy, education and income. Each of these factors may influence equally or unequally the technology acceptance in the developing countries. Several of these factors may affect the perceived usefulness which Davis [23] describes as "*the degree to which a person believes that using a particular system would enhance his or her job performance*". There are several researchers that identify the influence of these human development factors in the perceived usefulness.

M. Khosrow-Pour suggests that organizations should drive their education efforts on creating knowledge bases that will help individuals to perceive technology as useful in their work [27]. Baaren, et.al. in their research about the perceived usefulness of the high definition television [36], refer to the household income as an influence towards HDTV. Life expectancy, as the third element of the human development index is already included in the UTAUT model with the factor age. In this case, a new moderator should be added under the name of human development. This factor will include education, income and age (as a replacement of life expectancy). This new moderator factor will improve the UTAUT model so as to fit the characteristics of a model that shows the technology acceptance in the developing countries.

2.3 The Influence of the Context of Use

M-government is a concept which focuses on the mobile technology that drives the public political will. M-government uses the mobile technology as a platform for its actual use so it is important to find if the developing countries accept the mobile technology. Since UTAUT is a research model that was built for the acceptance of technology in general, we should improve this model by emphasizing on the mobile technology. The research approach to solve this issue is to find the aspects of the mobile technology that affect the intentional behavior of the mobile technology.

In their study Quiao, et al. discusses the main challenges to the design of the mobile device interface [37]. One of the main aspects is the device's size and the output/input Facilities. Ronkainen, et al. focus on the importance of the context of use [38] as an important element of the mobile devices. Context of use as a term includes all the environmental, technical or social factors that affect the use of a system. The device's size and input / output facilities are included in the context of use. According to an empirical research by Liu and Li [39], context of use is "the strongest predictor of mobile game adoption". This fact enables context of use to be considered as a determinant for the improved UTAUT model.

The context of use is an important factor that may influence the adoption of m-government by the citizens. The human oriented aspects of the citizens of the developing countries may vary from country to country depending on the culture and aesthetics that each population preserves. For instance, social mobility is a term which defines the movement between the social classes for individuals or social groups. The developing countries that are characterized by a high degree of social mobility may be more open minded towards the adoption of the use of the mobile devices for m-government services. The more people move in the hierarchy of the social classes the more they prefer to have access to the public services from anywhere and anytime so as to take care of the appropriate administrative documents for their mobility.

2.4 The Influence of Trust

When discussing about e-government, trust is usually taken into consideration as a requirement for the implementation of the governmental activities. Trust between the government and the citizens, encourages both parties to act by overcoming any potential barriers. Trust is also defined by Alzahrani and Goodwin as the new aspect of the UTAUT model for the e-government citizen acceptance in Saudi Arabia [40]. When discussing about the trust we have to divide it into two types of trust: a) trust from the citizens to the technology b) trust from the citizens to the government.

Mobile technology is a recent one and it is sure that the awareness of this type of technology may be limited in the developing countries where the society lacks from a sufficient number of information resources. The non access to knowledge about the mobile technology may discourage people from using it for m-government services. In this case it is recommended that before the implementation of the m-services, the government should test a conceptual m-government model to the opinion leaders with mobile device ownership in order to use them as the driving force for the transfer of the m-experience to those who do not trust the mobile technology.

The trust of the citizens to the government as the provider of the m-service is another important factor for the adoption of m-government in the developing countries. If we compare the m-government services to the E-Commerce services then we realize that both terms share common infrastructure but m-government intends to service the common interest without taking into account the financial profit. Nevertheless, the research into the literature of E-Commerce would be helpful to adapt a suitable concept for the adoption of m-government. According to Bhattacherjee, the trust towards an E-Commerce's provider depends on three aspects of its infrastructure: ability, integrity, benevolence [41]. The degree of presence of these beliefs in a service that is being delivered through m-government would probably affect the trust of the citizens towards the government as a provider of these services.

3 Hypotheses Building

The unique characteristics of the developing countries should feature trust and context of use because the political environment in these areas is not always the ideal one for the application of the governance. Moreover, the lack of technical infrastructure and the imbalance regarding the power distance in the social classes may arise the question of how a common context of use of the mobile services could be created for the developing countries. The suggested conceptual model is an improved UTAUT model that derives from the inclusion of trust and context of use as determinants, along with the human development index as a new moderator of the determinants.

Trust in the use of m-government deals is associated with the three aspects of ability, integrity, and benevolence. When discussing about ability, it means the expectations of the citizen from the government that is capable of offering services through the use of the mobile devices. Integrity in m-government deals with the perception that there will be a set of regulations during the transaction between the two parties. Benevolence means the expectation that the governance through the mobile devices is being applied for good intentions and without taking into account the interest from persons to make profit out of it. Thus the degree to which each of the three aspects is true, then it might affect the user's intention to adopt a behavior towards the use of m-Government. The proposed concept leads us to the following hypothesis:

H1: Trust affects the behavioral intention of the use of m-government.

Context of use refers to all the factors that affect the use of an n m-government system. There are several factors that might affect positively or negatively the adoption of this system. For instance, the place where the citizen finds himself located might discourage him from using his mobile device if the weather conditions are not suitable. A really warm or cold place is not offering you the joy of using technology but it isolates your mode of thought only in finding ways to avoid the location as soon as possible. On the other hand, being in a high technology oriented city that operates by using smart technologies, could encourage using your smart cellular phone to enjoy technology services. The awareness of these facts leads to the following hypothesis:

H2: Context of use affects the behavioral intention of the use of m-government.

The rest of the hypotheses, derive from the known determinants of the UTAUT model, performance expectancy, effort expectancy, social influence, facilitating conditions:

H3: Performance expectancy affects the behavioral intention of the use of m-government.
H4: Effort expectancy affects the behavioral intention of the use of m-government.
H5: Social influence affects the behavioral intention of the use of m-government.
H6: Facilitating conditions affect the behavioral intention of the use of m-government.

The human development index refers to the elements of life expectancy, education, and income. Human development is an index which influences the adoption of m-government by affecting the determinants of the UTAUT model. The degree to which the human development is present in a country shows the country's potentiality to adopt innovativeness and adjust itself to a new situation. For instance, a country with high educated habitants could adopt faster the mobile technology and be more liberal towards the application of m-government. This results in building of the following hypotheses:

H7: Human development index affects the trust
H8: Human development index affects the context of use
H9: Human development index affects the performance expectancy
H10: Human development index affects the effort expectancy
H11: Human development index affects the social influence
H12: Human development index affects the facilitating conditions

The improved UTAUT model is shown in Figure 4 and includes: Trust, context of use and human development index as additional determinants along with all the hypotheses that are formed:

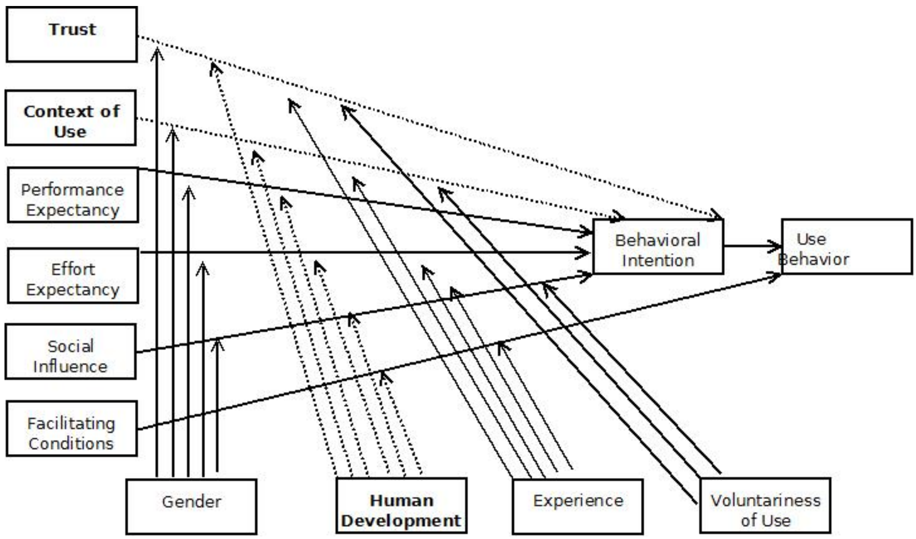


Fig. 3. UTAUT with trust, context of use, human development index

4 Discussion – Conclusion

Many researchers have analyzed adoption factors for e-government [30, 42]. Few if any studies have been carried out on m-government adoption [43, 44] and even fewer studies exist and examining adoption factors for developing countries. Limited research on m-government adoption leads to use conceptual models about e-government.

Trust and context of use, as new elements in our conceptual model are really important elements for the policy makers. The policy makers are the most influential persons before a new administrative procedure takes place in the nations. Their responsibility before suggesting a new policy makes them investing a large amount of time so as to find the appropriate policy. The inclusion of trust in the proposed model will act as an important keyword for the policy makers who want to recommend a trustworthy conceptual model for the government. On the other hand, Weerakkody and Duraipandian use technical oriented elements that might cause confusion among the policy makers [42]. Context of use is a factor in the proposed model that could be explored both from the technical and the human behavior's point of view. For instance, the social environment of the user and the device's user interface, both belong to the context of use. So, the competitive advantage of this element is its duality because it could draw the attention of IT and non IT policy makers.

According to the United Nations e-government survey 2012 [45], the least developed countries remain behind the big players in e-government level due to the lack of physical and human infrastructure. The recent statistics from the United Nations show the most progressive developing countries progress in terms of e-government (Table 3).

Table 3. E-government development in the least developed countries

	E-gov. development index		**World e-gov development ranking**	
Country	**2012**	**2010**	**2012**	**2010**
Samoa	0.4358	0.3742	114	115
Tuvalu	0.3539	N/A	134	N/A
Vanuatu	0.3512	0.2521	135	155
Lesotho	0.3501	0.3512	136	121

The political participation of the African people is discouraged by physical barriers such as corruption, fear, lack of funds and several other factors [46]. M-government could be the answer to all of these problems because the political participation through the mobile phones would raise the voice of the poor people that do not have the opportunity to protest in the streets or reach the governmental buildings.

The social and cultural status of the habitants is rapidly changing due to the flow of information that is being transferred from person to person and from device to device. The more people produce and receive information, the more gain knowledge to the information pool of new cultures and societies.

M-government is a new research field and it is really important to identify its advantages and disadvantages before it is applied for the citizen's service. The adoption of m-government by the developing countries is a challenge because these countries have low resources and infrastructure so as to use technology as a tool. Despite of the degree of the digital divide that these countries suffer from, the fact that they are familiar with the cellular phones and have not experienced m-government before will help us detect the strengths and weaknesses of the system. Of course, the specific characteristics of these countries trigger the adjustment of the research variables especially for them. The expected results will make us to reform our research and make it suitable for developed countries as well.

We are in the process of carrying out an empirical research in developing countries so as to test the current model. According to the potential results, we aim to revise the suggested model and adjust it to the needs of the developing countries.

References

[1] UnitedNations, Government Portal Project, http://www.unpan.org/DPADM/EGovernment/KnowledgeBaseofEGovernmentPractices/tabid/828/mctl/Study_View/ModuleID/2148/CaseStudyID/247/language/en-US/Default.aspx

[2] United Nations: Compendium of Innovative E-government Practices, New York, vol. III, p. 51(2009)
[3] Torora, B., Rheault, M.: Mobile Phone Access Varies Widely in Sub-Saharan Africa, http://www.gallup.com/poll/149519/mobile-phone-access-varies-widely-sub-saharan-africa.aspx
[4] Yelton, A.: Bridging the Digital Divide with Mobile Services. American Library Association, Chicago (2012)
[5] OECD, International Telecommunication Union: M-Government Mobile Technologies for Responsive Governments and Connected Societies: Mobile Technologies for Responsive Governments and Connected Societies. OECD Publishing (2011)
[6] Cupido, K., Van Belle, J.P.: Increased Public Participation in Local Government Through the Use of Mobile Phones: What Do Young South Africans Think? In: Proceedings of the 12th European Conference on e-Government, Barcelona, Spain, pp. 159–168 (2012)
[7] Zmijewska, A., Elaine, L., Seele, R.: Towards understanding of factors influencing user acceptance of mobile payment systems. In: Proceedings of IADIS International Conference, Madrid, Spain (2004)
[8] Rokhman, A.: E-Government Adoption in Developing Countries; the Case of Indonesia. Journal of Emerging Trends in Computing and Information Sciences 2(5) (2011)
[9] Ghyasi, F., Kushchu, I.: M-Government: Cases of developing countries. In: Proceedings of the 4th European Conference on E-Government, Castle Dublin, Ireland (2004)
[10] Foster, V., Garmendia, C.: Africa's infrastructure: a time for transformation. World Bank Publications, Washington (2010)
[11] Reddick, C.: Comparative E-government, pp. 241–258. Springer (2010)
[12] Rannu, R., Saksing, S., Mahlakõiv, T.: Mobile Government: and Beyond. Working Document. Mobi Solutions, Ltd. (January 2010)
[13] Carroll, J.: What's in It for Me? Taking M.Government to the People. Paper Presented at the 19th Bled eConference, Bled (June 5, 2006)
[14] Nagel, R.: Human Development Index - an Elaborate Means of Evaluating a Country's Hd. GRIN Verlag, Norderstedt (2007)
[15] ISO 13407: Human-centred design processes for interactive systems. Geneva: International Standards Organisation. Also available from the British Standards Institute London (1999)
[16] Maguire, M.: Context of Use within usability activitie. Int. J. Human-Computer Studies 55, 453–483 (2001)
[17] Clarke, K.: Trust in Technology: A Socio-Technical Perspective. Springer, Dordrecht (2006)
[18] Qiyang, C.: Human computer interaction: issues and challenges. IGI Global, Hershey (2001)
[19] Harindranath, G., Zupančič, J.: New perspectives on information systems development: theory, methods, and practice. Springer, New York (2002)
[20] Knight, G., Roosa, M., Umaña-Taylor, A.: Studying Ethnic Minority and Economically Disadvantaged Populations: Methodological Challenges and Best Practices. Amer Psychological Assn., Washington (2009)
[21] Ryan, M.J.: Behavioral intention formation: The interdependency of attitudinal and social influence variables. Journal of Consumer Research 9, 263–278 (1982)
[22] Sharma, S.: E-adoption and Technologies for Empowering Developing Countries: Global Advancements. IGI Global, Hershey (2012)

[23] Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly 13(3), 319–340 (1989)
[24] Ajzen, I., Fishbein, M.: Understanding attitudes and predicting social behavior. Prentice-Hall, Englewood Cliffs (1980)
[25] Ajzen, I.: From intentions to actions: A theory of planned behavior. In: Kuhl, J., Beckmann, J. (eds.) Action Control: From Cognition to Behavior. Springer, New York (1985)
[26] Venkatesh, V., Morris, M.G., Davis, F.D., Davis, G.B.: User Acceptance of Information Technology: Toward a Unified View. MIS Quarterly 27(3), 425–478 (2003)
[27] Khosrow-Pour, M.: Advanced Topics in Information Resource Management, vol. 1, pp. 106–107. Idea Group Publishing, London (2002)
[28] Aronson, E., Carlsmith, J.M.: Performance expectancy as a determinant of actual performance. Journal of Abnormal and Social Psychology 65(3), 178–182 (1962)
[29] Friedkin, N.E.: A Structural Theory of Social Influence. Cambridge University Press, New York (1998)
[30] Duraipandian, I., Rakesh, T.: Empirical study of factors influencing acceptance of e-government services in India. In: ICEGOV 2011 Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance, pp. 141–146. ACM, New York (2011)
[31] MacFarland, T., Parker, R.: Expert systems in education and training. Educational Technology, New Jersey (1990)
[32] Rogers, E.: Diffusion of Innovations, 5th edn. Simon and Schuster, New York (2003)
[33] United Nations: Human Development Report 1990. Oxford University Press, Oxford (1990)
[34] Sullivan, A., Sheffrin, S.: Economics: Principles in action, p. 473. Pearson Prentice Hall, New Jersey (2003)
[35] Lauren Soelberg Treasure, Mobile Phone Usage Positively Linked to Life Expectancy, `http://healthdecide.orcahealth.com/2012/09/05/national-cell-phone-usage-may-improve-life-span/`
[36] Baaren, E., van de Wijngaert, L., Huizer, E.: 'I Want My HDTV'? Underlying Factors of Perceived Usefulness for High Definition Television. In: Tscheligi, M., Obrist, M., Lugmayr, A. (eds.) EuroITV 2008. LNCS, vol. 5066, pp. 283–292. Springer, Heidelberg (2008)
[37] Qiao, L., Feng, L., Zhou, L.-z.: Information Presentation on Mobile Devices: Techniques and Practices. In: Zhang, Y., Yu, G., Bertino, E., Xu, G. (eds.) APWeb 2008. LNCS, vol. 4976, pp. 395–406. Springer, Heidelberg (2008)
[38] Ronkainen, S., Koskinen, E., Liu, Y., Korhonen, P.: Environment Analysis as a Basis for Designing Multimodal and Multidevice User Interfaces. Journal of Human-computer Interaction 25(2), 148–193 (2010)
[39] Yong, L., Hongxiu, L.: Exploring the impact of use context on mobile hedonic services adoption: An empirical study on mobile gaming in China. Journal of Computers in Human Behavior 27(2), 890–898 (2011)
[40] Alzahrani, M.E., Goodwin, R.D.: Towards a UTAUT based Model for the Study of EGovernment Citizen Acceptance in Saudi Arabia. World Academy of Science, Engineering and Technology 64(6), 8–14 (2012)
[41] Bhattacherjee, A.: Individual trust in online firms: Scale development and initial test. Journal of Management Information Systems 19(1), 211–241 (2002)

[42] Weerakkody, V., Al-Shafi, S., Zahir I., Habin, L.: E-Government Adoption In Qatar: An Investigation Of The Citizens' Perspective, Adoption and Diffusion Interest Group In Information Technology. Proceedings. Paper 3 (2009), http://aisel.aisnet.org/digit2009/3

[43] Vincent, J., Harris, L.: Examining adoption behavior of mobile government: How do we reach the tipping point? Information, Communication & Society 11(3), 395–413 (2008)

[44] Shareef, M.A., Archer, N., Dwivedi, Y.K.: Examining adoption behavior of mobile government. Journal of Computer Information Systems 53(2), 39–48 (2012)

[45] United Nations, E-Government Survey (2012), http://unpan1.un.org/intradoc/groups/public/documents/un/unpan048065.pdf

[46] Chatora, A.: Encouraging Political Participation in Africa: The Potential of Social Media Platforms. Institute for Security Studies (2012), http://dspace.cigilibrary.org/jspui/bitstream/123456789/32998/1/15Mar2012SocialMedia.pdf?1

NoizCrowd: A Crowd-Based Data Gathering and Management System for Noise Level Data

Mariusz Wisniewski[1], Gianluca Demartini[1], Apostolos Malatras[2], and Philippe Cudré-Mauroux[1]

[1] eXascale Infolab, University of Fribourg, Switzerland
[2] Pervasive and Artificial Intelligence Group, University of Fribourg, Switzerland
{firstname.lastname}@unifr.ch

Abstract. Many systems require access to very large amounts of data to properly function, like systems allowing to visualize or predict meteorological changes in a country over a given period of time, or any other system holding, processing and displaying scientific or sensor data. However, filling out a database with large amounts of valuable data can be a difficult, costly and time-consuming task. In this paper, we present techniques to create large amounts of data by combining crowdsourcing, data generation models, mobile computing, and big data analytics. We have implemented our methods in a system, NoizCrowd, allowing to crowdsource noise levels in a given region and to generate noise models by using state-of-the-art noise propagation models and array data management techniques. The resulting models and data can then be accessed using a visual interface.

1 Introduction

Big data has become a key element to support decision making at different levels of granularity, for example on continuous streams of data generated by sensors spread around a given environment (e.g., meteorological stations in cities). Analytics on sensor data is of high importance to support many critical decisions, such as when stopping the car traffic on certain days to limit pollution. However, sensors are expensive to install and maintain and need to be placed at specific geographical locations, which are typically application and context-dependent. Acquiring, deploying, and maintaining a large-scale sensing infrastructure for a given problem is hence difficult, costly and time-consuming.

In this paper we propose NoizCrowd: a crowd-sensing approach to big data generation using commodity sensors. The proposed system, built on top of a database management system for data-intensive applications, is able to collect streams of data from participating users' mobile devices and generate additional data thanks to statistical models based on the crowd input. NoizCrowd also provides dynamic visualizations of both sensor and model data to support user decision queries, for instance to determine regions with low noise levels in a given neighborhood to build a house or buy an apartment.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 172–186, 2013.

In more detail, NoizCrowd consists of four different components: i) a mobile application allowing the crowd to measure noise levels accurately using commodity smartphones ii) a scalable array storage layer to manage all the pieces of data gathered from the smartphones iii) a higher-level modeling layer capable of generating continuous models from the (potentially) sparse data gathered by the crowd and iv) a data export and visualization layer to interface with the end-users and support interactive decision making.

The proposed approach has the following advantages: i) it allows any mobile user to report the noise level at any place and time ii) it generates and manages large-scale data about noise levels in order to cover different regions and time intervals and iii) it supports users by answering data analytics requests on noise levels.

The rest of the paper is structured as follows. We start below in Section 2 by comparing NoizCrowd with previous work including sensor data generation applications running smartphones and noise mapping applications. In Section 3, we describe the architecture components of NoizCrowd in more detail, including the mobile application used to gather sensor data as well as the scalable storage system. We present in Section 4 the crowdsourcing model used to obtain noise-level data from geographically distributed users, as well as the data generation models used to integrate and extend potentially sparse user data. Section 5 presents the results of an experimental evaluation of the proposed system based on several live deployments. Finally, Section 6 concludes the paper by highlighting the main findings of our project.

2 Related Work

The notion of participatory sensing [2], namely user-centric monitoring and sensing of environmental conditions by means of high-end mobile phones, has recently emerged as a promising, low-cost alternative to traditional large-scale, costly and difficult to manage sensing infrastructures based on sensor networks [6]. Whereas there are several potential shortcomings for such an approach, especially in terms of privacy [4] and quality of data collection [11], its benefits nonetheless are far from negligible. In particular, attributed to the ubiquity of users with mobile devices equipped with sensors, participatory sensing applications can provide great data collection services at high granularity (spatial and temporal) and with a low cost [14]. Such applications, spurred by active research and development efforts in the domains of pervasive computing and the Internet of Things, are inherently distributed and lacking centralized infrastructures and therefore ensure robust operation and minimal management needs [3].

Participatory sensing applications have been considered for a wide range of sensed information, with typical examples including environmental impact [19], green vehicle routing [10], bargain shopping [7], etc. NoizCrowd focuses instead on the collection of urban noise levels using participatory sensing applications and their spatio-temporal representation on maps. This has generated great interest in the research community for a variety of reasons, the most important of which is the ubiquity of sound sensors, i.e. microphones, in current mobile devices. Moreover, urban noise levels are becoming increasingly important due to the related health concerns [18], as well as the associated regulatory frameworks and citizen concerns [1]. It is noteworthy that during an experiment regarding citizen engagement based on participatory sensing applications [8], people identified noise pollution as one of the most prominent information that should be monitored using applications like NoizCrowd.

NoiseTube [9] is one of the most interesting participatory noise mapping applications. It has been made available to the public through an open-source license and this has led to its widespread usage leading to contributions regarding noise levels for more than 250 locations around the world. NoiseTube allows users to annotate their data readings with social tags, thus allowing for semantic analysis of the collected data, while it additionally provides powerful signal processing techniques on the mobile phones to process received sound levels with a high degree of accuracy. NoiseMap [24] is an application built on the same principles as NoiseTube, with the distinctive characteristics of allowing users control over the collected data, while in parallel supporting real-time representation of user submitted data. Another application that maps noise levels in a city exploiting participatory sensing readings is SoundOfTheCity [23], in which semantic tags are also used to annotate noise readings. The SoundSense framework [15] also collects audio data from the user's vicinity, but instead of submitting raw data to the centralized participatory application it instead utilizes lightweight yet powerful machine learning algorithms to classify sound-related events that it then depicts on a map. Amongst other similar systems, we distinguish the NoiseSPY [12] application that maps noise levels using users' smartphones as sensors; the latter pre-processes the data prior to making it available, by calculating noise levels according to specific guidelines. This ensures more homogeneous data being represented on the map, namely classes of different noise levels, as well as reduced communication overhead in terms of information reporting.

As compared to existing systems, the advantages of NoizCrowd are twofold: NoizCrowd is capable of producing noise data in RDF format. This enables new kind of semantic applications by linking the generated data with other datasets in the Linked Open Data cloud[1]. Furthermore, the use of a scalable storage system together with noise propagation models to generate missing data guarantee better coverage of user requests both in time and space.

Regarding the collected data, there are a lot of issues concerning their granularity, i.e., lack of data for particular locations, which can compromise the operation of such participatory sensing applications. The Ear-Phone participatory noise mapping system [21] has a functional behavior similar to that of the previous applications and aims to address the problem of incomplete data in the context of noise collection. In this respect, it employs compressive sensing techniques and data projection models to enhance and account for the aforementioned problem. The simulation and real-world deployment evaluation results regarding data optimization are quite promising. In the same line of thinking,

[1] http://linkeddata.org

Mendez *et al.* present several data interpolation methods to improve the quality of incomplete and random data in participatory sensing applications targeting environmental monitoring [17]. With the same optimization goal, the DrOPS system that was recently presented in [20] utilizes model-driven approaches and online learning mechanisms to predict missing data readings from existing ones. We diverge from this work by focusing on noise data collection and their particularities, i.e. sound dissipation. Lastly, one of the biggest concerns in participatory sensing is ensuring that users are actively contributing and sharing their data [22], since it could eventually lead to poor performance due to the lack of accurate and informative representations. This problem has also been considered in the context of noise mapping, where a persuasive, motivating game was considered in [16] to stimulate user data collection and sharing.

3 Architecture

As stated above, NoizCrowd consists of a scalable system allowing the crowd to accurately measure noise levels using their mobile phones, an array-based storage component for data-intensive applications, a model generation and management layer, as well as a data export and a visualization interface to support decision making. The overall system architecture is depicted in Figure 1. The rest of this section presents each of those four components in more detail.

3.1 Data Gathering

For a noise level mapping system such as NoizCrowd, gathering vast amounts of data over wide areas is key. Deploying noise sensors over a large geographical area would introduce substantial deployment and maintenance costs. Our approach to solve this problems is to use crowdsourcing. Indeed, nowadays many people use smartphones with an integrated global positioning system (GPS), a microphone as well as an Internet connection. These are the elements needed to record noise levels, measure the coordinates of the location where it was measured, and send such data to the NoizCrowd back-end server.

Since smartphones were typically not designed as noise meters, the first problem we face is to map the sound levels one can capture with his or her smartphones to standards decibels. This requires a third-party sound level meter to calibrate the microphone of the smartphone. Since such devices are relatively uncommon, we decided to crowdsource this problem too; For each smartphone model, any user from the crowd can enter through our application a conversion table linking the noise levels as recorded by his or her device to real decibels as recorded by a sound level meter. Such conversion tables are then shared to all other participants having a similar smartphone through our application. We apply a majority-vote algorithm to select the most popular values in those conversion tables when several participants register different values for a given sound level.

The main goal of our smartphone application is to allow participants to determine and share the sound levels in their surroundings. Our application, which

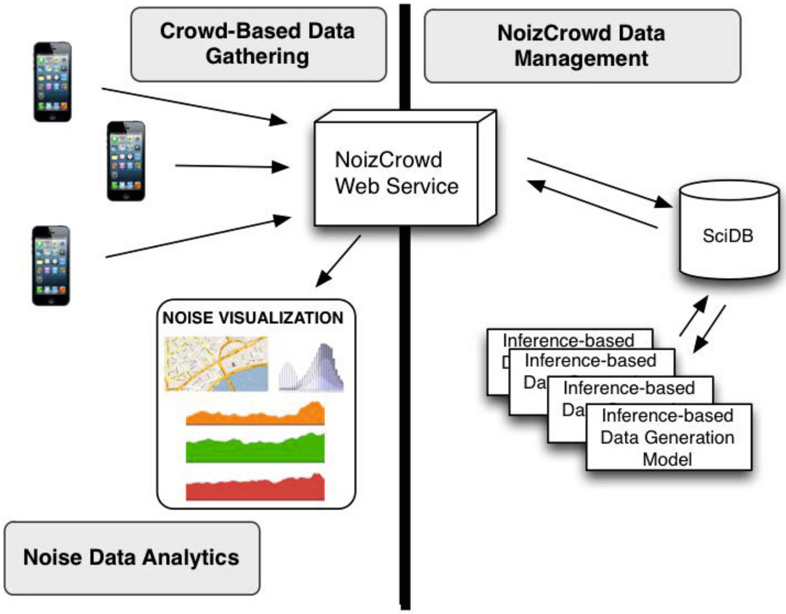


Fig. 1. NoizCrowd architecture diagram: participants can share noise level data in their surroundings using a smartphone application; All measurements are then durably stored in an array database back-end; users can query the system using a visual interface, which triggers the generation of high-level interpolation models based on the raw data.

is currently available for the iOS platform, allows the users to record the noise level at their current time and location by leveraging the embedded microphone and the iOS SDK audio recording functionalities. It returns the average noise level in decibels thanks to the crowdsourced conversion tables described above. Next, the application connects to the data back-end (see below) through a Web service and transmits the median and peaks of the measured values over few seconds.

3.2 Data Storage

We designed a new scalable back-end to be able to durably store and efficiently process all data points shared by the users. Since the data we are working with is highly non-relational (we are dealing with multidimensional data in space and time mostly in the context of this project), we decided to base our storage components on SciDB [5], a new array-based open-source database system for data-intensive applications. The back-end of NoizCrowd is responsible for three main tasks: receiving and durably storing all the data points shared by the users, providing all the required data to build higher-level models, and transforming and exporting relevant data for visualization purposes.

We store all value points shared by the user in multidimensional arrays. Two of the array dimensions represent spatial information (latitude and longitude), discretized using a fixed grid size[2]. The third array dimension represents an unbounded temporal axis, also discretized every hour in order to regroup temporally close measurements into one single value. All array values received by the crowd are durably stored in the database; the cell values are versioned [25] such that if a new value is received for a given array cell within one hour, it is stored as new versions of the older value.

This potentially extremely large array is stored as a compressed and sparse matrix on disk (i.e., only those cells that contain an actual value are materialized on disk). The array is chunked (i.e., split) both in time and space, and the resulting array chunks are stored on clusters of commodity machines if very large amounts of data are received from the crowd (see [5] for details on chunking and distributed storage in SciDB).

This array database layer is used to efficiently build higher-level data models (see below Section 3.3) by selecting slices of values in space and time. It is used in a similar manner by the data export and visualization component to extract relevant data and present them to the end-user to support aggregated information visualization and decision making.

3.3 Noise Modeling

The noise level values captured by the crowd are intrinsically sparse and noisy. We cannot expect all areas and time periods to be covered by end-users. Realistically, the data gathered will hence be highly skewed, with urban areas receiving way more data points than remote areas for instance. Also, the values collected by the users are inherently noisy, because of potential hardware differences between the smartphones (e.g., slightly different microphones used for a given smartphone model), and the high variability of ambient noise levels in urban areas.

To tackle both data sparsity and data fluctuations, NoizCrowd does not provide raw measurements to the end-users but builds instead higher-layer models from the data gathered. The modeling component that integrates overlapping values and generates missing data by interpolating crowdsourced observations is described in more detail in Section 4, while our interpolation models are experimentally validated in 5.

3.4 Data Export and Visualization

The data gathered from the crowd as well as the data generated by means of interpolation can then be used by end-users, either as an RDF export, or through a data visualization layer.

[2] Typically, we use a grid size of 10m given the accuracy of current smartphone positioning systems.

The export component regularly pulls data from the models and converts them into RDF triples for export. We use our own dipLODocus[RDF] system [26] to compactly store all data and to expose a SPARQL interface to whomever is interested in querying some of the exported data.

The visualization interface allows users to retrieve a representation of the noise levels of a region over a map. In the visualization component, the region we display is represented as a grid of squares with sides of 10 meters, hence directly matching the data stored in the array back-end and processed by our models. The temporal information (i.e., different noise levels at different points in time) is shown as an additional chart, as illustrated in Figure 2). An alternative visualization of NoizCrowd data may be obtained by adding a third axis for time and have our data model represented as a cube with latitude, longitude and time.

We use the Google Map API to display additional information at the location where the noise levels are visualized. This allows us to overlay noise levels on top of the map of a given region. Each time the user uses the zoom or moves the map, the model component is queried for new values corresponding to the displayed area. The opacity of our noise level overlay indicates the noise level, while the color indicates if the displayed values are the average sound level or the peak sound level, since the interface gives the users the choice of which value should be displayed. Right now each square is represented on the map, and if we zoom out too far, the overlays stop being displayed. We chose not to display

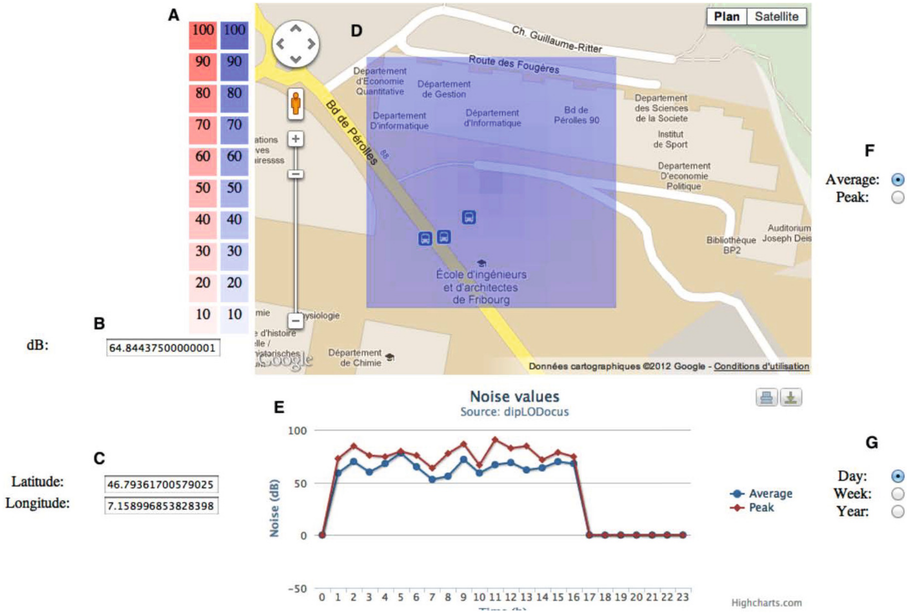


Fig. 2. NoizCrowd Visualization Interface

overlays in that case because of the important load on the database when all squares of a big spatial region are queried and because of the prohibitive cost of rendering a multitude of small squares on the client side.

We use Highcharts[3] to generate the graphs. Highcharts is a charting library written in JavaScript, offering an easy way of adding interactive charts to Web sites or Web applications. The graph is used to display statistics of a given area. The user can select what type of statistics he is interested in and then retrieves the statistics of an area by clicking on it on the map. Right now the interface allows the users to get the average of all the measurements of a given area for every hour of a chosen day, to show the average of all the measurements of the area for every day of the week, and to show the average of all the measurements of the area for every month of the year.

Both average and peak values are always displayed on the graph simultaneously using the same colors as for the map (red for peak values and blue for average values). The axes of the chart change according to the options chosen and the values returned by the database management system. A future improvement would be to allow the users to select a whole region of the map and to get detailed statistics over it.

4 Models

As compared to well positioned sensors, the use of crowdsourcing does not guarantee an even coverage of the monitored geographical area over time. Thus, there will be missing data points both on the time and space dimensions. Since NoizCrowd needs to be able to answer queries about any location and time interval, our system adopts data generation methods by means of interpolation over different dimensions. In addition, the use of crowdsourcing can add fluctuations to the measurements as described above, and hence require models to integrate the measurements taken by different participants. In the following, we present the various models we have developed to tackle these issues in the context of NoizCrowd.

4.1 Spatial Interpolation Model

NoizCrowd uses an interpolation model to generate missing data between two measurements that have been provided by the crowd and stored in the database over the same time interval. In order to obtain valuable interpolations we need to carefully select different measurements based on their geographical distance. Note that we do not consider temporal interpolation at this point and focus on space solely in the following. To restrict the inference to adjacent measurements only, NoizCrowd defines a maximal distance threshold on the distance between two measurements used for inference[4]. In order to maintain the computational

[3] `http://www.highcharts.com/`

[4] In the current version of NoizCrowd the maximum distance has been set to 50 meters.

complexity of the inference tractable given the large-scale data NoizCrowd potentially needs to handle, we defined computationally simple interpolation models specifically tailored for our array database back-end. User queries can be generalized in our case as *slab* queries, i.e., bi-dimensional range queries in space: $q = \{x_i, x_j, y_i, y_j\}$. Once such a query is received, the back-end proceeds to a full-scan of the array values overlapping the slab (such slab scans are extremely optimized in SciDB, since the data is store in a compressed, vertical and sparse format using multidimensional chunks). Once all the values stored in the database for the given query are selected, we interpolate the data and compute the noise level for every cell value v in the slab by using a k-nearest neighbor interpolation based on the Manhattan distance:

$$v_{i,j} = MD \sum_{k=1}^{n} v_k md^{-1}(v_{i,j}, v_k)$$

where md represents the Manhattan distance between two cells and MD is a normalization factor ($MD = 1/\sum_{k=1}^{n} md^{-1}(v_{i,j}, v_k)$). Figure 3 shows an example of our spatial interpolation with $k = 2$, i.e, interpolating values given their two nearest-neighbors only.

The above interpolation can be executed very efficiently and in a scalable manner in our array backend, by keeping a small index on the available values in main memory and by executing a parallel swipe over all the missing values and computing each new value in parallel.

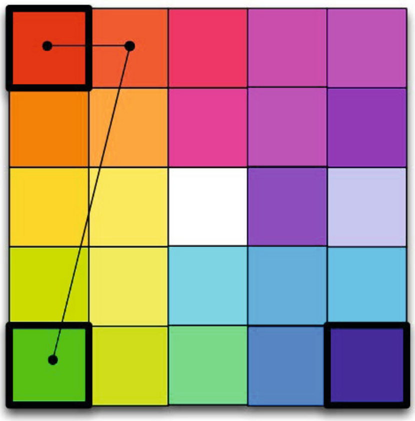


Fig. 3. Noise signal interpolation using two nearest-neighbors and three noise sources. The noise value of each region is computed as the average noise level of the two nearest noise sources.

4.2 Temporal Interpolation Models

We experimented with two different time interpolation models. For short time ranges (e.g., time intervals lasting a few minutes), we extend the spatial model above to take into account spatiotemporal slab queries and Manhattan distances in three dimensions. In order to infer data over longer unobserved time intervals (e.g., hours or days), NoizCrowd adopts inference models that look for common patterns in the available data. We focus on finding repetitive patterns based on hour and day intervals in that context. For example, if every Monday at 11 a.m. the noise level for a given area is around 50dB and if we did not get any measurement last Monday, then we can assume that there is a high probability that the missing value is also 50dB. As another example, if we had a value for a given area two hours ago and another similar one for the same area right now, then we can assume that doing an interpolation over those two values might lead to reasonable results. On the other hand, NoizCrowd does not perform interpolation for such cases when the values are too dissimilar (i.e., when for longer time intervals $|v_{t1} - v_{t2}| > \tau$, where τ is a system constant).

4.3 Noise Propagation Models

In addition to the interpolation models described above, we have created a model implementing the formula for the propagation of sound in real atmosphere based on Lamancusa's model [13]. Such complex models can be particularly useful in case we know the location of a sound source or want to locate a particular source given some measurement. The sound pressure level Lp is derived in this model as follows:

$$Lp = Lw - 20log(r) - 11 + DI - Aabs - Ae$$

Where Lw stands for the sound power level, r is the distance from the source in meters, DI is a directivity index, $Aabs$ is the atmospheric absorption, and Ae is the excess attenuation.

In free space, the directivity index value is 0dB while it is 3dB on a perfectly reflecting surface. Our models take the case of a perfectly reflecting surface into consideration. The atmospheric absorption is the energy dissipated in the air by viscous loss and relaxational processes over some distance. While the concept is easy to understand, it is difficult to measure such parameters using smartphones. However, we can set this variable as a constant if the region we take our measures in is known a priori.

The excess attenuation variable is defined as follows:

$$Ae = Aweather + Aground + Aturbulence + Abarrier + Avegetation$$

where $Aweather$ represents the meteorological conditions including the effects of wind and temperature. In our case, we only take the temperature into account (since we can get it directly using smartphone sensors or by looking up values given the GPS coordinates and time). $Aground$ represents the ground interaction and $Aturbulence$ the atmospheric turbulence. We take standard values for both parameters due to the difficulty of measuring/looking up such parameters in our context. *Abarrier* and *Avegetation* represent obstructions and vegetation. We do not take these parameters into account.

By using the formula above, our model can generates accurate data values for the area surrounding a noise source. Locating the source of some sound by using three or more measurements in decibels can also be performed using the sound propagation in atmosphere formula. For each of the measurement we give to the model, we check each nearby cell and measure the value it would have if it were the source. The square that has the closest resulting values for each measurement is selected as the most likely source. We experiment with such models in Section 5.

4.4 Models and Late Materialization

As described above, we store data in two different ways: We store raw measurements in our SciDB backend, and higher-level, cleaned and interpolated data using higher-level models. As for traditional view mechanisms in database systems, we thus have to choose how we materialize (i.e., precompute and store) the model data. Since some of the models described above can be computationally intensive, and since the total time and geographical areas covered by the system can be very large, we decided to opt for late materialization strategies. The materialization of the models is thus performed at query-time: NoizCrowd only generates model data when a request is sent by the visualization interface about a specific spatiotemporal range. The resulting data is then cached and indexed as a new array in our backend, and can also be cached for future requests.

The model views are stored using a dedicated storage space in SciDB and are replaced using a scalable clock-replacement policy. When new data is inserted into the system, all views overlapping with the new data are selected. Two outcomes can then occur depending on the view: If the view can be updated (i.e., for the spatial and temporal models described above), then our backend updates the view incrementally by inserting the new data only and recomputes parts of the values only. It the view cannot be updated incrementally (i.e., for the noise propagation model), then the view is dropped and will have to be materialized again for future queries.

5 Performance Evaluation

In order to test the validity of our models, we performed series of live deployments using smartphones and mobile sound sources. We report below on a few of such experiments.

5.1 Spatial Interpolation

We tested our interpolation model through 30 outdoor deployments, 10 times using 2 smartphones simultaneously, 10 times with 3 smartphones, and 10 times with 4 smartphones. The location of the smartphones in the deployments were randomly selected in a given flat region of 50 by 50 meters. We chose a relatively busy neighborhood in an urban setting with relatively constant noise levels. We took a professional-grade noise level meter to measure the real values of the sound levels we were trying to interpolate using the smartphones available and our model.

In summary, the results we obtained were as follows: 85% of the resulting interpolated data was with an error of less than 6dB to the real values, with 63% of those values within 4dB or less. We find those results very encouraging given the setting chosen and the few values recorded.

5.2 Sound Dissipation and Source Location

We used a controlled setting to test our sound dissipation and source localization approaches. We picked a 100dB sound source on a 60 by 60 meter baseball field, covered in snow to avoid reverberation. We placed the source in one corner of the field, and then measured the sound levels in the neighboring area using our application and three smartphones. Figure 4 shows the results. Normally, the values should steadily decrease as we get further and further from the source. However, as we can observe on the figure, this is not strictly the case in practice, given that the measurements were taken over several minutes in an open area using different smartphones.

We then tested our sound propagation and source localization models given this relatively noisy input (which is in our opinion close to what the crowd would give us in a larger-scale deployment). We performed 10 tests using 3 random measurements picked from our live deployment, 10 using 4 measurements, and 10 others using 5 measurements.

In summary, the results we obtained are as follows: the error on the sound level value of the source determined by sound dissipation and our smartphone measurements decreases steadily with the number of measurements available:

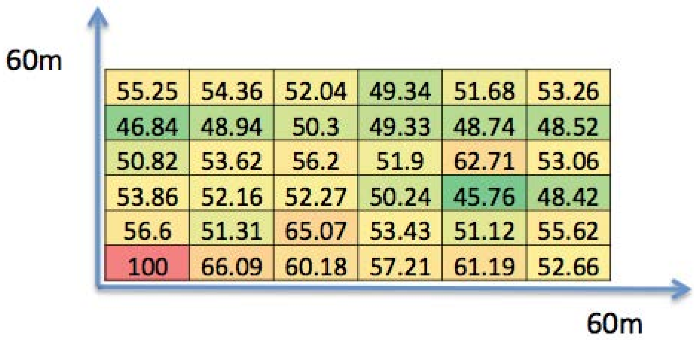


Fig. 4. NoizCrowd live measurements in space using three smartphones and a noise source of 100 db (lower left of the figure)

on average, the error is 16% for 3 available measurements, 10% with 4 measurements, and 9% with 5 measurements. The sound localization performed well too, as we were able to locate the source within a 3 meter radius on average.

6 Conclusions

Generation and management of large-scale data is key for data exploration and decision making. In this paper, we have proposed NoizCrowd: a system to crowdsource noise level data using smartphones. Our system is able to scale out the gathering and management of noise level signals by means of a mobile application and a scalable array back-end. In addition, NoizCrowd autonomously generates missing data by means of interpolation over time and space. Finally, a visual dashboard allows users to query for the noise levels in a specific time and space interval. Our experimentations using real-world deployments of our system have shown that the proposed interpolation models can accurately generate noise level data with noisy input values.

As future work, in addition to porting our application to different mobile platforms (e.g., Android), we plan to extend it with additional functionalities to incentive people to use it. Indeed, using an application just to measure and share the surrounding noise brings little incentives on its own. Therefore, the data recording should be processed in the background while the participant uses the application for other purposes.

Finally, since computing models dynamically at query-time for any user query can be expensive, we plan to explore materialization strategies in more detail in the future. More specifically, we plan to materialize models at various granularities in an offline manner in order to let the users freely zoom in or out on a maps in real-time.

Acknowledgments. This work was supported (in part) by the Swiss National Science Foundation under grant numbers PP00P2_128459 (Infrastructures for Community-Based Data Management) and 200021_130132 (BioMPE).

References

1. Barreiro-Hurl, J., Sanchez, M., Viladrich-Grau, M.: How much are people willing to pay for silence? a contingent valuation study. Applied Economics 37(11), 1233–1246 (2005)
2. Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., Srivastava, M.B.: Participatory sensing. In: Workshop on World-Sensor-Web (WSW 2006): Mobile Device Centric Sensor Networks and Applications, pp. 117–134 (2006)
3. Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musolesi, M., Fodor, K., Ahn, G.-S.: The rise of people-centric sensing. IEEE Internet Computing 12(4), 12–21 (2008)

4. Christin, D., Reinhardt, A., Kanhere, S.S., Hollick, M.: A survey on privacy in mobile participatory sensing applications. Journal of Systems and Software 84(11), 1928–1946 (2011)
5. Cudré-Mauroux, P., Kimura, H., Lim, K.-T., Rogers, J., Simakov, R., Soroush, E., Velikhov, P., Wang, D.L., Balazinska, M., Becla, J., DeWitt, D.J., Heath, B., Maier, D., Madden, S., Patel, J.M., Stonebraker, M., Zdonik, S.B.: A Demonstration of SciDB: A Science-Oriented DBMS. PVLDB 2(2), 1534–1537 (2009)
6. Cuff, D., Hansen, M., Kang, J.: Urban sensing: out of the woods. ACM Communications 51(3), 24–33 (2008)
7. Deng, L., Cox, L.P.: Livecompare: grocery bargain hunting through participatory sensing. In: Proceedings of the 10th Workshop on Mobile Computing Systems and Applications, HotMobile 2009, pp. 4:1–4:6. ACM, New York (2009)
8. DiSalvo, C., Nourbakhsh, I., Holstius, D., Akin, A., Louw, M.: The neighborhood networks project: a case study of critical engagement and creative expression through participatory design. In: Proceedings of the Tenth Anniversary Conference on Participatory Design 2008, PDC 2008, pp. 41–50. Indiana University, Indianapolis (2008)
9. DHondt, E., Stevens, M., Jacobs, A.: Participatory noise mapping works! an evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring. In: Pervasive and Mobile Computing (2012)
10. Ganti, R.K., Pham, N., Ahmadi, H., Nangia, S., Abdelzaher, T.F.: Greengps: a participatory sensing fuel-efficient maps application. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys 2010, pp. 151–164. ACM, New York (2010)
11. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine 49(11), 32–39 (2011)
12. Kanjo, E.: Noisespy: A real-time mobile phone platform for urban noise monitoring and mapping. Mob. Netw. Appl. 15(4), 562–574 (2010)
13. Lamancusa, J.S.: Outdoor sound propagation, pp. 10.6–10.7. Penn State University, PA
14. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. IEEE Communications Magazine 48(9), 140–150 (2010)
15. Lu, H., Pan, W., Lane, N.D., Choudhury, T., Campbell, A.T.: Soundsense: scalable sound sensing for people-centric applications on mobile phones. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, MobiSys 2009, pp. 165–178. ACM, New York (2009)
16. Martí, I.G., Rodríguez, L.E., Benedito, M., Trilles, S., Beltrán, A., Díaz, L., Huerta, J.: Mobile application for noise pollution monitoring through gamification techniques. In: Herrlich, M., Malaka, R., Masuch, M. (eds.) ICEC 2012. LNCS, vol. 7522, pp. 562–571. Springer, Heidelberg (2012)
17. Mendez, D., Labrador, M., Ramachandran, K.: Data interpolation for participatory sensing systems. Pervasive and Mobile Computing 9(1), 132–148 (2013); Special Section: Pervasive Sustainability
18. Moudon, A.V.: Real noise from the urban environment: How ambient community noise affects health and what can be done about it. American Journal of Preventive Medicine 37(2), 167–171 (2009)

19. Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R., Boda, P.: Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, MobiSys 2009, pp. 55–68. ACM, New York (2009)
20. Philipp, D., Stachowiak, J., Alt, P., Dürr, F., Rothermel, K.: DrOPS: Model-Driven Optimization for Public Sensing Systems. In: 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom) (PerCom 2013), pp. 1–8. IEEE Computer Society, San Diego (2013)
21. Rana, R.K., Chou, C.T., Kanhere, S.S., Bulusu, N., Hu, W.: Ear-phone: an end-to-end participatory urban noise mapping system. In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2010, pp. 105–116. ACM, New York (2010)
22. Reddy, S., Estrin, D., Srivastava, M.: Recruitment framework for participatory sensing data collections. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive 2010. LNCS, vol. 6030, pp. 138–155. Springer, Heidelberg (2010)
23. Ruge, L., Altakrouri, B., Schrader, A.: Soundofthecity - continuous noise monitoring for a healthy city. In: 5th International Workshop on Smart Environments and Ambient Intelligence (SENAmI 2013) at IEEE International Conference on Pervasive Computing and Communication (PerCom 2013), San Diego, California, USA, March 18-22 (2013)
24. Schweizer, I., Bärtl, R., Schulz, A., Probst, F., Mühlhäuser, M.: Noisemap - real-time participatory noise maps. In: Second International Workshop on Sensing Applications on Mobile Phones, ACM SenSys 2011 (2011)
25. Seering, A., Cudré-Mauroux, P., Madden, S., Stonebraker, M.: Efficient Versioning for Scientific Array Databases. In: ICDE, pp. 1013–1024. IEEE Computer Society (2012)
26. Wylot, M., Pont, J., Wisniewski, M., Cudré-Mauroux, P.: dipLODocus [RDF]—Short and Long-Tail RDF Analytics for Massive Webs of Data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 778–793. Springer, Heidelberg (2011)

Smarter Mobile Apps through Integrated Natural Language Processing Services

Bahar Sateli[1], Gina Cook[2], and René Witte[1]

[1] Semantic Software Lab, Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada
[2] iLanguage Lab, Montréal, Canada

Abstract. Smartphones are fast becoming ever-present personal assistants. Third-party 'apps' provide users with nearly unlimited customization options. A large amount of content read on these devices is text based – such as emails, web pages, or documents. Natural Language Processing (NLP) can help to make apps smarter, by automatically analyzing the meaning of content and taking appropriate actions on behalf of their users. However, due to its complexity, NLP has yet to find widespread adoption in smartphone or tablet applications. We present a novel way of integrating NLP into Android applications. It is based on a library that can be integrated into any app, allowing it to execute remote NLP pipelines (e.g., for information extraction, summarization, or question-answering) through web service calls. Enabling a separation of concerns, our architecture makes it possible for smartphone developers to make use of any NLP pipeline that has been developed by a language engineer. We demonstrate the applicability of these ideas with our open source Android library, based on the Semantic Assistants framework, and a prototype application 'iForgotWho' that detects names, numbers and organizations in user content and automatically enters them into the contact book.

1 Introduction

The hand-held devices market has never been a quiescent one. With fierce competition among big mobile companies and rapid advancements in the hardware industry, customers are now in possession of mobile devices with relatively large amount of memory and processing resources that are powerful enough for a wide range of tasks, from social networking to document editing and sharing. With fast, ubiquitous Internet connections, smartphone users have access to an ever growing amount of information available in web pages or email boxes. Unfortunately, they are still left alone to deal with this information overload on their usually small-screen devices. While state-of-the-art techniques from the Natural Language Processing (NLP) domain have proven to be useful in dealing with such situations, it is not yet feasible to deploy a complete NLP solution on smartphones, mainly due to its resource-intensive nature. This major limitation prompts the need for novel approaches that can bring NLP power to smartphones, taking into account their limited input and processing capabilities.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 187–202, 2013.

Towards this end, efforts have been made to develop various applications for mobile platforms, such as Google's Android[1] or Apple's iOS,[2] which target specific productivity issues, like retrieving information about places or events with question-answering (QA) applications like Siri[3] or context-sensitive information extraction (IE) like Google Now.[4] However, this means that users must install and use different apps for their various information needs and the market is still lacking a generic NLP app that can offer several text processing capabilities, such as summarization or entity detection, in a unified manner. Preferably, while offering various sophisticated NLP techniques, such an app should be simple enough so that customers, as well as other app developers with no technical NLP background, can make use of its capabilities.

In this paper, we present the first open source NLP library for the Android platform that allows various applications to benefit from arbitrary NLP services through a comprehensive, service-oriented architecture. This work is significant since we offer an application-independent software layer for Android that can be integrated into any existing app in need of NLP support, rather than enhancing a single application. Our approach introduces a novel Human-AI collaboration pattern that can be leveraged to aid mobile users with information-intensive tasks across various domains, such as health care, law, engineering, e-learning, e-business, among others [1–3].

2 Background

In this section, we briefly explain the conceptual and technical foundations for our work, namely, natural language processing and the web service-based Semantic Assistants framework with its NLP API.

2.1 Natural Language Processing (NLP)

The history of Natural Language Processing (NLP) goes far back a few decades. One of the ultimate goals of NLP is to derive intelligence from unstructured content expressed in a natural language, such as English or German, using a variety of techniques from the Artificial Intelligence and Computational Linguistics domains. *Text Mining* is an immensely popular application of NLP that aims at extracting patterns and structured information from textual content. Due to its importance, many frameworks have been developed to facilitate the development of text mining applications, such as the open source *General Architecture for Text Engineering* (GATE) [4], designed both for language experts who need to implement concrete NLP pipelines, as well as software developers seeking to embed NLP capabilities in their applications.

[1] Google Android, `http://www.android.com/`
[2] Apple iOS, `http://www.apple.com/ios/`
[3] Siri, `http://www.apple.com/ios/siri/`
[4] Google Now, `http://www.google.com/landing/now/`

Mobile Applications of NLP. In what follows, we present a number of standard NLP tasks, with a focus on those relevant for mobile applications. However, this list is by no means exhaustive; many other tasks exist, in particular, for domain-specific contexts (e.g., e-health, e-learning, or e-business).

Automatic Summarization. In the presence of mass information, there is a need to allocate the attention of the target efficiently among the overabundance of information sources [5]. Consider a mobile user that quickly needs to get the main content out of a number of emails, documents, or web pages. Browsing through large amounts of textual content on a small-screen device is not only tedious and time-consuming, but important information can be easily overlooked. In such a situation, where a user's information need is dispersed over a single long or multiple documents, NLP techniques can provide him with a *summary*, a compressed version of the original document(s) that preserves the main information as good as possible. While *generic summaries* are probably the most widely known, other summary types are even more interesting for mobile applications: *Focused summaries* start from a topic (such as a question stated by the user) and generate the summary targeted at this question. *Update summaries* take the user's reading history into account, summarizing only the new information that was not seen before [6].

Information Extraction (IE) is one of the most popular applications of NLP. IE identifies instances of a particular class of events, entities or relationships in a natural language text and creates a structured representation of the discovered information. A number of systems and APIs have been developed for this task, such as OpenCalais,[5] which can extract named entities like *Persons* and *Organizations* within a text and present them in a structured language, e.g., XML or RDF.[6] These techniques are also helpful for mobile applications, where users often deal with large amounts of textual content. For example, using an IE service in a mobile app can help users to automatically find all the occurrences of a specific type of entity, such as 'company', and gather complementary information in form of metadata around them.

Content Development. NLP techniques can also be helpful when developing new content: Typing (or dictating) longer texts on a mobile device is typically cumbersome and time-consuming. Here, NLP services such as summarization or IE can support users by generating part of the new content. For example, when replying to an email in response to a question, a focused summary can provide the main body of information that needs only editing and verification by a user.

Question Answering (QA) has become one of the most prominent applications of NLP on mobile devices, in particular due to Apple's Siri app. In contrast to focused summaries, which also address an information need of a user, but are typically answering open-ended questions in an essay-style (e.g., *"What is the importance of refrigeration for ice cream deliveries?"*), QA aims at providing answers to factual, closed questions (e.g., *"Where do I find the nearest ice cream*

[5] OpenCalais, `http://www.opencalais.com/`

[6] Resource Description Framework (RDF), `http://www.w3.org/RDF/`

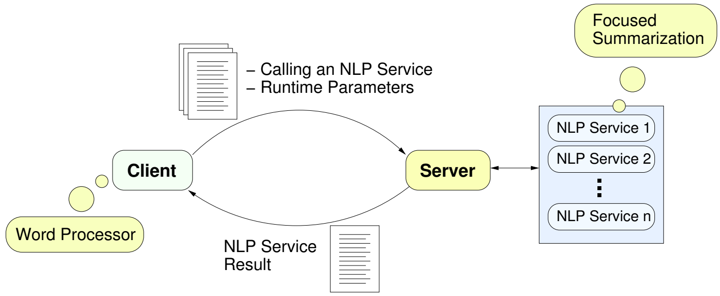


Fig. 1. The Semantic Assistants service execution workflow [7]

parlor?"). QA is a significant improvement to simple (keyword-based) information retrieval, which simply presents a list of possibly relevant documents to the user, because it allows users to pose questions against its knowledge base in natural language. Then, after "understanding" the question, an answer formulation step brings back the extracted information to the user. While systems like Siri apply to a general domain, QA systems on mobile devices have much broader potential: For example, a company might want to develop an app that can answer questions about its products or services; or a university could provide students and staff with an app that is capable of answering questions about its courses or buildings.

2.2 The Semantic Assistants Framework

As we described in the previous section, NLP pipelines are diverse in nature and in some scenarios, more than one NLP technique might be useful in respect to the user's task at hand. The Semantic Assistants framework [7] is an existing open source architecture that provides a unified manner of offering NLP capabilities to clients, ranging from desktop applications to web information systems, in form of W3C standard web services.[7] The goal of the Semantic Assistants architecture is to wrap concrete analysis pipelines, developed based on existing NLP frameworks, and broker them to connected clients through a service-oriented architecture.

Corresponding to the definition of service-oriented architectures, the Semantic Assistants architecture has a repository of NLP pipelines that are formally described using the Web Ontology Language (OWL).[8] Such an infrastructure allows for dynamic discovery of NLP services in the architecture and reasoning capabilities over pipelines before recommending them to clients. For example, based on the user's context or the language of source content, Semantic Assistants can recommend a subset of NLP services to the user. This way, any NLP service deployed in the Semantic Assistants repository becomes available to all connected clients through a WSDL[9] interface description.

[7] Web Services Architecture, `http://www.w3.org/TR/ws-arch/`
[8] Web Ontology Language (OWL), `http://www.w3.org/2004/OWL/`
[9] Web Services Description Language (WSDL), `http://www.w3.org/TR/wsdl`

As shown in Fig. 1, an NLP execution workflow is initiated by the client through sending textual content (e.g., the URL of a document or literal text) to a Semantic Assistants server. Optionally, clients can customize the pipelines' behaviour by setting a number of runtime parameters, e.g., output formats or special thresholds. These requests trigger the execution of an actual NLP pipeline on the provided content, and if successful, return the results to the client in form of a unified XML document. The receiving client is then responsible for parsing the results and providing the user with a proper presentation, for instance, highlighting annotations in a text or opening up a new document.

In order to facilitate the integration of new clients, the Semantic Assistants architecture offers a Client-Side Abstraction Layer (CSAL) library. Essentially, the CSAL library is a Java archive of common communication and data transformation functionalities that can be reused by different clients. Such an abstraction layer eases the implementation of the client-server communication process in new clients, as well as the transformation of NLP results to other useful data types.

What distinguishes the Semantic Assistants framework from other approaches is its emphasis on a *separation of concerns*: Users who interact with the NLP pipelines are not concerned with how the analysis is performed on their provided content; Application developers can use the CSAL library to easily connect to the back-end server and invoke services and retrieve the results; and Language engineers can develop sophisticated NLP pipelines without worrying about how they are going to be offered to the end user. Currently, the Semantic Assistants architecture supports NLP pipelines developed based on the GATE[10] framework, but also provides for the execution of pipelines based on OpenNLP[11] and UIMA[12] through their GATE integration.

3 Design of the Semantic Assistants Android NLP Layer

Following the description of how different NLP techniques can aid mobile users and the Semantic Assistants framework that provides such capabilities to its connected clients, we now describe a high-level design overview of our novel Android-NLP integration process. The ultimate goal we are trying to achieve in this integration is to provide a software layer to the Android platform that can be used by various applications in need of NLP support.

Our research hypothesis is that numerous applications running on a smartphone can benefit from natural language processing support: interactions with the user will become faster through automation; tasks that are currently difficult to perform on small-screen devices will be significantly improved.

Robust, open source implementations for tasks like information extraction or automatic summarization are readily available, in the form of NLP pipelines

[10] GATE, `http://gate.ac.uk/`
[11] OpenNLP, `http://opennlp.sourceforge.net`
[12] UIMA, `https://uima.apache.org/`

running within a text analysis framework. However, these frameworks cannot be executed on an Android device, as NLP is very resource-intensive and mobile devices have both hard- and software limitations [8]. Furthermore, rather than just enhancing a single application with NLP capabilities, we aim to offer an application-independent software layer that can be integrated into any existing application requiring NLP support. Similarly, we do not want to restrict the type and number of possible NLP analysis services in advance: rather, it should be possible to dynamically add new services and have them discovered by the mobile device, i.e., we aim to design a service-oriented architecture (SOA). The vision is that applications on a mobile device can dynamically request NLP services, such as entity extraction, summarization [9, 10], or question-answering [11], to support the user in knowledge-intensive tasks, just like a human assistant would.

3.1 Requirements

We start by defining a set of requirements for our integration.

Remote Execution of Services (R1). In order to tackle the limited memory and processing capacities of mobile devices, the integration must execute the NLP services on a remote machine and bring the results back to the mobile device.

NLP Service Independence (R2). Mobile users deal with various types of information, ranging from news articles to personal documents, for which generic or domain-specific NLP pipelines may be useful. Irrespective of how the NLP pipelines are concretely implemented, the integration must offer them within a single unique interface.

App Independence (R3). The integration must be realized in such a way that various apps can seamlessly benefit from NLP services, rather than enhancing a single app only.

Flexible Response Handling (R4). Depending on the application area, NLP pipelines can generate various output formats, e.g., annotations for IE or new documents for summarization. The integration must accordingly represent and transform the NLP pipeline results in a mobile device.

Inter-App Communication (R5). Various apps must be able to share content with the integration layer for analysis and receive the results, where applicable. It should also be possible to receive content from one app and write the results to another.

3.2 Developed Solution

Here, we describe our design decisions to meet the requirements enumerated in the previous section. Our goal is to design an architecture that allows Android apps to connect to a remote Semantic Assistants server and benefit from its NLP services, either directly within their environment or through another *service app*, i.e., an application that performs the invocation process and returns the results to the app that originally requested the NLP service.

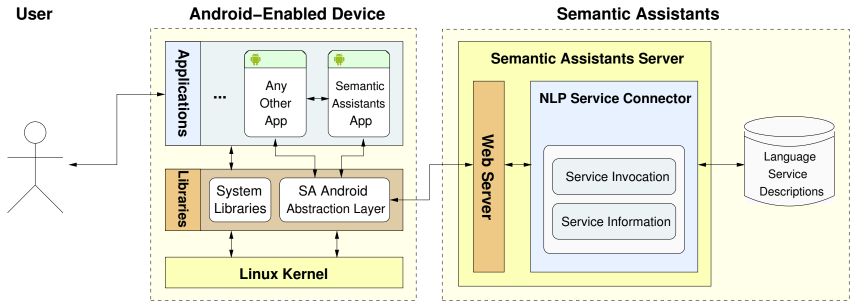


Fig. 2. The Semantic Assistants-Android integration high-level architecture

Fig. 2 illustrates a high-level overview of our integration architecture, where the client is always an Android app requesting an NLP service, and the server side is the Semantic Assistants server brokering these pipelines as Web services (R1). The client-server communication is facilitated through an Android-specific client-side abstraction layer, designed as an extension to the Semantic Assistants framework. Because of the absence of Simple Object Access Protocol (SOAP) [12] libraries in Android, both components in our architecture have to communicate in a RESTful way. To provide a RESTful communication over the HTTP protocol, we extended the Semantic Assistants server tier with a REpresentational State Transfer (REST) [13] interface that additionally provides a secure HTTPS channel.

The Semantic Assistants Android abstraction layer encompasses common functionalities for Android apps to communicate with the Semantic Assistants server by creating RESTful requests for service inquiry, execution, user authentication and eventually, parsing the retrieved results. It also contains pre-defined Android *intents*[13] for remote execution of NLP services. In the Android platform, intents are system-wide messages that contain abstract descriptions of operations to be performed. Intents are the Android's platform facility for communication between components within one or multiple applications. The Semantic Assistants Android abstraction layer library has an extensible structure for NLP intents that can be extended as more frequently-used NLP services are made available in the Semantic Assistants repository. For example, the Semantic Assistants Android abstraction layer has a built-in entity detection intent that extracts *Person* and *Location* named entities from a given text and returns the extracted results as series of annotations to the invoking application. We will demonstrate a use case for this intent within a demo app in Section 4.

4 Implementation

Our two novel components in the Android-NLP integration are the *Semantic Assistants App* and the *Semantic Assistants Android Abstraction Layer* library

[13] Intents and Intent Filters, `http://developer.android.com/guide/components/intents-filters.html`

shown in Fig. 2. The Android abstraction layer plays the role of a system-wide library that can be referenced by all Android apps installed on the application layer of the Android OS architecture. The Semantic Assistants App[14] is an implementation of a general-purpose NLP app that can offer various NLP services to users both within its user interface, as well as other apps using system messages.

4.1 The Semantic Assistants Android Abstraction Layer

Technically, the Semantic Assistants Android abstraction layer is an Android library[15] project that holds shared code and resources for apps that need NLP support in their workflow (R3). The abstraction layer classes implementing the SA-Android communication and data transformation are compiled by the Android development tools into a Java archive (JAR) file and installed on the system. Thereby, various apps can refer to the Semantic Assistants abstraction layer as a *library* dependency in their code. Using the library classes, other apps can directly connect to a given Semantic Assistants server interface and invoke the NLP services available in the server's repository without the hassle of implementing connection handlers to a remote server. The library also helps developers in preparing requests conforming to the Semantic Assistants server endpoint description file and parsing the resulting XML document. The current implementation of the Semantic Assistants Android library supports NLP service inquiry, invocation, and user authentication on the Semantic Assistants server over HTTP and HTTPS protocols.

4.2 The Semantic Assistants App

As a part of our contribution and in order to demonstrate a general-purpose app offering arbitrary NLP services to Android mobile users, we have implemented an Android app, called the *Semantic Assistants App*, that offers a unique user interface to inquire and invoke NLP services on a user-provided content. The Semantic Assistants App's main activity allows users to authenticate themselves and configure the app to connect to a specific Semantic Assistants server endpoint. Once the user settings are stored, the Semantic Assistants App inquires about the available assistants from the server and generates a dynamic list containing service names (R2), their descriptions and possible further configuration options, such as applicable runtime parameters, as shown in the top part of Fig. 3. When the user selects a service from the list, the Semantic Assistants App features a text field for the user to enter content for analysis. For example, in our screenshot taken from a tablet running the Semantic Assistants App, we have provided a short sentence for a sample analysis scenario and chosen the "Person and Location Extractor" as the NLP service. Pressing the "Invoke" button

[14] Semantic Assistants App is available at `http://www.semanticsoftware.info/sa-android`

[15] Android Library Project, `http://developer.android.com/tools/projects/index.html`

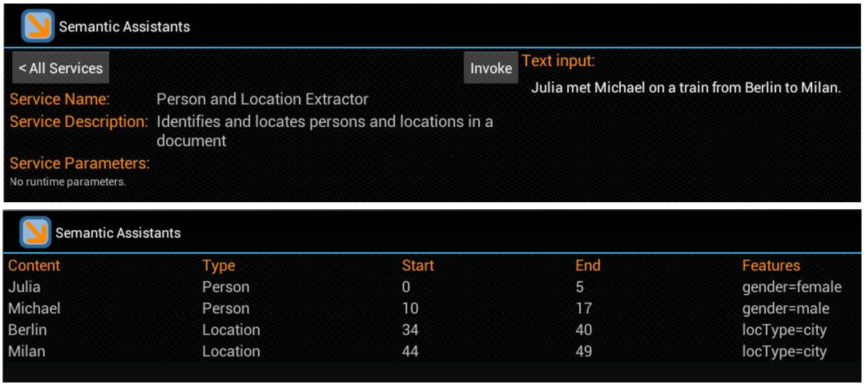


Fig. 3. The Semantic Assistants App Service User Interface

sends a request to the remote Semantic Assistants server through the Semantic Assistants Android abstraction library, which in turn triggers the execution of the actual pipeline on the server. The Person and Location entities found in the text are first received by the Semantic Assistants Android abstraction library in form a typical Semantic Assistants XML document. The library then parses the embedded entities to an array of Java objects and sends them back to the Semantic Assistants App, where a list is automatically populated from the entities' literal content, their semantic types, their exact character offsets in the original text, as well as any additional features provided by the pipeline (Fig. 3, bottom).

Of course, manually pasting content into the Semantic Assistants App's text field is still not convenient for most mobile users, especially on small screens. That's why the Semantic Assistants App also *listens* to the system's *sharing* intents, broadcasted from other existing apps on the device (R5). Sharing intents are fired in the system whenever an app provides the user with the ability to share content directly with another app, e.g., sending a paragraph from a web page to an email application as a newly composed message. Because the Semantic Assistants App registers itself in the Android system as a listener component for sharing intents, its name pops up in the contextual menu of sharing actions where applicable. Hence, users can choose the Semantic Assistants App as the content receiver application and thereby automatically populate the text field of the app, preparing it for a new NLP analysis session. Based on the NLP pipeline's result format, the Semantic Assistants App can eventually present the extracted annotations in a list format, or open up the generated output file in the Android's configured default browser (R4).

Listening to system-wide sharing intents provides Semantic Assistants App users with a convenient method to invoke arbitrary NLP pipelines using the app's graphical user interface. However, another important target user group of our integration are Android app developers who need to embed NLP capabilities in their own application. In order to suppress the need to interact with the Semantic

```
<service
  android:name="info.semanticsoftware.semassist.android.service.
      SemanticAssistantsService"
  android:process=":semassist_service"
  android:label="semassist">
    <intent-filter android:label="Semantic Assistants Open Intents">
      <action android:name="org.openintents.action.PERSON_LOCATION_EXTRACTOR" />
      <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</service>
```

Fig. 4. A Semantic Assistants open intent example

Assistants App's GUI for each analysis session, the Semantic Assistants App is designed to offer its "*Semantic Assistants Open Intents*" to all applications, once it has been installed on a device. The idea behind open intents is to allow other apps to invoke specific NLP services in a headless manner by sending a broadcast message across the system asking for a specific intent. If the intent is recognized by the Semantic Assistants App, it is received and transformed into a corresponding NLP service execution request. Subsequently, the results of a successful execution are returned to the app that sent the broadcast message. Fig. 4 shows an example Semantic Assistants open intent to extract person and location named entities from a given text.

5 Application

Following a thorough description of the Android-NLP integration, in this section we explain how Android app developers can now embed NLP capabilities in their own apps using our novel architecture. We begin this section by an example scenario, in which an app developer wants to write a *smart* app for his client. Our developer's client frequently travels to different conferences and meets new people. He tries to remember and keep in touch with his new connections by creating contact entries in his Android phone. However, every once in a while, he goes back to his contact book, though by just reading a name he doesn't remember in which context he met that specific person. So he asks our developer to create an app for him that can automatically extract useful metadata and related context information from the emails his connections send him and create new contacts or add notes to his contact book entries.

In a typical Android development process, our developer starts by designing an Android app that receives shared content from an email app on the device. Once he receives the message content in his app, he has to apply various heuristics, such as regular expressions or string matching, on the text to find useful entities for his client. Such an approach is computationally expensive and does not usually yield an acceptable precision or recall when implemented by someone unfamiliar with the text mining domain. Therefore, after a short research, our developer finds a

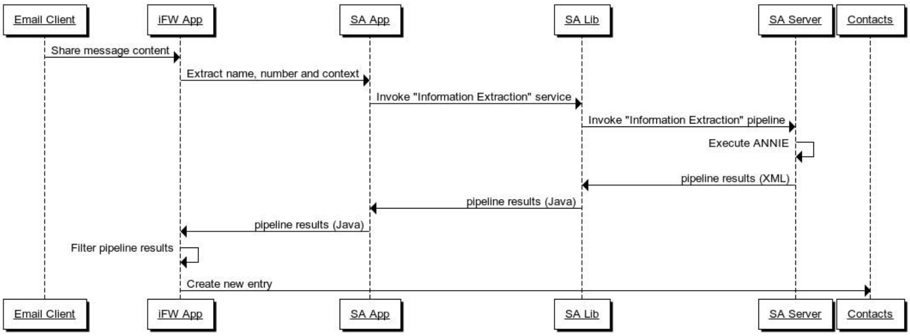


Fig. 5. The iFW app workflow sequence diagram

```
1  <!-- Transparent activity bound to the system share intent -->
2  <activity android:name=".activity.ContactFinderActivity" android:theme="@style/Theme.
       Transparent">
3          <intent-filter android:label="iForgotWho">
4                  <action android:name="android.intent.action.SEND" />
5                  <category android:name="android.intent.category.DEFAULT" />
6                  <data android:mimeType="text/*"/>
7          </intent-filter>
8  </activity>
9
10 <!-- Broadcast receiver for the Semantic Assistants App messages -->
11 <receiver android:name=".services.NotifBroadcastReceiver">
12         <intent-filter>
13                 <action android:name="info.semanticsoftware.semassist.android.BROADCAST"/>
14         </intent-filter>
15 </receiver>
```

Fig. 6. The iForgotWho app manifest file snippet

Semantic Assistants open intent that can extract various entities, such as person, organization, location and address entities, from a given text. The developer then embeds the Semantic Assistants Android abstraction library in his project and delegates the entity extraction responsibility to the Semantic Assistants App via a simple Java method call, provided by the abstraction library classes. In this case, the processing flow of this new application, called the 'iForgotWho' app, follows the sequence shown in Fig. 5.

To better demonstrate this use case, we implemented the iForgotWho (iFW) Android app and used its NLP capability on an example email message. The iFW app's manifest file[16] (Fig. 6) denotes two important features about this app: The iFW app accepts textual content shared from other apps (lines 1–8) and listens to broadcast messages sent from the Semantic Assistants App (lines 10–15).

[16] The AndroidManifest.xml File, `http://developer.android.com/guide/topics/manifest/manifest-intro.html`

```java
public class ContactFinderActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent service = new Intent("org.openintents.action.INFORMATION_EXTRACTOR");
        /*The share intent carries the user selected text.
         * We simply pass the text to the designated Semantic Assistants service. */
        String input = getIntent().getExtras().getString(Intent.EXTRA_TEXT);
        service.putExtra(Intent.EXTRA_TEXT, input);
        /* True silent mode means that iForgotWho app
         * will take care of the results presentation. */
        service.putExtra(Constants.SILENT_MODE, "true");
        // call the service
        startService(service);
        // close the activity
        finish();
    }
}
```

Fig. 7. The iForgotWho activity requesting an open intent

The iFW app contains only two activity classes[17] – one transparent activity that delegates user requests to the Semantic Assistants Android library and another one that lists the extracted entities and asks for the user's permission before adding them to his contact book. Fig. 7 contains the complete source code of the iFW's service-requesting activity, showing how the complicated task of entity detection can be simply achieved through using a Semantic Assistants open intent.

In our example scenario, our developer's client uses the iFW information extraction feature to automatically create a new contact entry from an email message he has received from one of his connections. The client selects the content of his email message and shares it with the iFW app. The iFW app then calls the `INFORMATION_EXTRACTOR` service of the Semantic Assistants App to extract useful information, such as the person's name or phone number from the email message, through executing GATE's ANNIE pipeline [4] on the server side. Upon receiving the results, iFW prompts the user with the detected entities and lets our client verify the results. Once approved by the user, iFW automatically creates a new contact entry and populates the corresponding fields, such as the contact's name, phone number and a sentence from the (email) message containing a named entity, like the sender's name, as contextual information to help our user remember the contact. Fig. 8 shows the sharing process and the contact entry that is eventually created by the iFW app in the user's contact book.

As we showed in our iFW app, our Android-NLP integration architecture provides the possibility for developers to enhance their applications' capabilities with state-of-the-art techniques from the natural language processing domain, implementing novel use cases of human-AI collaboration patterns within the context of mobile devices at the convenience of calling open intents.

[17] Activity classes provide user interface components that users interact with.

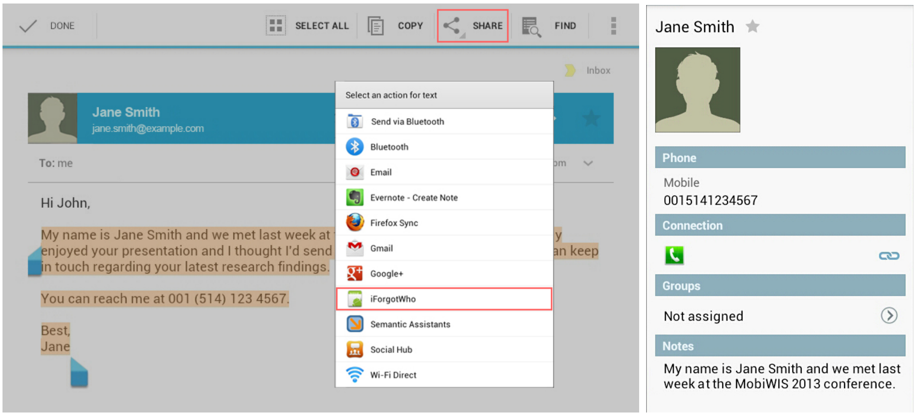


Fig. 8. Sharing content with the iFW app (left) and the resulting contact entry (right)

6 Related Work

Using natural language processing techniques in the context of mobile devices has recently gained attention, both among academic and industry communities. The lavish launch of Apple's Siri as a voice-controlled personal assistant was a major demonstration of how state-of-the-art NLP techniques can aid mobile users with productivity tasks, like finding a nearby restaurant. This has motivated other frontier mobile development companies to increase their efforts in adopting NLP techniques for mobile devices. Google Now integrates additional techniques, such as personalization, on top of NLP. However, many of these applications are still largely limited to "speak to command" use cases, which merely provides a natural language interface for users to issue spoken, intuitive commands to a mobile device for tasks like question-answering or sending text messages in a hands-free manner [14]. Moving beyond speech recognition, the apparent need for other more complex tasks requiring sophisticated NLP analysis is being satisfied through various apps for information-intensive areas like personalized news recommendation [15] or web page summarization [9]. Unlike these rapidly emerging task-specific apps, our mobile-NLP integration aims at providing a general NLP solution that can offer various techniques through a unified interface. On the foundations of the Semantic Assistants' service-oriented architecture, a multitude of generic and domain-specific NLP pipelines can be deployed and utilized within Android apps. In addition, in contrast to [16] and [17], where authors try to adapt resource-intensive NLP techniques like information extraction and machine translation to mobile contexts, our integration architecture offers the full power of NLP applications to mobile users.

Alongside the efforts to develop NLP-enabled apps that target specific productivity issues in mobile contexts, others are providing more general solutions to embed NLP capabilities within mobile apps by providing software APIs.

AlchemyAPI[18] offers an Android SDK for app developers to make use of its various text mining techniques, such as keyword extraction or concept tagging, within their apps through embedding a Java archive file that makes remote web service calls to the AlchemyAPI text mining platform. Maluuba[19] is another NLP API for mobile apps that offers two separate interfaces for "interpreting" spoken commands and "normalizing" textual content for Android and Windows Phone devices. In contrast, our approach not only provides a unique interface to offer various NLP techniques, but also provides ready-to-use NLP *intents*. This is an advantage over merely publishing an API that first needs to be understood by developers and then employed in their apps. Moreover, unlike these commercial offerings, both the Semantic Assistants framework and our Semantic Assistants-Android integration are open source software[20] that allow the developers to deploy their own custom NLP pipelines (e.g., for healthcare, biomedical research, e-learning, or entertainment) and extend our integration's open intents, as their need arises. Finally, by providing a clear separation of concerns, mobile developers with no NLP background can now make use of a multitude of existing open source pipelines, in particular those developed based on the popular GATE framework, as well as custom pipelines that language engineers will develop without the concern on how these pipelines are going to be integrated in mobile apps.

7 Conclusion

For many users, smartphones have become their go-to-device for industry news feeds, checking email, and scheduling their agenda. The sheer quantity of text which is read on these devices, and the number of screen-taps needed to get things done, could be significantly reduced by applying existing Natural Language Processing (NLP) pipelines, such as automatic summarization or information extraction, to news feeds, emails, or attachments. Summarized text can also be consumed in an eyes-free manner using the smartphone's Text-To-Speech capabilities. Dates, locations and people can be automatically detected using named entity recognition and integrated in the creation of new events in a user's agenda and entries in the contact book as we demonstrated with the iFW app. We developed the first open source API for Android that provides an easy way to integrate NLP capabilities into an application. It relies on interactions familiar to mobile application developers, without requiring any background in computational linguistics. With our framework, application developers can easily integrate complex text mining tasks into a smartphone application, either based on existing open source NLP tools, or by delegating the pipeline development to a language engineer. We believe that the resulting architecture has significant potentials to make mobile apps 'smarter' across a wide range of domains.

[18] AlchemyAPI, `http://www.alchemyapi.com`
[19] Maluuba, `http://www.maluuba.com/`
[20] Available at `http://sourceforge.net/projects/semantic-assist/`

References

1. Aanensen, D.M., Huntley, D.M., Feil, E.J., al Own, F., Spratt, B.G.: EpiCollect: Linking Smartphones to Web Applications for Epidemiology, Ecology and Community Data Collection. PLoS ONE 4(9), e6968 (2009)
2. Doukas, C., Pliakas, T., Maglogiannis, I.: Mobile healthcare information management utilizing Cloud Computing and Android OS. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), September 4, pp. 1037–1040 (2010)
3. Kamel Boulos, M.N., Wheeler, S., Tavares, C., Jones, R.: How smartphones are changing the face of mobile and participatory healthcare: An overview, with example from eCAALYX. Biomedical Engineering Online 10(1), 1–24 (2011)
4. Cunningham, H., et al.: Text Processing with GATE (Version 6). University of Sheffield, Deptartment of Computer Science (2011)
5. Simon, H.A.: Designing organizations for an information rich world. In: Greenberger, M. (ed.) Computers, Communications, and the Public Interest, pp. 37–72. The Johns Hopkins Press (1971)
6. Witte, R., Bergler, S.: Next-Generation Summarization: Contrastive, Focused, and Update Summaries. In: International Conference on Recent Advances in Natural Language Processing (RANLP 2007), Borovets, Bulgaria, September 27-29 (2007)
7. Witte, R., Gitzinger, T.: Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In: Domingue, J., Anutariya, C. (eds.) ASWC 2008. LNCS, vol. 5367, pp. 360–374. Springer, Heidelberg (2008)
8. Park, S.Y., Byun, J., Rim, H.C., Lee, D.G., Lim, H.: Natural language-based user interface for mobile devices with limited resources. IEEE Transactions on Consumer Electronics 56(4), 2086–2092 (2010)
9. Alam, H., Hartono, R., Kumar, A., Rahman, F., Tarnikova, Y., Wilcox, C.: Web Page Summarization for Handheld Devices: A Natural Language Approach. In: 7th International Conference on Document Analysis and Recognition (ICDAR 2003), vol. 2, pp. 1153–1157 (2003)
10. Buyukkokten, O., Garcia-Molina, H., Paepcke, A.: Seeing the whole in parts: text summarization for web browsing on handheld devices. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 652–662. ACM, New York (2001)
11. Jilani, A.: Mobile Phone Text Processing and Question-Answering. In: Future Technologies in Computing and Engineering: Proceedings of Computing and Engineering Annual Researchers' Conference 2010: CEARC 2010, pp. 130–135 (2010)
12. Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S., Winer, D.: Simple Object Access Protocol (SOAP) 1.1. W3C Note, World Wide Web Consortium (May 2000), `http://www.w3.org/TR/SOAP/`
13. Fielding, R.T.: Architectural styles and the design of network-based software architectures. PhD thesis (2000)
14. Zhou, L., Shaikh, M., Zhang, D.: Natural Language Interface to Mobile Devices. In: Shi, Z., He, Q. (eds.) Intelligent Information Processing II. IFIP, vol. 163, pp. 283–286. Springer US (2005)
15. Tavakolifard, M., Gulla, J.A., Almeroth, K., Ingvaldsen, J.E., Nygreen, G., Berg, E.: Tailored News in the Palm of Your HAND: A Multi-Perspective Transparent Approach to News Recommendation. In: Proceedings of the 22nd International World Wide Web Conference, WWW 2013, May 13–17, Rio de Janeiro, Brazil (2013)

16. Seon, C.N., Kim, H., Seo, J.: Information extraction using finite state automata and syllable n-grams in a mobile environment. In: Proceedings of the ACL 2008: HLT Workshop on Mobile Language Processing, Columbus, Ohio, pp. 13–18. Association for Computational Linguistics (June 2008)
17. Homola, P.: A Distributed Database for Mobile NLP Applications. In: Proceedings of the ACL 2008: HLT Workshop on Mobile Language Processing, Columbus, Ohio, pp. 27–28. Association for Computational Linguistics (June 2008)

Towards Intelligent Migration of User Interfaces

Giuseppe Ghiani[1], Jussi Polet[2], and Ville Antila[2]

[1] ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy
giuseppe.ghiani@isti.cnr.it
[2] VTT Technical Research Centre of Finland, Kaitoväylä 1, 90571 Oulu, Finland
{jussi.polet,ville.antila}@vtt.fi

Abstract. The multitude of interactive devices we use daily has steadily increased since the advent of personal computers. Also, the spreading ubiquitous computing leaves the users with increasing number of device ensembles. There is often a need to change the device in use, which requires moving applications, data, user interfaces or parts of them to other device(s) and back. This is the case, when something is manipulated on one device and a need to migrate it towards another device exists as well. In order to study the problems of such migration functionality, we defined, designed and implemented a proof-of-concept prototype for automatic context-aware migration of Web applications between devices. The prototype was evaluated in three distinct modes (manual, assisted and automatic) with a user study to collect technical data and user feedback. The results highlighted interesting correlations between the system behaviour and user ratings, and statistically relevant differences on how users perceived the proposed modes of the system.

Keywords: Context-Awareness, Device Ensembles, Migration, Migratory User Interfaces, Multi-device Environments.

1 Introduction

People are increasingly accessing applications and services through diverse types of interactive devices, depending on their current location and needs. The users are facing different kinds of device ensembles at home, work as well as on the move. In addition to providing ways and means for moving and synchronising information, whether that is applications, plain data or user interfaces (UI) between all these devices, it is also challenging to *"keep your digital identity in sync"* [20] – contact lists, media libraries, *Facebook* credentials and so forth need to be in sync and accessible on several different devices. The task of keeping each device up to date with the latest applications and data can be sometimes tedious. Therefore, data and applications are increasingly in the cloud, keeping everything in one place, accessible by all devices.

Also, Web applications are a typical example of applications accessed in different places and with heterogeneous devices, as the actual application and user data is always located on the backend server(s) (i.e. cloud). Documents, music, photos etc. are all accessible from the application front-end that is run with each device's Web browser (or with a dedicated native application that displays the Web content).

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 203–217, 2013.

In this paper, we take a look at the possibilities for migrating Web application front-ends (i.e. the Web page and its state information) between devices. The actual application back-end and data will always be accessed from the original application server. Moreover, in this paper we focus on the context-awareness of such system and the possibilities it could give for enhancing the interaction between the system and the user. Specifically, we take a look at three different ways of interacting with the system: *automatic*, *assisted* and *manual*. Two of these are context-aware: automatic and assisted. With automatic mode, when the system senses a predefined trigger for migration, it performs this action automatically. With assisted mode on the other hand, when sensing a trigger for migration, the system notifies the user of this possibility, leaving the final decision to migrate to the user.

The prototype environment for context-aware migration is an extension of a previously developed multi-device and multi-user Web-based *Migration Platform*. Earlier studies with that platform [11] highlighted a need for intelligent, context-aware and more automatic migration of user interfaces. To address these needs we have implemented and evaluated a context-aware system for Web application migration. The solution presented in this paper is especially targeted towards the needs of task continuity and continuous movement, characterising today's technological multi-device environments. In addition, the added context-awareness eases migration triggering and thus minimises user effort. Our early results indicate that adding context-awareness into the migration process is a notable benefit for the user. We have also discovered important correlations between the technical qualities of the system to the perceived usability and user experience.

In the next sections, we will first present some related research on multi-device interaction and interaction techniques and discuss the benefits of context-awareness in multi-device environments. Then, we describe the architecture and implementation of our prototype system and the user study setting with the test scenarios. After that, we present our evaluation methodology, analyse and discuss the user study results and draw some final remarks useful to better identify the need for such a concept and system in the near future.

2 Related Work

Users' behaviour towards the usage of multiple kinds of devices has recently been considered in a *Google* research report [12], which identifies two major ways of interaction with multiple devices: *sequential* and *simultaneous*. Our approach is more sequential, though by migrating only part(s) of the Web application to a target device by keeping them active also into the source device, even simultaneous workflow can be achieved.

Multi-device interaction has also been investigated in the past, focusing on techniques people use to access multiple devices: *Dearman and Pierce* [7] found that one of the main challenges is how to support *seamless device change*. This is the core problem that we are aiming to tackle by providing an integrated, context-aware functionality for device changes.

One such solution for sharing information and session during device changes in multi-device environments is *Pick-and-Drop* [17], which was a first interesting solution to support the dynamic graphical selection of application elements in one device and easily move them to another device. However, that approach was mainly limited to move data across devices, while people would like also to move entire interactive applications. *Berry et al.* [2] deal with view sharing in cooperative work environments, and tackle the privacy issues of a presenter which wants to show/hide parts of the view according to the type of audience. The solution focuses on collaborative tasks, while in this paper we focus on single-user scenarios in which our platform can be useful.

Support for migratory UIs aim to improve user access to information through multiple devices. In this regard, *Wäljas et al.* [24] have also studied how people access various multi-device applications during a few weeks, and have proposed an initial framework for analysing cross-platform service user experience (UX), which relates to three main aspects: *composition*, *continuity* (fluency of the content and task migration), and *consistency*. Our approach pays attention to all these aspects by allowing the user to either migrate the whole UI or just parts of it (composition) between devices, preserving the continuity of the UX by allowing seamless device changes and aiming for a consistent UX by focusing on the Web application domain. The *eLabBench* [23] relies on an infrastructure for distributing data across laboratory devices and biologists personal computers. The focus is on ubiquitous management of digital data, rather than on UI migration. The approach is thus different from ours, but the final aim is still to support task continuity of nomadic users coping with more than one device in different contexts. Their Fluid Computing middleware [3] was also about multi-device interaction. However, it differs from our Platform both on aims and on implementation: Fluid Computing is aimed at interaction synchronization when multiple users share the same interface, or parts of it, while we do not specifically deal with interface replication.

Deep Shot [5] is a solution for automatic migration of UIs across devices with state preservation. Migration of an interface, or part of it, is triggered by "shooting" it with a mobile device camera. In our platform, migration can be also triggered in a fully automated way without requiring explicit user interaction. Also, the authors of Deep Shot state that this approach is compatible even with native applications, but that existing ones need to be modified in order to support the *"deep shooting/posting"*. Our platform for Web migration instead relies on a proxy that injects all the needed support to the navigated pages, and does not require any modification to existing applications. Also, browser plugins are not needed for enabling migration. The only additional software to enable context-based migration is the *Context Monitor* (which is platform-dependent but compatible with consumer devices). The Context Monitor, described in the following sections, is an extension of an already existing platform for *"pushing/pulling"* of Web applications across devices [11]. Pushing and pulling issues were also investigated in [8], whilst our present work considers only the migration pushing i.e. migrations originated from the device in use. Other aspects of the platform that we consider in this paper have been already tackled: security/privacy is discussed in [10].

The *RELATE system* [9] supports user interaction with devices available in the surrounding. Specific hardware equipping the devices in the environment allows them to discover each other and to determine their relative position. Rather than on specific localization capabilities, in this study we aim to investigate how usability perception varies according to the system behavior.

Sensors embedded in the device or deployed in the environment provide low level data. As indicated in [4], such data can be used by means of evolving situation models to infer high level situations in which the users are involved. Our basic idea matches with the author's one, as we believe that context-awareness would enhance nomadic users' multi device experience.

Enabling technologies for communication, such as *near field communication* (NFC), *radio frequency identification* (RFID) and *Bluetooth*, can be used to sense the co-location of target devices in the proximity of the user [6,18]. Moreover, different characteristics of proximity in interacting in ubiquitous computing environments have been identified [13]. There has been research on techniques that facilitate interaction, taking into account the proximity factors between people and computing devices, such as location (e.g. room or building), position (close to a display), movement (coming towards a display) and orientation (facing towards the display). These factors allow better contextual knowledge about the current situation, so that the system can determine, for instance, whether the public display should facilitate interaction with someone or just display information [1,15].

In this paper, we do not tackle issues of difference in interface rendering due to device diversity, nor possible adaptations of the interface to the destination device (e.g., rearranging of presentation components), which are relevant but would require a dedicated paper. We instead demonstrate the ability to use a simple sensing application as an enabler of context-awareness in a multi-device environment. The system we propose is able to automatically detect some context variables, such as user location, device position and information about the user's current activity (e.g. walking or still). We discuss the possibilities of using this context information to allow the system to perform automatic or semi-automatic (assisted) migration of session from a device to another. This type of automatic trigger can be related to the work on implicit *human-computer interaction* (HCI) driven by the context discussed in [21], in which the system acts proactively on the basis of context information.

3 Context Awareness in Migration Systems

While conducting usability tests for our previous prototypes of multi-device UI migration [11], it was noticed that the UX was highly related to the selection of the migration target device (*target acquisition*) and the UI migration triggering (*process initiation*). By making the system context-aware, these two aspects can be enhanced by the system offering/choosing relevant choices for target devices and by suggesting or initiating migration automatically, as discussed by *Schilit et al.* [19]. The offering/choosing of relevant target devices and suggesting/initiating migration is enabled by the system being aware of the surrounding devices, environment, users and social relations. Thus, the main benefits of making a migration system context-aware are:

- *Automatic target acquisition and selection* – with the system knowing and using information of the environment, a suggestion or decision can be made of the migration target device;
- *Suggested or automatic migration* – with the system knowing the current situation, it can either suggest the migration to a nearby device (assisted migration) or trigger the migration automatically (automatic migration).

4 System Architecture and Implementation

The context-aware migration system is based on four main components; 1) devices, 2) *Migration Client* running on each device's Web browser, 3) Context Monitor running on the phone and 4) Migration Platform Web server.

The migration is managed by the Migration Platform that utilises a proxy server. The *Migration Proxy* annotates existing Web pages in real-time, when a user is browsing the Web via Migration Client. The browsed pages are injected with additional scripts that enable the migration from one device to another on request, while the original functionalities of the Web application are preserved. Code injection performed by the proxy and strategies to manage security are fully discussed in [11] and [10], respectively. Context Monitor allows triggering the migration based on sensed values of selected context parameters and applied custom rules, providing increased intelligence into the migration. The Context Monitor utilises the added external triggering functionality of the Migration Platform.

4.1 General Architecture Description

To enable context-aware migration, the originating (source) device and the destination (target) device must run the Migration Client in their Web browsers. The Migration Client is a simple Web application for logging the environment and for browsing Web pages via the Migration Platform proxy server.

Typically, the source device is already running the Migration Client, since it is used to navigate Web pages via the Migration Proxy. However, the user must ensure that the target device also has the Migration Client running in order to manage the incoming migration request when the context-based migration is triggered. It is a reasonable constraint in our system that the Migration Client needs to be up and running in the devices' Web browsers.

An overview of the system architecture is shown in Fig. 1, where the environment and main communications are depicted. In Fig. 1 (left), Devices A (phone) and B (desktop computer) have the Migration Client running and the phone has Context Monitor running as well. When the user opens a Web page on Device A, the request goes through (1) the proxy server, which relays it (2) to the actual application server. The application server then responds back (3) with the resource to the proxy server. The Migration Proxy annotates the received resource (e.g. a *HyperText Markup Language* (HTML) page) by adding additional *JavaScript* code and sends the annotated resource as response to Device A (4). The injected code enables the page to be subsequently migrated, and does not affect its original layout or content.

Depending on the selected context parameters and its configuration, the Context Monitor is aware of the Device A situation. For example, it knows the movement and position of the Device A, nearby devices and the preferences of the user. When the system senses an opportune moment for migration (e.g. Device A is in the close proximity of the Device B and placed display-up on a flat surface, such as a table), the Context Monitor sends an external migration request (5) to the Migration Platform, in which the source and target device IDs are specified. A more complete architecture description of the basic Migration Platform, i.e. without context awareness, can be found in [11].

The Migration Platform relays this request to the Migration Client of the specified source device (6), which then invokes the previously injected JavaScript code for the browsed page. The process serialises the *document object model* (DOM) of the browsed page by including the state of the interaction (e.g. content of page elements) and sends this serialised data back to the Migration Platform (7), which saves it locally. The Migration Client running on the target device is then notified (8), with the address to access the Web page to be migrated and automatically opens the page on the Web browser of Device B.

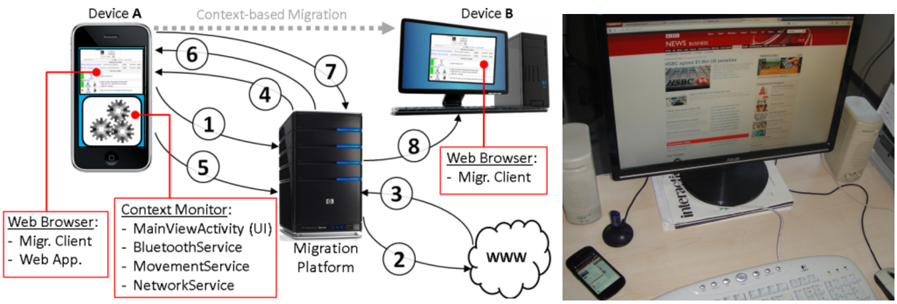


Fig. 1. Overview of the system architecture (left) and typical scenario (right)

It is also to be noted that the Migration Client continuously communicates with the Migration Platform server and provides a unique ID of the devices and user credentials. Therefore, since each instance of the Migration Client is uniquely identified, more than just one potential target device can be running the Migration Client (i.e. be active) in the same environment at the same time.

4.2 Context Monitor Overview

The Context Monitor is a mobile application targeted for *Android* smartphones, which runs on the background and observes the selected context parameters. The context parameters can be e.g. the proximity of other devices (obtained by using the Bluetooth *received signal strength indicator* (RSSI)), the current physical activity of the user (fall detection, still, walking and running activities obtained by using the device accelerometers), the current virtual activity of the user (the currently running applications and currently used applications on the device, call and *short message service*

(SMS) activities etc.), location (*global positioning system* (GPS) coordinates, *wireless local area networks* (WLAN), *base station cell ID* (CID)) and other, such as audio and light etc.

The context parameters can be chosen and configured, so that the user can set when the Web application should be migrated from the smartphone (referred in the following as *phone*) to another device and vice versa. Thus, for example, the user can set the target devices for the migration and also the context-dependent rules, such as *"perform migration from the phone to the office computer when the phone is still after being placed on a table, facing upwards and in the close proximity of the office computer"* and *"perform migration from the office computer to the phone when the phone has been lifted from the table, is moving, and the office computer is not in close proximity"*.

It needs to be noted though, that the Context Monitor is a very limited mobile application developed for the purposes of this study and for the evaluation of the concept of context-aware UI migration. Therefore, the application does not aim to tackle the problems related to the wide field of activity recognition, as those questions are out of the scope of this paper.

5 Evaluation

In this section we present the evaluation of our prototype system to explore how proactive and context-aware UI migration is perceived from user's point-of-view and which technical requirements are needed for. The factors that we were interested in were related to the existence of relevant preferences for a specific migration mode (manual, assisted and automatic) and of possible relationships between individual user factors, interaction behaviour and declared ratings for the tested system.

We recruited a total of 24 persons and as heterogeneous trial group as possible, consisting of both computer science professionals and of people who only use computers and smartphones for basic office worker's purposes. We recruited the trial participants by sending an email within our organisations, asking people matching our recruitment criteria to join study. We personally recruited as many people as possible from within our social working circles matching the recruitment criteria.

We scheduled the user tests for a time span of 1.5 weeks, approximately three to four participants per day, maximum 60 minutes each. Users tested the Context Monitor and the Migration Platform with the simple use case scenarios and answered to the questionnaire formed of the aforementioned interesting factors, such as preferences for specific migration mode and relationships between individual user factors and system behaviour.

The test tasks consisted of Web page migration from a Android phone to a PC and vice-versa using three different interaction modes (manual, assisted and automatic). The diverse available modes to operate an interactive system are discussed in [22], where the authors tackle the prevention of *"mode errors"*, i.e. situations in which users forget the mode in use. The authors discuss how to provide feedback to the users to remind them how the system is operating. It is worth pointing out that in our test setting the system did not provide any explicit mode-dependent feedback, because the users were provided with instructions for each task and informed about the current mode before starting the interaction. In addition to a background questionnaire, we had the participants fill an in-between questionnaire after each interaction mode.

We also recorded the laboratory studies on video to count the focus shifts between the devices and to catch any freeform comments and suggestions during the laboratory tests. By focus shifts, we loosely mean the *macro attention shifts* described by *Holleis et al.* [14].

5.1 Test Scenarios

The current developed versions of the Migration Platform and Context Monitor were used in the laboratory test. The contextual factors we focused on were the phone stability and the Bluetooth presence/proximity information. The two test scenarios are presented in the following.

Scenario 1: The user is approaching the office while browsing some Web page on her Android phone. As soon as s/he gets to her desk and puts the phone on the table (see Fig. 1 (right)), the browsed Web page is migrated, depending of the test mode, either automatically or assisted, from the phone to the computer.

Scenario 2: The user is accessing some Web application on the desktop computer and suddenly needs to leave his office for a meeting. S/he only takes the phone with her and walks away. The Web application is then migrated, either automatically or after user confirmation (depending on whether the selected mode is automatic or assisted), from the computer to the phone. This way, the workflow is not interrupted by the transition from the office to the meeting room.

5.2 Data Analysis

In this section we provide analysis of the data gathered during the user test. The data sources are the questionnaires the participants filled in and the technical data gathered from the Migration Platform during the laboratory tests.

Demography and Background Information

The 24 test participants were in average 32 years old (median 31 years), youngest participant being 26 years old and the oldest participant 44 years old. Only four female were involved in the study. Most of the people were *Finnish* (eleven) and *Italian* (eleven) by nationality, but also one *German* and one *Portuguese* participated. Education level was high: six *Bachelors*, thirteen *Masters* and five *Doctors*. Two participants were partially impaired (one in wheelchair and one without one hand). Approximately half of the participants were "tech-savvy" (developers, early adopters and highly interested in technology) and the rest had "normal technical skills".

System Functionality

We logged all relevant event times at the Migration Platform to count the latencies of the system. Latency directly affects UX because it forces the user to wait for the migration to be carried out. Thus, latency has been considered as indicator of technical quality of the system under investigation and studied in correlation with other variables, as discussed later. We also recorded the user tests on video in order to count the users' focus shifts between the devices during each migration mode and to catch comments. To minimise any learning effects, we shuffled the migration modes for each test participant. By permuting the manual, assisted and automatic modes, we thus obtained six different migration mode orders.

The hypothesis for the focus shifts was that when using the manual mode for migration, there would be more focus shifts between the devices, as the users would check between the devices, and that there would be less focus shifts when moving towards more automatic migration.

In order to get the latencies, we recorded all the relevant events at the Migration Platform server, such as: mobile device placed on the table, migration to PC triggered on the mobile device, Web page opened on PC, mobile device picked at hand, migration to mobile device triggered on the PC and "Web page opened on mobile device.

For the manual mode, we calculated the latencies between the user triggering migration on the mobile device to when the Web page is opened on the computer (A) as well as between the user triggering migration on the computer to when the Web page is opened on the mobile device (B). Focussing on the manual mode, the higher latency of the PC to Mobile migration can be explained by the time taken by the mobile browser to render the migrated page.

For the assisted and automatic modes, we calculated the latencies between the mobile device being placed on the table to when the Web page is opened on the computer (C and E) as well as between the mobile device being picked at hand to when the Web page is opened on the mobile device (D and F). These latencies are depicted respectively in Fig. 2 (left). As it can be seen, in the assisted and automatic modes there was significantly more latency. This is due to the context detection algorithms that take a few seconds to detect the events of *"arrived to the office and placed on the table"* / *"lifted from the table and left the office"*. Indeed, the algorithm continuously monitors acceleration/tilting and needs some time in order to avoid false positives (unwanted migrations). Therefore, the latencies were also higher for the PC to mobile cases, because the context detection algorithms take more time to notice the event of *"leaving the office"* than to detect the *"arrived to the office"* –event.

In case of manual trigger, instead, the time behaviour is mainly due to: a) latency of the page elaboration (client- and server-side) and b) network delays.

Client-side, the page elaboration consists of the serialisation of the page, i.e. the creation of the *eXtensible Markup Language* (XML) string document from the DOM on the browser. Client/side serialisation is done by a JavaScript procedure. Server-side, the elaboration consists of: parsing the XML string to create a document object, filling the document with additional information (e.g. user interaction state) and serialising it into a text/HTML file that will be accessed by the target device browser.

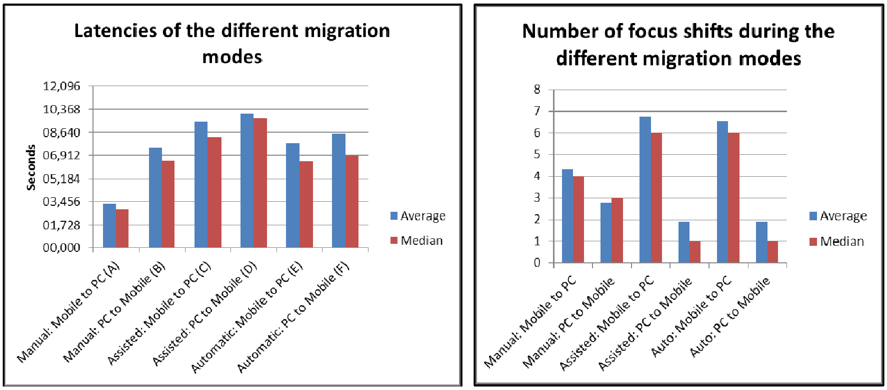


Fig. 2. Latencies (left) and focus shifts (right) of the different migration modes

We argue that, because of the system latencies, assisted and automatic migration led to the highest amount of focus shifts (see Fig. 2). Therefore, we had to claim our initial hypothesis (less focus shifts when moving towards more automatic migration) false for this study. As a future work, the assisted and automatic mode latencies will be reduced as much as possible. The system, as pointed out by many of the test participants, should also give some sort of hint of the migration being in progress, for example, by using a small progress bar over the screen.

Nielsen [16] discusses the acceptable approximate latencies/response times for any application in regards to usability. If the system response time is more than 1 second, feedback during the delay is important and required or otherwise the users do not know whether the system is actually doing anything. With our current prototype the overall response time of the system is more than 1s, therefore it is imperative to give a progress indicator(s) during the migrations. Furthermore, considering the context-detection algorithms and a) the latency of the page elaboration (client- and server-side) and b) the network delays, it will be very challenging to develop the system to perform in the time range of less than a 1 second. However, the participants commented on the free-form sections of the questionnaires that the system already performed surprisingly well, which give us initial direction that a migration latency of few seconds is acceptable, as long as hint(s) of the system progress is displayed. In general, we also gained good insight into the trade-off between the system latency and the detection quality. The participants commented that a slight latency is tolerable, if the context detection does not produce false negatives/positives and system progress is visualised. Half of the participants faced at least one false positive (i.e. an unwanted migration) or negative (i.e. a migration that did not occur when needed). Among such failures, 33 were false negatives and 5 false positives. Most issues affected the mobile-to-PC migration, and thus the "arriving to the office" event was not detected as well as the "leaving the office" event.

Statistical Analysis

Statistical tests have been applied to the data gathered during the trials. The aim was to discover possible correlations between several aspects (i.e. variables), such as between the number of video-detected focus shifts of the user and her/his rating. We ran *Pearson correlation tests* over the system latency and focus shifts in relation to the average usability, system speed (usability), suitability of the interaction mode and the willingness to use the system (and the selected mode) in the future. The Pearson correlation tests are summarised in Table 1.

Table 1. Summary of the Pearson correlation tests

VARIABLES		MIGRATION TASK					
		Manual		Assisted		Automatic	
		Mobile - PC	PC - Mobile	Mobile-PC	PC-Mobile	Mobile-PC	PC-Mobile
Latency, shifts	Corr.	0,151	-0,042	-0,052	-0,108	0,125	-0,116
	Sig.	0,48	0,845	0,809	0,617	0,561	0,588
Shifts, average usability	Corr.	-0,228	-0,344	-0,066	0,016	0,215	-0,361
	Sig.	0,285	0,1	0,76	0,941	0,313	0,083
Latency, average usability	Corr.	-0,255	0,147	-0,212	-0,013	-0,037	0,019
	Sig.	0,228	0,492	0,32	0,953	0,866	0,931
Shifts, speed usability	Corr.	-0,187	-0,256	-0,054	0,196	-0,041	-0,168
	Sig.	0,383	0,228	0,802	0,359	0,851	0,433
Latency, speed usability	Corr.	-0,351	-0,083	-0,216	0,133	-0,54	0,231
	Sig.	0,092	0,699	0,311	0,535	0,006	0,277
Shifts, suitability	Corr.	0,321	-0,131	0	-0,093	0,107	-0,03
	Sig.	0,126	0,541	1	0,665	0,618	0,891
Latency, suitability	Corr.	-0,473	0,075	0,187	0,079	0,146	-0,201
	Sig.	0,02	0,729	0,381	0,714	0,496	0,347
Shifts, willingness	Corr.	-0,132	-0,391	-0,169	-0,075	-0,114	-0,411
	Sig.	0,539	0,059	0,431	0,726	0,596	0,046
Latency, willingness	Corr.	0,01	0,189	0,056	-0,179	0,081	0,258
	Sig.	0,961	0,376	0,795	0,402	0,707	0,223

The following variables have been considered:

- *Latency:* time needed to perform migration, as previously described in subsection System functionality;
- *Shifts:* number of focus shifts between devices. A focus shift occurs anytime the device the user is gazing changes (i.e. from PC to phone or from phone to PC);
- *Average usability:* mean value among the following usability-related ratings of the user: task easiness, system behaviour understandability, system interaction suitability, speed, reliability, willingness to use the system in the future;
- *Speed usability, Suitability and Willingness:* These refer to system speed, system interaction suitability and willingness to use the system in the future, respectively.

The correlation factor, labelled as *Corr.*, indicates whether the correlation is *moderate* ($0.3 < |\text{Corr.}| \leq 1$), *weak* ($0 < |\text{Corr.}| \leq 0.3$) or *null* ($|\text{Corr.}| = 0$, i.e. variables are independent).

The correlation significance is indicated by *Sig.:* correlation can be *highly significant* ($0 \leq \text{Sig.} < 0.01$) or *statistically significant* ($0.01 \leq \text{Sig.} < 0.05$).

It is worth pointing out that the aim of these tests is mainly exploratory, because they are not actually devoted to accept/reject any predefined hypothesis. Therefore, even if multiple comparison tests involving the same variables were performed, no correction has been applied to the significance intervals.

Such tests show the following statistically interesting aspects:

1. There is highly significant (negative) correlation between latency and system speed usability when using automatic mode from mobile to PC;
2. A statistically significant (negative) correlation exists between latency and suitability when manually migrating from mobile to PC;
3. A statistically significant (negative) correlation for the automatic mode was found between focus shifts and willingness to use the system in the PC to mobile task;
4. A moderate (negative) correlation, although not statistically relevant, was found between latency and speed usability when manually migrating from mobile to PC.

Given these findings, we can argue that they support our notions on how the system was perceived during the tests. Automatic migration took a longer time (due to the context-detection algorithms delay) when migrating from mobile to PC and clearly had the participants question themselves about system usability. Also, when using the manual mode in migrating from mobile to PC, the latency reduced the interaction suitability and had the participants commenting on the manual mode usability.

It is also worthwhile to mention that somewhat surprisingly the results neither show correlation between latency and focus shifts, nor between these two variables and the average usability. This lack of correlation is indicated by the low correlation factor (weak, in most cases) as well as by the significance value, which is much bigger than 0.05. We could argue that the system already performs “good enough” to provide such context-aware migration functionality. As mentioned before, this also shows in the participants’ free-form comments that, overall, they already found the system to be reasonably good.

We also ran statistical difference tests on the ratings (*Likert* scale 1-5) the participants gave during the laboratory tests and the system latencies and found out several significant differences. The null hypothesis on the tests was H0: *“The modes perform similarly”* (i.e. the means of the groups are the same).

The results show that we have to reject our null hypothesis (using the conventional 0.01 and 0.05 alphas) on the following cases of the system functionality assessment (numbers in round brackets indicate the p-value):

a) Task easiness, assisted vs. manual, highly significant (0,0001)
b) Task easiness, auto vs. manual, highly significant (0,0008)
c) Interaction suitability, assisted vs. manual, statistically significant (0,0233)
d) Interaction suitability, auto vs. manual, highly significant (0,0059)
e) Willingness to use in the future, assisted vs. manual, highly significant (0,0093)
f) Willingness to use in the future, auto vs. manual, statistically significant (0,0281)
g) Latency, mobile-to-PC, assisted vs. manual, highly significant (< 0,0001)
h) Latency, PC-to-mobile, assisted vs. manual, statistically significant (0,0284)
i) Latency, mobile-to-PC, auto vs. manual, highly significant (< 0,0001)

Such statistics confirm a relevant preference towards automatic and assisted modes (with respect to the manual mode).

These statistics are related to confirmatory data analysis, as they are devoted to reject the null hypothesis. *Bonferroni Correction* has thus been applied with significance level ($\alpha/2$), aiming to reduce possible family-wise errors. Some variables (i.e. latency in mobile-to-PC manual mode and task easiness manual mode) were indeed involved in two comparisons. The correction lowers the alphas from 0.01 and 0.05 to 0.005 and 0.025, respectively. If such corrected alphas are considered, then some downgrades in the statistical significance occur: d) and e) are re-classified from highly significant to statistically significant, while f) and h) change from statistically significant to insignificant. Nevertheless, even after applying Bonferroni correction, most comparisons are still statistically meaningful. For instance, the e) corrected comparison between willingness to use assisted and manual modes, is also significant, which is interesting as it suggests a preference of the users towards the assisted mode.

Based on these results, we can argue that the assisted and automatic modes were easier to use in the test and suited the given interaction situation better. The participants were also keener to use to the assisted and automatic modes in the future vs. using the manual mode. However, it is worth mentioning that, given further development to system by making the manual mode more easier, the users commented that the manual mode could be useful in some cases as well (e.g. in public spaces when definite explicit control is more desirable).

6 Conclusion and Future Development

We have presented and studied an integrated solution for automatic (implicit) and assisted (suggested) context-aware migration of Web applications, by comparing it with a manual (explicit) one.

The focus has been put on relationships between technically gathered data and users' feedback. We discovered interesting correlations between, for instance, system response time and subjective aspects such as suitability and perceived reactivity of the system. Statistically relevant preferences for automatic and assisted migration modalities with respect to the conventional manual triggering have also been highlighted. Based on the Pearson correlation tests, we claim that 1) even though there was latency in the assisted and automatic modes, the average usability of context-aware migration was still perceived as quite good by the participants and 2) for further development more visual cues of the system progress need to be given. Also, based on the statistical difference tests on the users' ratings, we can argue that the assisted and automatic modes performed better and that the users would prefer those over explicit migration.

According to the study results, we can argue that migratory functionalities for UI and data across different kinds of devices are judged as promising by the users. The migration mode (manual, assisted or automatic) is highly dependent on the task at hand and also impacts on how users might perceive technical parameters such as system latency. We may use an incremental strategy to reduce latency: rather than serializing the full document on the source device and moving it to the target, a document copy could be kept state-persistent in the Migration Platform by means of updates sent from the source. Technical aspects still deserve more investigation. However, if context-based migratory functionalities were available and properly tuned, they would therefore likely to be used by the great public.

Acknowledgements. We gratefully acknowledge support from the EU ARTEMIS SMARCOS Project[1].

References

1. Ballendat, T., Marquardt, N., Greenberg, S.: Proxemic Interaction: Designing for a Proximity and Orientation-aware Environment. In: Proc. ITS 2010, pp. 121–130. ACM Press (2010)
2. Berry, L., Bartram, L., Booth, K.S.: Role-based Control of Shared Application Views. In: Proc. UIST 2005, pp. 23–32. ACM Press (2005)
3. Bourges-Waldegg, D., Duponchel, Y., Graf, M., Moser, M.: The Fluid Computing Middleware: Bringing Application Fluidity to the Mobile Internet. In: Proc. SAINT 2005, pp. 54–63. IEEE (2005)
4. Brdiczka, O.: Integral Framework for Acquiring and Evolving Situations in Smart Environments. Journal of Ambient Intelligence and Smart Environments 2, 91–108 (2010)
5. Chang, T.H., Li, Y.: Deep Shot: A Framework for Migrating Tasks Across Devices Using Mobile Phone Cameras. In: Proc. CHI 2011, pp. 2163–2172. ACM Press (2011)
6. Cheverst, K., Dix, A., Fitton, D., Kray, C., Rouncefield, M., Sas, C., Saslis-Lagoudakis, G., Sheridan, J.G.: Exploring Bluetooth-based Mobile Phone Interaction with the Hermes Photo Display. In: Proc. MobileHCI 2005, pp. 47–54. ACM Press (2005)
7. Dearman, D., Pierce, J.S.: It's on My Other Computer! Computing with Multiple Devices. In: Proc. CHI 2008, pp. 1144–1153. ACM Press (2008)
8. Dees, W.: Usability of Nomadic User Interfaces. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part III, HCII 2011. LNCS, vol. 6763, pp. 195–204. Springer, Heidelberg (2011)
9. Fischer, C., Gellersen, H., Gostner, R., Guinard, D., Kortuem, G., Kray, C., Rukzio, E., Strengin, S.: Supporting Device Discovery and Spontaneous Interaction with Spatial References. Personal and Ubiquitous Computing 13(4) (2009)
10. Ghiani, G., Isoni, L., Paternò, F.: Security in Migratory Interactive Web Applications. In: Proc. MUM 2012. ACM Press (2012)
11. Ghiani, G., Paternò, F., Santoro, C.: Push and Pull of Web User Interfaces in Multi-Device Environments. In: Proc. AVI 2012, pp. 10–17. ACM Press (2012)
12. Google. The New Multi-screen World: Understanding Cross-platform Consumer Behaviour. Research Report (August 2012), `http://services.google.com/fh/files/misc/multiscreenworld_final.pdf`
13. Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R., Wang, M.: Proxemic Interactions: The New Ubicomp? Interactions 18, 42–50 (2011)
14. Holleis, P., Otto, F., Hußmann, H., Schmidt, A.: Keystroke-Level Model for Advanced Mobile Phone Interaction. In: Proc. CHI 2007, pp. 1505–1514. ACM Press (2007)
15. Marquardt, N., Diaz-Marino, R., Boring, S., Greenberg, S.: The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In: Proc. UIST 2011, pp. 315–326. ACM Press (2011)
16. Nielsen, J.: Usability Engineering. Morgan Kaufman Publishers Inc., San Francisco (1993)
17. Rekimoto, J.: Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In: Proc. UIST 1997, pp. 31–39. ACM Press (1997)
18. Rekimoto, J., Ayatsuka, Y., Kohno, M., Oba, H.: Proximal Interactions: A Direct Manipulation Technique for Wireless Networking. In: Proc. Interact 2003, pp. 511–518. IOS Press (2003)

---

[1] `http://www.smarcos-project.eu`

19. Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. In: Proc. WMCSA 1994, pp. 85–90. IEEE Computer Society Press (1994)
20. Schilit, B.N., Sengupta, U.: Device Ensembles. IEEE Computer 37, 56–64 (2004)
21. Schmidt, A.: Implicit Human Computer Interaction through Context. Personal and Ubiquitous Computing 4(2/3), 191–199 (2000)
22. Sellen, A., Kurtenbach, G., Buxton, W.: The Prevention of Mode Errors through Sensory Feedback. Human Computer Interaction 7(2), 141–164 (1992)
23. Tabard, A., Hincapié-Ramos, J.D., Bardram, J.E.: The eLabBench in the Wild – Supporting Exploration in a Molecular Biology Lab. In: Proc. CHI 2012, pp. 3051–3060. ACM, New York (2012)
24. Wäljas, M., Segerståhl, K., Väänänen-Vainio-Mattila, K., Oinas-Kukkonen, H.: Cross-Platform Service User Experience: A Field Study and an Initial Framework. In: Proc. MobileHCI 2010, pp. 219–228. ACM Press (2010)

Generation of Multi-Device Adaptive MultiModal Web Applications

Marco Manca, Fabio Paternò, Carmen Santoro, and Lucio Davide Spano

CNR-ISTI, HIIS Laboratory, Via G. Moruzzi 1,
56124 Pisa, Italy
{marco.manca,fabio.paterno,carmen.santoro,
lucio.davide.spano}@isti.cnr.it

Abstract. This paper presents a set of tools to support multimodal adaptive Web applications. The contributions include a novel solution for generating multimodal interactive applications, which can be executed in any browser-enabled device; and run-time support for obtaining multimodal adaptations at various granularity levels, which can be specified through a language for adaptation rules. The architecture is able to exploit model-based user interface descriptions and adaptation rules in order to achieve adaptive behaviour that can be triggered by dynamic changes in the context of use. We also report on an example application and a user test concerning adaptation rules changing dynamically its multimodality.

Keywords: Model-based design, Multimodal User Interfaces, Adaptation, Adaptation rules.

1 Introduction

Recent years have been characterized by the increasing availability of interaction platforms, differing also in modalities. One example is the vocal modality, which nowadays enjoys greater support and is used in various applications in the mass market (e.g. Google Voice, iOS SIRI). Unfortunately, although Web applications are widely used for many purposes, this richness in interaction modalities has penetrated the Web only in a limited manner. Indeed, besides the issues of addressing the heterogeneous set of available devices, Web applications are affected by the problems determined by traditional input/output interactive mechanisms that can result in rigid and unnatural interactions, which are often inconvenient especially for mobile Web users. Exploiting multimodality in Web applications is an important opportunity, since multimodal interaction can significantly increase the communication bandwidth between the user and the devices by offering flexibility and freedom. However, to date, there is a lack of approaches able to flexibly address the issues of developing multimodal Web applications able to adapt to the different contexts of use to better support mobile users. Some reasons for this can be found in the inherent complexity of developing adaptive multimodal interactive systems.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 218–232, 2013.

To this regard, the exploitation of model-based descriptions can be a promising solution for addressing the increasing complexity of current interactive applications. Furthermore, the potential of model-based approaches can be better exploited when addressing user interfaces using different modalities, but unfortunately their application has mainly focused on mono-modal interfaces or desktop-to-mobile adaptation, basically involving just the visual channel. In addition, most model-based approaches have mainly considered adaptation at design time (e.g. [7, 13]). One major challenge is thus to be able to handle effectively and dynamically adaptation of user interfaces at runtime. This seems a viable solution for providing users with the most effective user interfaces in various contexts, as it is able to address unforeseen (at design time) situations and also configure solutions for future needs.

We present a novel solution that supports multimodal (combining graphical and vocal modalities) adaptive user interfaces by exploiting a model-based language. In particular, the main contributions are: a novel generator of multimodal interactive applications that are implemented in HTML annotated with particular CSS classes, which can be easily interpreted by different clients in different devices; and an adaptation support able to change the multimodality at various granularity levels according to dynamic contextual events.

In the paper, after discussing related work, we provide some useful background information to introduce some concepts that have been further refined and exploited in this work. Next, we describe the overall architecture supporting multimodal adaptive Web interfaces, including the descriptions of some elements, such as the language for specifying adaptation rules. We provide details on the innovative solution proposed in this work that is able to generate multimodal Web pages. Then, we introduce an example application and report on an evaluation in which various multimodal adaptations have been considered. Lastly, we draw some conclusions and provide indications for future work.

2 Related Work

A model-based Framework for Adaptive Multimodal Environments (FAME) has been proposed in [5]: it includes an architecture for adaptive multimodal applications, a new manner to represent adaptation rules and a set of guidelines to assist the design process of adaptive multimodal applications. However, FAME's objective is just to guide the development of adaptive multimodal applications, but it does not provide support for automatic application development, as in our case.

Octavia et al. [9] describe an approach to design context-aware application using a model-based design process, but they do not address multimodal adaptation. The model-based approach is considered also in [12] for its flexibility. The context of use is a structured information space whose goal is to inform the adaptation process. When some changes in the context occur, a module identifies the appropriate transformation to adapt the UI to the current change. However, authors of [12] do not consider run time adaptation without regenerating the whole interface and they do not consider multimodal adaptation.

A model and an adaptation architecture for context-aware multimodal documents has been put forward [3]. However, the adaptation model described in that paper only considers static adaptation phase (at design time), and there is no runtime adaptation.

MyUI [11] provides a framework for run-time adaptations while supporting accessibility. To cover the heterogeneity of users, environments and devices, MyUI relies on an extensible repository of multimodal UI design patterns expected to contain the knowledge needed to perform the described three-stage process of UI generation and adaptation. MyUI's approach, although focused on accessibility issues, shares with our work some concepts. However, its design-patterns approach can quickly become cumbersome, e.g. in the need of consistently specifying/maintaining the relations existing between the design patterns specified within the repository.

Another approach [6] proposes and investigates a semi-automatic rule-based generation of multimodal user interfaces where a discourse model (representing modality-independent interaction) is transformed to different, automatically rendered modalities. Although the model transformation process exploited in that work can be compared to the one used in our approach, tool support has not been deeply investigated in that approach.

DynaMo (Dynamic multiModality) [1] proposes an approach aimed at handling dynamic environments where devices and applications are rapidly evolving. It is based on partial interaction models, thus it combines and completes them at runtime. Differently from this work, we use UI models which are complete, and are able to cope with the evolving context by dynamically adapting the UIs at runtime according to current conditions of users, devices, and surrounding environment.

W3Touch is a tool that aims to collect user performance data in order to help identify potential design problems for touch interaction. In mobile devices it has been exploited [8] to support adaptation based on two metrics related to missed links in touch interaction and zooming issues to access Web content. However, that work has not considered multimodal UIs. While [2] introduces some initial ideas about adaptive user interfaces through the use of model-based approaches, in this paper we present a novel solution for this purpose able to build multimodal Web interactive applications, which can be executed in any browser-enabled device, without requiring developers to use any particular API in addition to standard HTML, CSS, and JavaScript.

3 Background

We considered the MARIA language [10] for describing interactive applications as a starting point of our work because it is one example of engineered model-based language, with already some tool support. It provides one language for the Abstract User Interface level (AUI), and multiple languages for platform-dependent concrete UI (CUI) descriptions. The associated tool (MARIAE) supports multiple transformations between levels, and generators for various platforms (desktop, vocal, multimodal). However, the multimodal generator provided [7] generates X+V implementations, which was the implementation language considered also in other model-based generators [13]. X+V was an attempt to combine VoiceXML and XHTML, but it is no longer supported by current browsers. Thus, we decided to investigate novel solutions able to generate multimodal user interfaces that can be exploited with current browsers.

The MARIA concrete multimodal language uses the CARE properties to specify how the various modalities are allocated. Such properties were originally defined in [4]. We interpret them in the following way: *Complementarity*: when the considered part of the UI is partly supported by one modality and partly by another one; *Assignment*: the considered part of the UI is supported by one assigned modality; *Redundancy*: the considered part of the UI is supported by both modalities; *Equivalence*: the considered UI part is supported by either one modality or another.

To provide a flexible definition of multimodal interfaces it is possible to apply such properties to various UI concepts of the MARIA language: composition operators (groupings, repeaters, …), interaction and only-output elements. In order to have different modalities allocated even at a finer grain, interaction elements themselves can be further structured into prompt, input, and feedback with this meaning:

- *Prompt*: it represents the UI output indicating that it is ready to receive an input;
- *Input*: it represents how the user can provide the input;
- *Feedback*: it represents the response of the system after the user input.

We interpret the CARE properties from the user viewpoint. Thus, the *equivalence* property is applied only to the input part of an interaction element when multiple modalities are available, in order to indicate the possibility to insert the input indifferently through one of these modalities. We do not apply it to output elements because even if the system can choose between two different modalities to present output, in the end only one of them will be provided and perceived by the end user, and so the user will perceive a modality assignment to the output element. *Redundancy* is applied to output elements and to prompt and feedback but not to input for interaction elements since once the input is entered, it does not make sense to enter it again through another modality.

We have also developed a high-level XML-based description language for defining the transformations affecting the interactive application when something occurs at the context level or in the interactive application. This language enables the specification of adaptation in terms of rules expressed as Event, Condition, and Action(s):

- *Event*: It can be an elementary event or even a composition of events occurring in the interactive application or in the context of use. In any case, its occurrence triggers the application of the rule.
- *Condition* (optional): a Boolean condition to be satisfied to execute the associated action(s). It can be related to something happened before, or some state condition.
- *Action*: How the abstract/concrete/implementation description of the interactive application should change in order to perform the requested adaptation.

The impact of adaptation can be at different granularities: complete change of UI; change of some UI parts; change of attributes of specific UI elements.

Figure 1 shows an example rule related to environment noise. If the noise level is greater than 50 (decibels) as specified in the condition part (rows 6-10), then this rule is triggered and the presentation is adapted in a way that only graphical interaction is supported: indeed the corresponding action updates the presentation CARE value by setting the graphical assignment value to the output attribute (row 13-16).

```
1  <rule priority="0" name="onlyGUI" id="rule4">
2      <event>
3          <simple_event event_name="onEnvironmentNoiseLevelIncreased"
4                        xPath="/user/environment/@noise_level" externalModelId="ctx"/>
5      </event>
6      <condition operator="gt">
7          <entityReference xPath="/user/environment/@noise_level"
8                           externalModelId="ctx"/>
9          <constant value="50" type="int"/>
10     </condition>
11     <action>
12         <update>
13             <entityReference xPath="/descendant::presentation[@id='agencyList_iu']/@output"
14                              externalModelId="cui"/>
15             <value>
16                 <constant value="graphical_assignment" type="string"/>
17             </value>
18         </update>
19     </action>
20 </rule>
```

Fig. 1. Rule example

4 Approach

Figure 2 shows the overall architecture of the adaptation environment that can perform different kinds of adaptations, not only the multimodal one. By using an authoring environment at design time, it is possible to obtain both the logical specification of the interactive application ("Logical UIs" in Figure 2) and the specification of the rules to be used for adaptation ("Adaptation Rules") which are both passed as input to the Adaptation Engine. The latter determines the optimal adaptation for the interactive system at hand in the current context of use, and based on the specified adaptation rules. To achieve this goal, the Adaptation Engine receives from the Context Manager information about the current context (e.g. events detected), and checks whether some adaptation rules could be triggered, by analyzing if the event/condition parts of some rules are satisfied. In the positive case, the corresponding action part of the selected rule is sent to the module in charge to perform the associated modifications. Such module can change depending on the type of adaptation defined in the triggered rule. As Figure 2 shows, there are three possibilities:

- if there is a complete change of modality (e.g. from a graphical interface to a vocal UI) the CUI of the current UI is sent to the adapter for the target modality, which has to change the structure of the application in order to make it suitable for the new modality, and then provide the new specification to the relevant generator;

- if there is a change in the structure of the current user interface (e,g. a change in the user interface layout structure), then the adaptation engine modifies the current CUI and sends the result to the generator to update the UI;
- if there is a small change in terms of UI attributes (e.g. the font sizes are increased) then for sake of efficiency the adaptation is performed directly on the current UI implementation through some scripts.

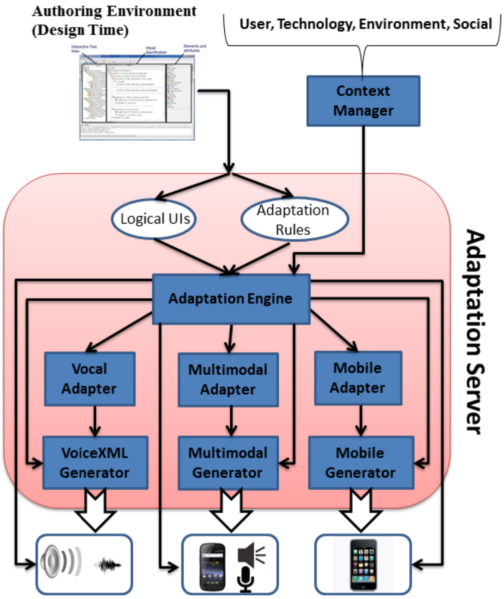


Fig. 2. Overall architecture

The adaptation engine receives events from the Context Manager, which handles relevant information regarding context (which is defined in terms of four key dimensions: user, technology, environment, social aspects). The Context Manager has a client/server architecture (its description is out of the scope of this paper for lack of space). The context events are detected by Context Delegates (small applications installed on the devices), which monitor environment parameters and send updates to the Context Manager.

5 Multimodal Generator

The goal of the new multimodal UI generator is to produce implementations in standard HTML+CSS that can be executed by any browser-enabled device. Such implementations are structured in two parts: one for the graphical UI and one for the vocal one. The vocal part is supported by JavaScript code accessing the Google APIs for vocal interaction.

In order to understand when to activate the JavaScript functions that access the vocal libraries at run-time (see Figure 3), each user interface element is annotated with a specific CSS class in the implementation generated at design time, according to the indications of the CARE properties. If it contains a vocal part, the class *tts* for the output elements and prompt part of interaction element is added, while the class *asr* is added for the input parts of interaction elements. The generated elements are marked with these classes because the included scripts use them to identify all the graphical elements having an associated vocal part.

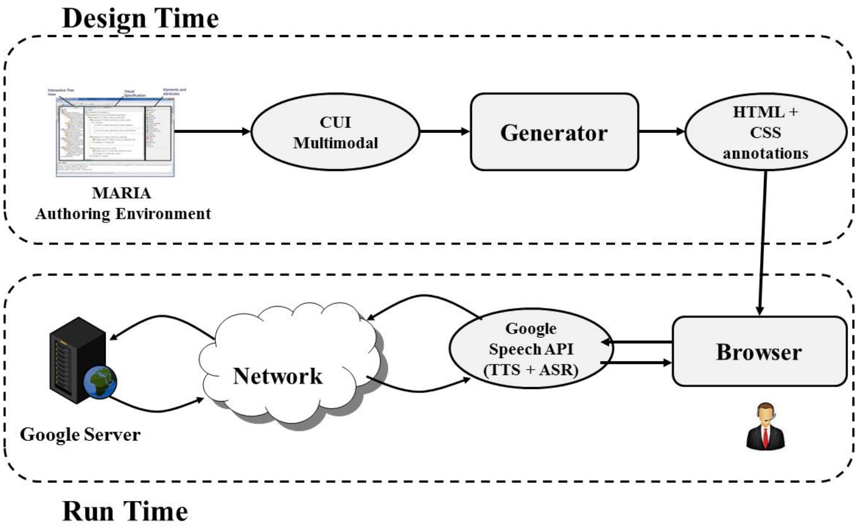


Fig. 3. The generation process and the Multimodal support

It is worth pointing out that the way in which JavaScript code accesses the vocal libraries depends on the platform type (desktop or mobile). More specifically, for desktop applications we exploit the Chrome extension mechanism. Chrome extensions are small software programs that can modify and enhance the functionality of the Chrome browser. They can be written using Web technologies such as HTML, JavaScript, and CSS. We have implemented an extension that is able to exploit the Chrome APIs for automatic speech recognition (ASR) and text-to-speech synthesis (TTS). The basic idea is that such extension, by analysing the DOM of the multimodal web page can identify the elements that need vocal support and call the corresponding vocal Google libraries (depending on the CSS class).

Since the mobile version of Chrome is currently not able to support extensions, for mobile devices we cannot access the Google APIs for ASR and TTS through JavaScript. Thus, for the mobile Android platform we developed a Multimodal Browser Application that exploit a Java component called WebView, which is able to show Web pages using the WebKit rendering engine. In this case the DOM of the Web pages is accessed through a JavaScript that is inserted in the page by the WebView component created for this purpose. The difference with respect to the desktop solution is that the calls to the vocal synthesizer and recognizer are not carried out through JavaScript but using Java functions. Thus, the mobile version still uses JavaScript to access the DOM and find the vocal elements but then call Java functionalities for implementing the vocal part. This is obtained thanks to the fact that Android supports the possibility to invoke methods contained into a Java class through a JavaScript interface in a WebView component. In the Java class we can define the methods we want to expose to JavaScript, such as the methods to synthetize a text or to recognize a vocal input. In this way it is possible to call from the JavaScript executed in the WebView the Java methods defined in the JavaScript interface. Besides these technical differences in the implementation, both solutions can interpret the same HTML descriptions of the multimodal UIs, which are structured in the same way in both mobile and desktop versions.

Figure 4 shows a code example for multimodal interaction generated from a concrete user interface (CUI) and concerning the definition of a *select* element with both a graphical and vocal part.

As you can see, even the vocal part (which starts from row 11) is obtained through HTML (through *span* and *p* tags), and it is hidden (see row 11: the *style* attribute value is "display:none") but it is always accessible from the DOM.

The label element (row 2) is associated with the *tts* class because the prompt part is associated with the redundancy property. The corresponding vocal prompt has the same id plus the suffix '_tts' (row 11). It contains the specification of two prompts (by using the p element annotated with tts_speech class, each prompt has different and sequential count attribute, which is used to indicate in which order the prompt will be synthesized) and a value for the break length (p element annotated with tts_break class) that represents a pause after synthesizing the prompt and before recording the vocal input. For each output vocal element it is possible to define a number of properties (e.g. row 12) such as pitch, rate, gender, volume that permit to control the Text-to-Speech Engine.

The *select* element (and obviously all the input elements) is annotated with ASR class (row 5) because its input part value is "Equivalence", the corresponding vocal element has the same id plus a suffix that depends on the element type (row 22: suffix '_select'). This element contains some parameters: listValues (row 23): when true after the prompt all the possible input values will be synthesized; askConfirmation (row 24): when true the multimodal support asks users to confirm the vocal input recorded; vocal Feedback (row 25): it indicates that the multimodal support will synthesize the text indicated in the element and the input recorded; vocal Event (row 29) such as 'no input' and 'no match': this element indicates the behavior of the multimodal support when a user does not enter an input or enter an unrecognizable input, it is possible to synthesize a message or synthesize the next prompt (the prompt identified by the next value in the count attribute).

We introduce this type of annotations because the multimodality support automatically attaches an event handler triggered by the *onload* event that identifies all the elements associated with the *tts* and *asr* classes, and sets a handler related to the focus event for each of them. This additional event handler identifies the corresponding element in the vocal part (same id plus a suffix, see an example in Figure 4 where in row 5 there is the graphical element and in row 22 the corresponding vocal element) and activates it in order to obtain multimodality when the corresponding graphical element receives the focus. In case of equivalent input when the UI element receives the focus, the vocal prompt is rendered (if the prompt is associated with vocal modality as well); in any case the vocal input recording is then started.

If the input is entered graphically, the application generates an *onchange* event (all the input elements have a handler for this), then the vocal input recording is stopped; if the feedback is associated with the vocal modality, it is rendered accordingly. In addition, all the graphical elements with an associated vocal part have a *tabIndex* attribute with consecutive values: if the application parameter named “continuous reading” is set to true, then after having synthesized a text or recorded an input, the focus is automatically moved to the element with the next *tabIndex* value; otherwise the application waits for the user to set the focus manually.

```
1  <!-- GRAPHICAL PART -->
2  <label for="departure_city" id="departure_city_label" class="tts" tabIndex="46">
3      Choose the departure city
4  </label>
5  <select id="departure_city" name="departure_city" class="asr" tabIndex="47">
6      <option id="departure_city_1" value="milan">milan</option>
7      <option id="departure_city_2" value="rome">rome</option>
8      <option id="departure_city_3" value="florence">florence</option>
9  </select>
10 <!--VOCAL PART -->
11 <span style="display:none" id="departure_city_label_tts">
12     <p class="tts_speech" title="{'count' : '1', 'timeout' : '10s', 'pitch' : 'medium',
13                                  'rate' : 'medium', 'gender' : 'female'}">
14         Choose your departure city
15     </p>
16     <p class="tts_break">1s</p>
17     <p class="tts_speech" title="{'count' : '2', 'timeout' : '10s', 'pitch' : 'medium',
18                                  'rate' : 'medium', 'gender' : 'female'}">
19         Choose the departure city, try to say Rome
20     </p>
21 </span>
22 <span style="display:none" id="departure_city_select">
23     <p class="listValues">true</p>
24     <p class="askConfirmation">true</p>
25     <span class="vocalFeedback" title="{'count' : '1', 'timeout' : '3s'}">
26         The departure city is
27     </span>
28     <p class="tts_break">1s</p>
29     <span class="vocalEvent">
30         <p class="noInput" title="{'reprompt' : 'true'}">No input</p>
31         <p class="noMatch" title="{'reprompt' : 'true'}">
32             You choose a city not available, sorry
33         </p>
34     </span>
35 </span>
```

Fig. 4. Multimodal generated code

The management of the two modalities through the JavaScript event handlers is completely automated by the multimodal support. In the Web application markup code there is only the need to indicate the graphical and the vocal parts through annotation of the CSS classes. This is performed automatically by the multimodal generator that builds final UIs from MARIA specifications. In addition, the differences in the implementation of the desktop and mobile multimodal support do not have impact on the HTML + CSS syntax used for specifying the UI, which is valid for both platforms.

6 Adaptation

Figure 2 shows the architecture of the environment, in order to receive the activated adaptation rules, the Adaptive Multimodal Application has to subscribe to the Adaptation Engine sending the application name and optionally the name of the current user (if logged-in). Consequently, when the Adaptation Engine module receives the application subscription it has to subscribe itself to the Context Manager in order to be notified of the occurrence of the contextual events expressed in the adaptation rules associated to the subscribed application.

As previously mentioned, the adaptation rules are expressed as Event, Condition and Action. The action part can be expressed by the following elements: create, read, update, delete and invoke functions. However, for supporting multimodal adaptation the update element, which changes the CARE values to modify graphical/vocal interaction, is sufficient. This change can be sent either to the multimodal generator (if it is defined in terms of the concrete user interface language) or directly to the implementation. In the former case, first the CUI is updated and then a new UI is generated accordingly.

If the rule actions are sent by the adaptation engine directly to the multimodal support of the application implementation, then it is interpreted in the following way:

- *Prompt* and *Output* part: if the new CARE value is “graphical assignment”, the adaptation script (in the multimodal support) removes the TTS class so that the application does not activate the vocal part; otherwise, the script adds the TTS class. If the new CARE value is “vocal assignment” then the graphical part is hidden;
- *Input* part: if the new CARE value is “graphical assignment” then the adaptation script removes the ASR class, otherwise the script adds it. If the new CARE value is “vocal assignment” then the graphical part is visible but the input can be entered only vocally, thus the input element is read only;
- *Feedback* part: it is implemented with a *span* with *vocalFeedback* class inside the vocal element (see Figure 4 row 27); if the CARE value is “graphical assignment” the adaptation script changes the value of the *span* class from *vocalFeedback* to *noVocalFeedback* so that the application does not synthesize the feedback.

Thus, changing the CSS class of the elements according to the CARE properties is sufficient to obtain the multimodal adaptation because the support will activate or not the vocal part accordingly.

7 Example Application

An adaptive multimodal application supporting interaction with a Car Rental service was obtained from a model-based specification using the MARIA language, through automatic transformations. Through this application users can search for cars by specifying some parameters such as location and maximum distance allowed from it.

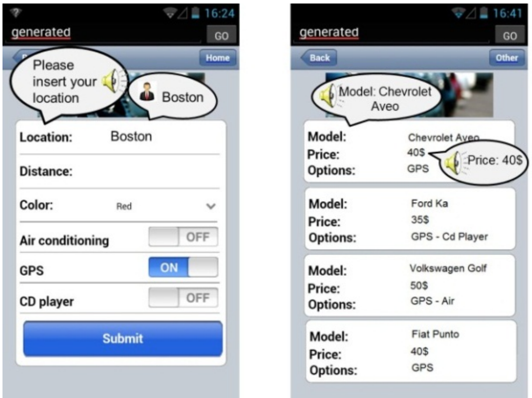


Fig. 5. The UI of the Car Rental application

Furthermore, the user can specify additional options regarding the car, i.e., the on-board availability of air conditioning, CD player and GPS. The implemented application exploits existing REST Web services described through the WADL language to support the expected functionalities. The application generated contains forms, navigation items, access to remote Web services, pages with dynamic content. Thus, it has all the main features that characterize Web applications. Through this Web service it is possible to obtain some information listing e.g. the agencies and their related information, cars available in an agency, etc. For instance, after the user specifies the search parameters through the application, a list of agencies that fulfill the search criteria is provided: for each of them the available cars along with their description, price and extra features are listed. Figure 5 shows example user interfaces of the Car Rental application obtained. The first one is the multimodal search page through which users can provide the search parameters (also using the vocal modality): for all interaction elements the prompt and the feedback part are redundant, while the input part is equivalent. The second page is the search result, rendered in a redundant way: all the output elements are rendered both graphically and vocally.

8 Evaluation

We carried out a user test on the multimodal car rental application. It was useful to gather some preliminary empirical feedback on various types of adaptations affecting a multimodal UI, and how they were perceived by users, in terms of their appropriateness and impact on user experience.

We identified five types of multimodal adaptations, described below. The rationale followed for selecting them was driven by the willingness of covering various types of adaptation of the multimodal UI considered, by varying the way in which the two modalities (graphical and vocal) were allocated. This led to selecting three 'extreme' cases (namely: *Adaptation 1, 2 and 3*) where the adaptations changed the number of modalities involved. For instance in Adaptation1, the UI moved from a multimodal (graphical + vocal) UI, to a mono-modal (only graphical UI). In addition, two cases (*Adaptation 4 and 5)* covered "intermediate" situations (i.e. the UI is multimodal before and after adaptation, but the allocation of graphical/vocal modalities to the various UI elements changes). We considered the following five adaptation scenarios.

- *Adaptation 1*: the user started with a multimodal UI where system output was redundantly provided and user's input was provided only graphically. A context change was triggered (simulating a noisy environment) and the UI became a (mono-modal) graphical UI. This is because when user enters a noisy environment, audio is less effective, while graphical modality is more appropriate.
- *Adaptation 2*: the initial UI was the same as in 1st scenario. Then, the user started walking quite fast, and the UI became a (mono-modal) vocal UI. This is because when the user walks fast, s/he cannot look often at the device display, so the audio channel is more appropriate.
- *Adaptation 3*: the initial UI was the same as in 1st scenario. The user started walking slowly, and the Adapted UI was a multimodal UI in which: i)UI output elements were rendered redundantly; ii) Selection/edit elements equivalently supported user input (either vocally or graphically), and redundantly rendered the prompt and the associated feedback; iii)for activation/navigation elements: user could graphically provide input; prompt was graphically provided, the feedback was redundantly given. This was done because when user walks slowly, there are some opportunities (more than the previous case) for the user to look at the screen. So, in the adapted UI, more flexibility is allowed (e.g. by setting Equivalence to the input of selection/edit elements instead of Graphical Assignment).
- *Adaptation 4*: at first the UI was the same as in the 1st scenario; in addition, since the user is elderly, in the adapted UI the elements providing output were redundantly supported. As for user interaction, input is vocal, prompt and feedback are redundantly rendered. This was done to offer more interaction flexibility to users.
- *Adaptation 5*: the initial UI was a mono-modal (vocal) UI; then the environment became noisy, and the adapted UI was a multimodal one in which: i)the system output is graphically provided; ii) for interaction elements, the user input and the associated prompt was provided graphically, while the feedback was provided in a redundant manner (both graphically and vocally).

Ten volunteers (4 females) aged 20 to 44 (M = 31.2, SD = 6.5) participated in the study. 7 out of 10 had never used a multimodal (graphical + vocal) UI before, while the others used it in Web environments (e.g. Opera or NetFront), or tried multimodal UIs in some interactive installations. As for participants' educational level: 4 held a PhD, 2 a Master Degree, 3 a Bachelor and 1 a High school degree.

As for the tasks of the test, users had to search for a car using the Car Rental application presented before. Then, upon receiving the result of their search, they were instructed to act as if they were unsatisfied with it and then they had to modify the search. Between the first and the second search the context changed, and the UI was adapted accordingly. Therefore, the second search was carried out by users by using an adapted UI.

The test started with a short presentation in which general information about the test was given to each subject by the moderator. After, each participant answered to general questions about their education, age and experience in using multimodal UIs. Most users were novices with multimodal UIs. Then, in order to get familiar with such UIs, each subject performed a set of practice tasks equivalent to those that would be presented to them in the main part of the experiment, e.g. fill in a form and then submit it by using a multimodal (graphical and vocal) user interface (different from the one used for the test). After this, each participant actually started the test. After experiencing each adaptation, users had to rank it according to a set of relevant criteria identified after reviewing the state of art on adaptive UIs: *Appropriateness*, whether the system selected an appropriate adaptation strategy; *Continuity*, to what extent it was easy for the user to continue the interaction after adaptation; *Transition*, to what extent the adaptation process allowed users to realise what was happening during adaptation; *Impact in decreasing interaction complexity*, to what extent a decrease in the interaction complexity was achieved; *Impact in increasing user satisfaction*, to what extent adaptation increased user satisfaction. Each criterion was ranked according to a 1-5 scale (1: the worst rating, 5: the best one).

All the users successfully completed the expected tasks. In terms of *appropriateness*, the best one was Adaptation 1 (M = 4.5, SD =0.7), the worst one was Adaptation 4 (M = 3.6, SD = 1.2). For Adaptation 2, users appreciated that the UI could switch to the only-vocal modality, given the supposed contextual situation of a user starting to walk fast. Nevertheless, considering the search task proposed in the test, participants appreciated the possibility of further analysing the search list (e.g. by comparing different results) also graphically, as it could be challenging doing that by just using the vocal modality. Thus, some users preferred to maintain the modality that better suites the task rather than the current context. Furthermore, after applying Adaptation 4, the user had to graphically select the field to complete, while the associated input was vocally expected. Even if users were fully aware of what Adaptation 4 would imply (since during the adaptation transition a message was given to users informing them of the changes occurring in the UI), that solution was not natural for the users: the moderator observed that after clicking the input field, the users started typing the value. A similar situation occurred for the vocal modality: if fields were automatically read by the TTS, the user replied vocally to the prompt. As for *continuity*, the best rate was for Adaptation 1 (M = 4.9, SD = 0.3), the worst one was for Adaptation 4 (M = 4, SD = 0.9). The latter one can be explained as before: mixing the modalities for selecting and completing a field was perceived as a factor interrupting the natural flow in which the users interact with the application before and after the adaptation. Regarding *transition*, Adaptation 1, Adaptation 2 and Adaptation 3 (M = 4.6, SD = 0.5) were rated the best, the other 2 were rated the worst, although they received quite positive scores: Adaptation 4 (M = 4.4, SD =0.5); Adaptation 5 (M=4.4, SD=1). In any case, users appreciated that the application showed a message before adapting its interactive behavior, to explain how the UI was going to change. When switching between modalities (e.g. from graphical to vocal), some users complained that, if walking fast (as in Adaptation 2), they should also have received a vocal notification. An *overall rating* was calculated from the ratings received by each adaptation rule on the various dimensions. As a result, the best adaptation was Adaptation 5 (M = 4.4, SD = 0.7), the worst one was Adaptation 4 (M = 3.8, SD = 0.9). Users appreciated Adaptation 5 as it moved from an unfamiliar only-vocal UI to a more flexible, multimodal UI. Alongside, the only-vocal input provided by Adaptation 4 was found a bit limiting, especially when the vocal recognition support did not work satisfactorily. Overall, users appreciated adaptations. This was positive because the majority of them had never used a graphical + vocal UI before the test.

9 Lessons Learnt

From the evaluation test we derived some more general findings about how to handle multimodality in adaptive applications and in particular how to select appropriate modality allocations when the UI adapts to a context change. One of the lessons learnt was that the selection of a particular modality should not be automatically fixed depending on e.g. the current context of use, but users should still have control over the adaptation process. In our test the users, while acknowledging the appropriateness of using particular modalities in specific contexts (i.e., in noisy environments the graphical modality is more effective than the vocal one), also highlighted the importance to maintain the control over the adaptation process to some extent (e.g. by selecting a "preferred" interaction modality and/or by directly controlling the adaptation strategy). Therefore, flexible environments enabling users to have some control over the performed adaptations are advisable. Moreover, it came out that when adapting the UI, the selection of the most appropriate modalities should also take into account the tasks to support, beyond the current context of use. As an example, the result of a search should also be graphically rendered in order to have an overall picture of the gathered data and then enable the user to perform on those data further refinements, filtering and comparisons in an effective manner. Rendering the result of a search only in a vocal manner could not support the same tasks in the same effective way.

Another lesson was that users tend to see the level of single UI elements as the finest UI granularity for assigning different interaction modalities. Attempts to go further in depth with finer adaptations might cause confusion. In our experiment we noticed that mixing modalities at the granularity of *parts* of single UI elements was not always appropriate. The case of a text field graphically selected and vocally filled in was illustrative: after graphically selecting one UI element, users expected to use the same (graphical) modality to interact with that UI element, and they were a bit puzzled when this did not happen.

10 Conclusions and Future Work

In this paper we have presented a novel solution for providing adaptive multimodal applications in Web environments, and report on an example application and a first user test. The new generator of multimodal Web applications is able to exploit current browsers (Chrome) and will soon be publicly available.

We have also reported on a first user test that provided some useful feedback about how users perceive the multimodal adaptation. We plan further empirical validation for better investigating these aspects and how they impact the user experience.

Future work will be dedicated to exploiting more effective and sophisticated mechanisms to increase intelligence in processing adaptation rules. In addition, we plan to carry out further assessment of our approach also involving designers/developers.

Acknowledgements. This work has been partly supported by the EU SERENOA STREP project (EU ICT FP7-ICT N.258030).

References

1. Avouac, P.-A., Lalanda, P., Nigay, L.: Autonomic management of multimodal interaction: DynaMo in action. In: Proceedings EICS 2012, Copenaghen, pp. 35–44. ACM Press (June 2012)
2. Bongartz, S., Jin, Y., Paternò, F., Rett, J., Santoro, C., Spano, L.D.: Adaptive User Interfaces for Smart Environments with the Support of Model-Based Languages. In: Paternò, F., de Ruyter, B., Markopoulos, P., Santoro, C., van Loenen, E., Luyten, K. (eds.) AmI 2012. LNCS, vol. 7683, pp. 33–48. Springer, Heidelberg (2012)
3. Celentano, A., Gaggi, O.: Context-aware design of adaptable multimodal documents. Multimedia Tools Appl. 29(1), 7–28 (2006)
4. Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., Young, R.: Four Easy Pieces for Assessing the Usability of Multimodal Interaction: the CARE Properties. In: Proceedings INTERACT 1995, pp. 115–120 (1995)
5. Duarte, C., Carriço, L.: A conceptual framework for developing adaptive multimodal applications. In: Proceedings of IUI 2006, pp. 132–139. ACM Press (2006)
6. Ertl, D.: Semi-automatic multimodal user interface generation. In: Proceedings EICS 2009, pp. 321–324. ACM Press (July 2009)
7. Manca, M., Paternò, F.: Supporting Multimodality in Service-Oriented Model-Based Development Environments. In: Forbrig, P. (ed.) HCSE 2010. LNCS, vol. 6409, pp. 135–148. Springer, Heidelberg (2010)
8. Nebeling, M., Speicher, M., Norrie, M.C.: W3Touch: Metrics-based Web Content Adaptation for Touch. To Appear in Proceedings CHI 2013, Paris (2013)
9. Octavia, J.R., Vanacken, L., Raymaekers, C., Coninx, K., Flerackers, E.: Facilitating adaptation in virtual environments using a context-aware model-based design process. In: England, D., Palanque, P., Vanderdonckt, J., Wild, P.J. (eds.) TAMODIA 2009. LNCS, vol. 5963, pp. 58–71. Springer, Heidelberg (2010)
10. Paternò, Santoro, C., Spano, L.D.: MARIA: A Universal Language for Service-Oriented Applications in Ubiquitous Environment. ACM Transactions on Computer-Human Interaction 16(4), 19:1-19:30 (2009)
11. Peissner, M., Häbe, D., Janssen, D., Sellner, T.: MyUI: generating accessible user interfaces from multimodal design patterns. In: Proceedings EICS 2012, pp. 81–90. ACM Press (2012)
12. Sottet, J.-S., Ganneau, V., Calvary, G., Demeure, A., Favre, J.-M., Demumieux, R.: Model-driven adaptation for plastic user interfaces. In: Baranauskas, C., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4662, pp. 397–410. Springer, Heidelberg (2007)
13. Stanciulescu, A., Limbourg, Q., Vanderdonckt, J., Michotte, B., Simarro, F.: A transformational approach for multimodal web user interfaces based on UsiXML. In: ICMI 2005, pp. 259–266 (2005)

The Impact of a Mobile Information System on Changing Travel Behaviour and Improving Travel Experience

Tim Skelley, Abdallah Namoun, and Nikolay Mehandjiev

Centre for Service Research
Manchester Business School, Booth Street West
Manchester, M15 6PB, United Kingdom
firstname.lastname@mbs.ac.uk

Abstract. Our cities are struggling. Increasing growth in urban cohabitation has strained cities' systems and infrastructural capacity, especially in regards to the transport domain. This paper investigates the impact which mobile information systems can have on travelling behaviour. In our investigation we employ Occy, a simple web-based mobile application which provides bus timetable information to commuters. In this study, we studied the adoption, usefulness and impact of Occy on users' commuting behavior over a three week period through a diary study which included 25 participants on one of the busiest bus corridors in the UK. This was consolidated by a separate focus group, which included 8 participants, to shed light on participants' acceptance of Occy as an intervention for altering their travel behaviour and improving travel experience. The results show that transport information delivered through Occy did influence the behaviour of travellers, who started trusting the system and used it to target specific busses and thus reduce their bus waiting times. The follow-up discussions identified the features of flexibility and trust as crucial to the successful adoption of such mobile applications.

Keywords: Mobile Application, Urban Transport Information System, Diary Study, Focus Group, Travel Behaviour.

1 Introduction

"Necessity is the mother of invention" [3]. Urban environments generate many logistical challenges for their growing populations, challenges that consequently act as catalysts for innovation. This paper reports on a project which investigates the effect innovative solutions in general and mobile information systems in particular can have on these issues, specifically within the field of Urban Transport Information Systems (UTIS). We believe that mobile UTIS can contribute towards improving the efficiency and utilisation of current transit infrastructure, thus helping to meet top societal priorities, namely economic, environmental and mobility challenges.

In order to manage the mobility demands of a growing populace, cities are increasingly reliant on UTIS to inform and influence travel decisions [1]. UTIS can perform one of two roles, either improving a traveller's efficiency by dictating the quickest

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 233–247, 2013.

route or managing resource utilisation across the whole transport system in order to resolve urban challenges on a macro scale. Whilst both of these objectives can never be fully achieved in parallel (as one demands an agile network and the other lean), a more effective compromise can be found than currently exists. Regardless of which goal is desired, it will only be realised through behavioural change. To that end, this research studies the impact of a small-scale UTIS on users' behaviour, and identifies key features of UTIS that improve system acceptance.

This investigation aims to recognise, measure and reflect on the impact of mobile UTIS on the behaviour of public transport users. This research uses changes in bus waiting times as an indicator for the influence of a UTIS on behavior, since in our setup travelers who start to trust the system would change the time they leave home to target a specific bus and thus would lower their waiting times. The project also seeks to gain a clearer understanding of the effect that mobile technological developments have on the acceptance of UTIS. In order to achieve these aims, the project will seek to answer two research questions.

The first research question (RQ1), "Do mobile Urban Transport Information Systems impact the behaviour of travellers?", considers the effect that mobile UTIS have on users of transport systems, looking to support the hypothesis (H1) that "mobile UTIS adoption would lead to reduced bus waiting times".

The second research question (RQ2), "What is the impact of new technology on the acceptance of UTIS?", investigates the way new technology is leaving its mark on the future of UTIS, with the hypothesis (H2) that "New technology is strengthening key features that lead to UTIS acceptance". New technology in this context should be understood to mean the hardware capabilities and software platforms facilitating current advances in smart-phone software development.

The remainder of this paper is organized as follows. Section two reviews the related literature about behaviour change and travel patterns. Section three details the procedure and methodology of the studies of this research, and presents the results. Section four discusses the implications of using UTIS to change travel behaviour and improve travel experience.

2 Related Work

"Traveler behaviour is the process of individual decision making about what trips to make, where to visit, when to depart, what mode of travel to utilize and what route to follow" [5]. This definition concisely introduces the core theme of our first study, which aims to further understand the impact that mobile UTIS can have on traveller's behaviour.

Mobile applications have become ubiquitous and are increasingly being used to positively change human behaviour. Their application spans a wide range of domains including the health domain [15, 16] and the transport domain [17]. The literature has already identified and discussed in detail a number of factors that contribute towards travel behaviour change in the form of weather forecasts [18], integrated multimodal travel information [19], service-related traffic [20], road pricing [22], rewards for commuters [21], and travel feedback programs [23]. However, our research extends previous findings by systematically discovering the influence of accessing bus information via a mobile application on travel behaviour change and travel satisfaction.

Research undertaken by Robinson et al. [4] on behalf of the National Cooperative Highway Research Program examines passenger use, perception and response to UTIS. Their research approach was to form traveller profiles within four cities across the U.S, using a range of data gathering methods including surveys, interviews, focus groups and travel logs. This comprehensive 'tool box' of techniques enabled the research to develop a holistic view of participants, producing more informed insights. The samples selected were well populated (over 2000 participants across all the research activities and locations) and mixed in both age and gender, with city locations spread across the U.S. The NCHRP's research method aligns closely with this research's chosen methodology, especially in its use of travel logs and focus groups. The NCHRP report measures the behavioural effect of a number of transport information systems on drivers.

The study required participants to record, by hand, any information sources used en route and if these influenced their route choice. Potential limitations of this approach include poor participant recollection of where information was attained when recording data sources, leading to ambiguity regarding which sources influenced decision making. A more effective method of measuring the impact on traveller decisions would have been to withhold all travel information during a control period, before permitting access to a single source of travel information. This strategy would have enabled analysis of traveller behaviour with the alteration of a single variable.

The results generated by the focus groups regarding pre- and en route decision making were too closely associated with road travel to generalise to public transport. Nonetheless, TV and radio were found to be the most common form of travel information source used to make pre- and en route decisions. We can therefore surmise that typical users do not actively seek travel information, but receive it as a consequence of engaging with media for other reasons. The unspecific, infrequent and uncustomised nature of travel broadcasts via TV and radio, especially regarding public transport, demonstrates that travel information is currently being transmitted/received in an inefficient manner.

Similarly, Khattak et al [2] researched the impact of transport information systems on the behaviour of users. Having sampled 5,000 households (adopting a mix of questionnaires, travel logs and interviews), their findings showed that half of travellers do not seek electronic travel information, with the majority that do only taking advantage of one source. The research goes on to substantiate that as the number of data sources and frequency of travel information attainment increase, the likelihood for behavioural change increases dramatically.

Overcoming behavioural inertia within the travelling population is key to improving transport system efficiency and Khattak has identified that this can be achieved, in part, through frequent interaction with travel information received through multiple devices. UTIS are a clear channel to reach this goal, so it is imperative that UTIS are designed to encourage both high access frequency and cross device interaction. However, there are a number of areas of progress still to be made in this field of research.

Firstly, the bulk of current research focuses on the changing decisions of car users and whilst some findings can be generalised, there is still a need to investigate travellers using public transport. Secondly, the current literature focuses on traditional forms of information dissemination (TV, radio and variable messaging signs), overlooking smart-phone operated UTIS. This research provides research addressing the two areas identified as lacking in Khattak's research.

3 Studies

3.1 Methodology and Procedure

In pursuit of our research objectives, we designed and conducted two studies to help us answer our research questions: a diary study and a focus group discussion. A diary study is a non-disruptive research method that necessitates asking users to record their activities and experiences over a period of time by completing paper diaries or web-based diaries [6]. Diary studies are advantageous as they provide a realistic indication of human behaviour, thoughts and activities within a context. In our transport context, where participants are mobile a diary study is a perfect choice since it allows us to capture their feedback on the fly. A focus group, however, is a research method that involves a group of users discussing their opinions, attitudes and views about a certain idea / topic [7]. Focus groups are useful to share and discover varying opinions of participants and discuss certain ideas in depth.

The first research question was addressed by conducting a three-week diary study of a group of travelers, by recording their bus waiting times during their typical commute. A mobile UTIS was introduced, as an intervention, mid-study to measure any change in their bus stop waiting times, and therefore behaviour. Whilst, in the wider context, behavioural change in response to UTIS adoption can take many forms, such as changing route or time of travel, this study focused specifically on how travellers use a mobile UTIS to fine tune the process of catching a bus. In addition to bus waiting times, users' overall satisfaction with the bus service was collected to monitor changes in the perceived quality of service following the introduction of the UTIS. Following these two phases, informal interviews were held with a number of participants to gather more qualitative insight into their experience. The UTIS used in this study was a web-based mobile bus timetable application developed by one of the authors as part of this research; the application was called Occy (see Methods and Instruments).

Using Occy, participants submitted their "bus waiting time" on a scale of 'less than 1' to 'greater than 15 minutes'. Participant's perception of the bus service was also measured on a 5-point scale, with 1 signifying 'Poor' and 5 signifying 'Perfect', which indicated their satisfaction with the service provided.

The second research question was investigated using a qualitative approach. Initially, a focus group discussed the research question for 50 minutes, with guidance from one of the authors. Thematic analysis, an inductive technique for identifying patterns within qualitative data [8], was then applied to identify important concepts from the meeting's transcription. The validity of these themes in contributing to UTIS acceptance was tested by a questionnaire using Davis' Technology Acceptance Model (TAM) [9] as a framework on which to structure the questions and analysis. Results from the questionnaire were then analysed for evidence that relationships exist between the themes identified and factors within the TAM. This investigation was different from that conducted within the RQ1 investigation. It included additional discussions on technologies that were excluded from the RQ1 discussions because of time constraints on that first phase.

3.2 Participants

The diary study included a total of 25 student participants, all of whom live and study along Manchester's Oxford/Wilmslow Road corridor, one of busiest commuter corridors in the UK. All participants were also selected to normally use the bus service to commute and to have access to a smart-phone. The demographic of the study's sample (25 participants) ranges from first to final year, male and female students living near different bus stops and studying a range of courses, with two thirds of the students studying at the Manchester Business School.

The focus group included a total of 8 students from the University of Manchester, ranging in nationality and age (they were selected as a subset of the diary study sample). By contrast the questionnaire used to help us answer the second research question was completed by a random sample of 29 participants. Participants in both studies were incentivised to guarantee their sustained participation by entering them into a draw for online vouchers.

3.3 Materials and Instruments

Web-Based Bus Timetable Application (Occy)

In order to attain a smart-phone based UTIS fulfilling the specific requirements demanded by this study (diary-study submissions, multi-device compatibility and route specific travel information) the only viable option was to develop one. Occy, a web based mobile bus timetable application, was developed using the Waterfall Development Life Cycle [10].

End user interviews with 9 students (these were different from the participants of the studies) and analysis of exiting UTIS reviews were conducted to gather functional and non-functional requirements. Below are the must have requirements identified for our mobile UTIS.

- Accessibility via all smart-phone devices (Non-Functional).
- A user-friendly and quickly navigated interface, allowing users to select their current location and direction of travel (Non-Functional).
- A database offering accurate timetable information for weekday, term time travel times between Fallowfield and the University's campus (Functional).
- Fast loading time (Non-functional).
- In-application diary study submission (Functional).

Design: next, wireframe and rich picture designs were produced based upon the requirements specified and the constraints of mobile interfaces, as depicted in Fig. 1. The key UI elements that constituted Occy include:

- A logo (A) to identify the app and link to an 'About Occy' screen.
- The background (B) will be minimal to reduce the loading time for users with poor network coverage.
- A Bus Stop Selector (C) to list the available bus stops in a drop down menu format, which meets Neilson's error prevention heuristic.
- A Destination (D) for the user to indicate direction of travel.

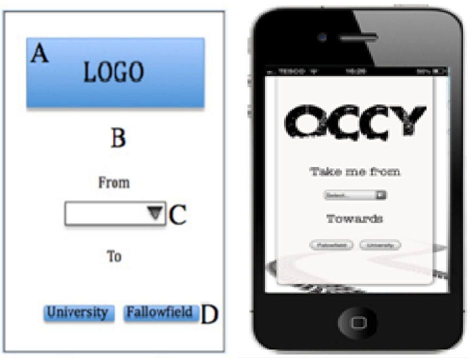


Fig. 1. Wireframe Designs of Input Screen of Occy

The final design can be seen in Fig. 2. Occy's sparse layout not only meets the functional requirements of a user-friendly interface but also delivers just the right information to the user in a clear and intuitive format. It is also data light so that poor reception will not inhibit the application's use.

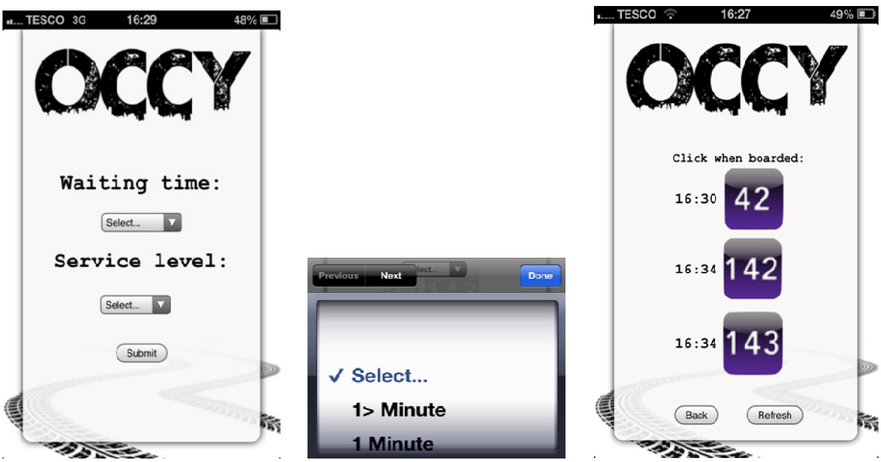


Fig. 2. Final Designs of Occy

Implementation: Web applications sit on top of layers of other software called a stack. Stacks come in a number of forms, with a key differentiator between stacks being the use of proprietary and open source software. The configuration used to support Occy was a 'LAMP Stack', which "is a loose collection of open source components that developers can combine to build various types of web applications" [12].

MYSQL was used to store the timetable data from which Occy draws its bus arrival time. Apache, the most popular web server since 1996 [13], is an open source HTTP server, compatible with both UNIX and Windows NT operating systems. This delivers web content to clients on request, using hypertext transfer protocol; using the most common and available web server was a logical choice. A server side scripting language was required to deliver the dynamic content within the application [14]. Whilst there are a variety of packages within the LAMP stack framework providing this function, including Perl and Python [14], PHP was selected for this task owing to its relative ease of use and strong online support. The application itself sits above all of these layers. Occy was written in HTML and enriched with features implemented using JavaScript. These elements, processed on the client side, helped to deliver a number of the more aesthetic components of the interface, such as an 'About Occy' screen, which is presented when the logo was selected. Cascading style sheets were used to format Occy, enforcing interface consistency.

Testing: Occy was tested at a component and whole system level by the developer against the system requirements. End user testing was also used to assess the more qualitative nonfunctional requirements. Results of these tests can be found in Table 1.

Table 1. Features of Occy, the mobile UTIS

Feature	**Functional / Non-Functional**	**Achieved**
Accessibility on various smart-phone devices	F	✓
A user-friendly interface allowing users to select their current location and direction of travel	N	✓
A database offering accurate timetable information for weekday term time travel times between Fallowfield and the university's south campus	F	✓
Fast loading time	N	✓
In application submission	F	✓
Multiple bus stop options beyond the most popular stops (Owens Park and University Place)	F	✓
Functionality outside of the weekday commute to ensure broader usage	F	✓

Focus Group Questions and UTIS Acceptance Questionnaire

We devised a number of questions to initiate and guide the focus group discussion focusing on a number of related areas, such as the use of information systems to travel, the type of communication used to make travel decisions, the motives for changing travel behaviour, and the features that commuters would like to see in existing UTIS. These questions enabled us to acquire a better understanding of the factors that influence user acceptance of mobile UTIS.

Next, we devised a 10-item questionnaire to gauge users' acceptance of mobile UTIS based on the Technology Acceptance Model [9]. Each item question was responded to on a 5-point Likert scale, ranging from 'Strongly Agree' to 'Strongly Disagree'. The questionnaire measured user perception towards five main constructs: perceived usefulness, perceived ease of use, flexibility, trust, and adoption (see Table 3).

3.4 Results

Study One Results

202 diary study entries were received from the 25 participants during this investigation's 3-week diary study; an average of 56% participation per person was recorded. Submission rate was initially high, totaling 22 per day on the second day of the study, however fell to an average of 12.6 responses a day for the remainder of the study.

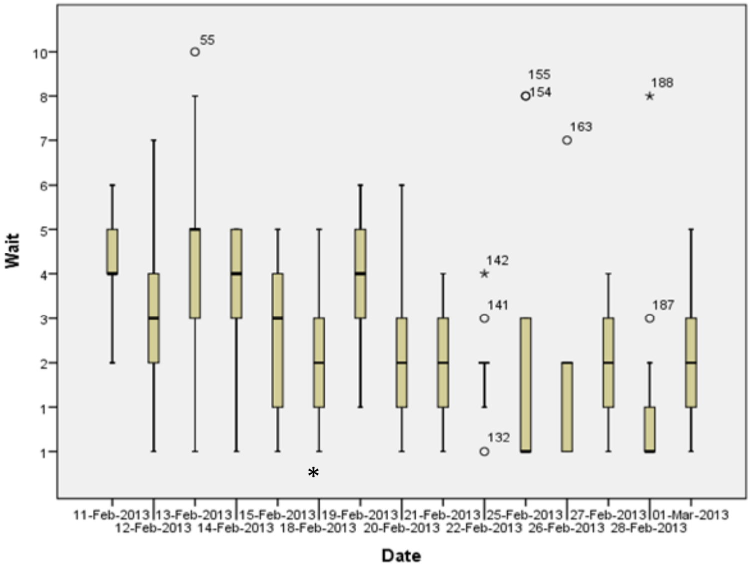


Fig. 3. Waiting Time Duration and Time Boxplot

Fig. 3, a boxplot chart, presents the daily waiting time of travellers over the 3-week research period, showing extremities of each day's submissions as well as the mean and upper and lower quartiles. The standard deviation across the results was 1.9 minutes, indicating that outliers were anonymous and that the average range of results was relatively limited. We can observe that on most days, at least one participant had a wait of one minute or less, with the longest wait of 8 minutes occurring on the third day of the study. A small spike in waiting time can be seen on the 27.02.13, as the median rises to 2 minutes. The key finding, however, is that the average waiting time in Phase I of 3.7 minutes dropped to 2.2 minutes in Phase II, once Occy was introduced. This strong relationship is mirrored in an equally statistically significant negative correlation between waiting time and study duration of r=-0.84. This means that as time passed and Occy was introduced, there was a noticeable alteration in bus waiting time.

Using Spearman's Rho a moderate correlation of 0.47 was identified between traveller satisfaction and the study duration. This can be understood to mean that as Occy was introduced, satisfaction levels improved.

Following the 3-week research period, 10 of the most responsive participants were invited to an informal interview regarding their experience using the mobile application. Below are the key outcomes from these participant interviews:

- Participants reported a noticeable reduction in their waiting time during Phase II.
- Participants confirmed waiting to be the greatest controllable variable affecting both their travel satisfaction and the efficiency of bus travel.
- Participants reported greater satisfaction in line with improved waiting times and expectation management during Phase II.

Study Two Results

The discussions of the focus group study were recorded, transcribed and analysed. Table 2 lists the overarching themes identified through thematic analysis [8] of the focus group's discussion.

Table 2. Emerging Themes from Focus Group Results

Flexibility	**Trust**
Real-time Being able to access information any time and for it to be up to date, enabling last minute or en route decision making that doesn't inhibit efficiency of travel. *Internet connected services and users*	Reliability of Implementation Belief that a UTIS was not going to fail in its delivery of travel information through poor development. *Applications approved and rated on open market places e.g. through Apple's App Store*
Location Based Being able to have a route mapped out for the traveller, dependent on their current location. *Location based services*	Reliability of Data Having data that can be relied upon for all travel needs, without secondary confirmation. *Data integrity checks*

Table 2. (*Continued*)

Flexibility	**Trust**
Variety of interface Participants valued being able to choose what format their travel information was displayed in, with some preferring maps to text. This also included the various form factors UTIS are available on, i.e. tablet or smart-phone. *Cross device applications*	Anonymous Travel Feedback Participants felt that the use of travel data from their smart-phone would increase travel information accuracy and they would happily permit access, provided the data was anonymous, and trusted that the UTIS wouldn't be hacked. *Providing anonymous data to support UTIS*
Alerts Receiving routine and major event notifications specific to users' typical travel patterns, which inform travel decisions. This reduces the need to actively research travel information. *Push Notifications*	Sources Participants were more likely to trust travel information from multiple sources, including crowd-sourced. They suggested a rating system indicating the trustworthiness of information sources. *UTIS fed by big data*
Customisation Being able to enter travel preferences regarding when and how they like to travel for personalised travel information. *Personalisation of mobile applications*	

The10-item questionnaire structured around the Technology Acceptance Model was completed by 29 participants. All questions were rated on a 5-point Likert scale, with '1= Strongly Disagree' and '5= Strongly Agree'. The internal consistency of these results was examined using Cronbachs Alpha Coefficient [11], with results showing that most scales had good reliability (> 0.72) apart from perceived usefulness (0.64). The average ratings of participants' acceptance of UTIS showed positive perceptions and attitudes towards Occy, where 'usefulness' and 'ease of use' of public transport applications were highly rated, m=4.27 and m=4.13 respectively. Participants also agreed that they are now using public transport applications more than before, m=3.86. The results of the questionnaire are listed in Table 3.

Table 3. Mobile UTIS Acceptance Questionnaire Results, STD = standard deviation

Concept	Question	Mean	STD
Flexibility	Public transport applications have become more tailored to my requirements over the last 3 years.	3.72	1.16
	Public transport apps I use for making travel decisions have become increasingly irrelevant in their content.	2.62	1.14
Trust	I trust the information provided by newly downloaded public transport applications.	3.75	1.12
	It takes me a long time to trust public transport apps enough to base important travel decisions on them.	2.48	1.35
Perceived Usefulness	Public transport applications have become more useful over the past 3 years.	4.27	0.99
	Over the past 3 years the time spent using public transport applications has become less worthwhile.	2.10	0.90
Perceived Ease of Use	Public transport apps become easier to use over the last 3 years.	4.13	0.69
	Public transport apps in the last 3 years have become harder to operate.	2.27	0.79
Adoption	I use more public transport applications than 3 years ago.	3.86	1.05
	The frequency with which I use public transport apps declined in the past 3 years.	2.24	0.95

The covariance of each of these factors was tested using Spearman's Rho correlation coefficient. All factors were positively correlated, with flexibility and perceived usefulness showing the strongest correlation, r=0.77, p=0.001. The correlation results of this analysis are illustrated in Fig. 4.

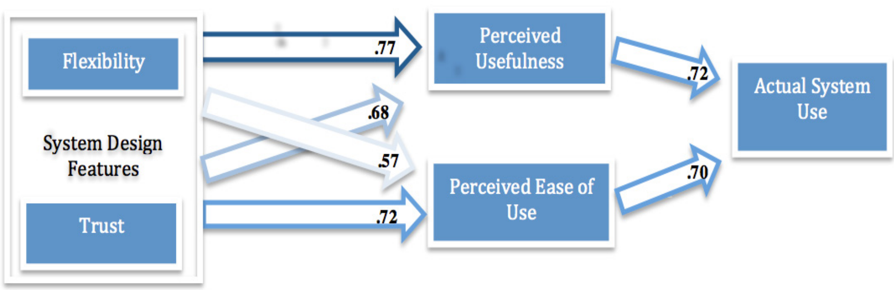


Fig. 4. Mobile UTIS Acceptance Model

4 Discussion and Implications

4.1 Impact of Mobile UTIS on Bus Waiting Time

The relationship between waiting time duration and the passing of time (during which period Occy was introduced) was a strong negative correlation of -0.84, supporting the finding that as Occy was introduced times were reduced by 40 percent. It can be deduced that during Phase I of the study, Participants arrived at their bus stop with enough spare time for the journey and additional time in anticipation of having to wait for a bus to arrive. This is seen in an average waiting time of 3.7 minutes, however during Phase II there was a marked reduction in the waiting time of participants. It can be concluded that as participants gained access to bus arrival times they could make informed decisions about when to leave for their buses. This eliminated waiting time from their total journey time, on average saving 1.5 minutes per journey. This finding is replicated in the post-study interviews, in which Participants reported a reduction in waiting time. Crucially, 90% of participants also endorsed the assumption made by H1, that reducing waiting time is the key to travel efficiency.

The findings from this investigation indicate that H1 is supported by the results produced by the diary study. This means that in response to the RQ1, "Do urban transport information systems impact the behaviour of travellers?", an answer can be given that yes, UTIS have the ability to influence travellers' decision making, therefore yes, UTIS influence travel behaviour.

4.2 Technology and Features Influencing UTIS Acceptance

Two key themes (flexibility and trust) were identified from the focus group discussions regarding important technological advancements (typically within mobile devices) and how they are improving UTIS acceptance.

- ***Flexibility***: flexibility encompasses UTIS features, such as location aware technologies and personalisation of travel preferences.
- ***Trust***: trust represents the concept of a user's belief that the system will deliver them to their destination on time and using the most efficient route. Constituting technology such as crowd sourced data and application validation.

These themes are also supported by the findings of Robinson et al [4], who identified that users and service providers agree UTIS should be flexible, trustworthy, and reliable. The validity of trust and flexibility as genuine stimuli for UTIS acceptance was examined by adapting Davis' TAM [9] into a UTIS Acceptance Model before testing the relationships between each of the elements using a questionnaire.

Figure 4 shows that each relationship within the UTIS acceptance model has a positive correlation, supporting H1, however some relationships are stronger than others. Flexibility can be seen to have a strong positive correlation of 0.77 with a UTIS' perceived usefulness. We can conclude from this that UTIS incorporating features such as location aware technology, personalised travel preferences and real-time travel disruption alerts are perceived to be more beneficial to the user. The relationship between flexibility and perceived ease of use is the weakest correlation in the UTIS Acceptance Model, at 0.57. The cause of this moderate relationship may be due to users misconstruing flexibility as complexity, given the greater freedom to customise.

Trust, it can be observed, holds a strong correlation with a user's perceived ease of use. This may be because as UTIS are perceived to be more trustworthy, the risk borne by the user in relying on the UTIS is reduced, consequently increasing the perceived ease of use. As Technology Acceptance Model suggests [9], the relationship between a system design feature (in this context, trust) and a system's perceived ease of use is a driving force in promoting actual system use. This is supported within the UTIS acceptance model by the statistically significant correlation of 0.70 between perceived ease of use and actual system use.

Therefore, in response to the second research question, "What is the impact of new technology on the acceptance of UTIS?", a response can be given stating that new technology has, and will, enable flexibility and trust within UTIS, both of which improve the acceptance of the system.

Taking the findings forward, we may stipulate that travellers are willing to change their behaviour based on mobile UTIS. However, there are caveats that must be taken into consideration. Firstly, UTIS must be flexible in order to deliver personalised, real-time content to users. Secondly, they must be reliable and accurate to secure the trust of users, without which at best, they become redundant. Lastly, drawing on Khattak work [5], users must be presented with data on a frequent basis across multiple devices. These system requirements can be met through exploiting developing technologies such as crowd-sourced data, location-based services and cross device applications.

UTIS' increasing influence will enable a two-way interaction between travellers and transport networks, facilitating a smarter compromise between an agile user-centric system and a lean network-focused system, with the result of improved mobility and reduction in environmental impact, both of which have a resolutely positive economic result.

5 Conclusion and Future Directions

This paper recognises the economic, mobility and environmental expectations placed on cities and the positive impact that improved public transport networks can have on all three. Having identified UTIS as a potential means of improving the efficiency of these networks, our study investigated this theme, focusing on two research questions. The first question aimed at substantiating the claim that UTIS adoption has a positive impact on the efficiency of travellers' journeys, thereby demonstrating an impact on travel behaviour. The second question extended this by considering the key features that strengthen UTIS' acceptance by commuters. This research did not only identify whether UTIS are best used to improve personal travel efficiency or the efficiency of a whole travel network, but rather whether they can influence travel behaviour to achieve these ends.

The findings showed that our mobile UTIS had a marked impact on the behaviour of travellers in this case resulting in more efficient journeys for individual travellers, and also improved their satisfaction with the bus service. Design features, grouped within the themes of flexibility and trust, were identified and validated as key to UTIS' acceptance. Features within these two themes improve users' perception of UTIS' usefulness and ease of use, leading to actual system use and should therefore be incorporated into future UTIS' development.

In the future we plan to test our mobile UTIS in other road corridors with different transport characteristics and capacity. We will extend the functionalities of Occy to include real-time bus news and traffic updates, where travellers will be warned of transport delays and disruptions as they occur. We also aim to increase the size of our sample and extend the length of the study to investigate travel behaviour change over a prolonged period of time.

References

1. BBC News. BBC Click (April 28, 2013), BBC iPlayer: http://www.bbc.co.uk/iplayer/episode/b01sb437/Click_27_04_2013/ (retrieved April 28, 2013)
2. Khattak, A., Pan, X., Williams, B., Nagui, R., Fan, Y.: Traveler information delivery mechanisms: Impact on consumer behavior. Transportation Research Record (2069), 77–84 (2008)
3. Plato. Republic. Ancient Greece (380 BC)
4. Robinson, E., Jacobs, T., Frankle, K., Serulle, N., Pack, M.: Deployment, Use, and Effect of Real-Time Traveler Information Systems. NCHRP. Transportation Research Board, Washington (2012)
5. Khattak, A., Koppelman, F., Schofer, J.: Stated preferences for investigating commuters' diversion propensity. Presented at the 71st Annual Meeting of Transportation Research Board, Washington D.C., vol. 20, pp. 29–49. Transportation (January 1992)
6. Mariño, R., Minichiello, V.: Reporting of events using diaries. In: Sullivan, G., Minichiello, V. (eds.) Handbook for Research in Health Sciences. Addison-Wesley, Sydney (1999)
7. Vaughn, S., Schumm, J.S., Sinagub, J.: Focus Group Interviews in Education and Psychology. SAGE Publications, Thousand Oaks (1996)
8. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qualitative Research in Psychology 3(2), 77–101 (2006)
9. Davis, F. D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Q. 13(3), 319–340 (1989)
10. Royce, W.W.: Managing the development of large software systems: concepts and techniques. In: Proceedings of the 9th International Conference on Software Engineering (ICSE 1987), pp. 328–338. IEEE Computer Society Press, Los Alamitos (1987)
11. Cronbach, L.: Coefficient alpha and the internal structure of tests. Psychometrika 16(3), 297–334 (1951)
12. Lawton, G.: LAMP Lights enterprise development Effort. Computer, 18 – 20 (2005)
13. Apache HTTP Server Project (2012, n.d n.d), apache.org: http://httpd.apache.org (retrieved April 14, 2013)
14. Lee, J., Ware, B.: Open Source Web Development With Lamp: Using Linux, Apache, Mysql, Perl, and Php. Addison-Wesley Professional, Boston (2002)

15. Mansar, S.L., Jariwala, S., Shahzad, M., Anggraini, A., Behih, N., AlZeyara, A.A.: Usability Testing Experiment For A Localized Weight Loss Mobile Application. Procedia Technology 5, 839–848 (2012)
16. Fjeldsoe, B.S., Marshall, A.L., Miller, Y.D.: Behavior Change Interventions Delivered by Mobile Telephone Short-Message Service. American Journal of Preventive Medicine 36(2), 165–173 (2009)
17. Centieiro, P., Romão, T., Dias. A. E.: A location-based multiplayer mobile game to encourage pro-environmental behaviours. In: Romão, T., Correia, N., Inami, M., Kato, H., Prada, R., Terada, T., Dias, E., Chambel, T. (eds.) Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology (ACE 2011). ACM, New York (2011)
18. Cools, M., Creemers, L.: The dual role of weather forecasts on changes in activity-travel behavior. Journal of Transport Geography 28, 167–175 (2013)
19. Grotenhuis, J.W., Wiegmans, B.W., Rietveld, P.: The desired quality of integrated multimodal travel information in public transport: Customer needs for time and effort savings. Transport Policy 14(1), 27–38 (2007)
20. Hebes, P., Menge, J., Lenz, B.: Service-related traffic: An analysis of the influence of firms on travel behaviour. Transport Policy 26, 43–53 (2013)
21. Ben-Elia, E., Ettema, D.: Rewarding rush-hour avoidance: A study of commuters' travel behavior. Transportation Research Part A: Policy and Practice 45(7), 567–582 (2011)
22. Rouwendal, J., Verhoef, E.T.: Basic economic principles of road pricing: From theory to applications. Transport Policy 13(2), 106–114 (2006)
23. Fujii, F., Taniguchi, A.: Determinants of the effectiveness of travel feedback programs—a review of communicative mobility management measures for changing travel behaviour in Japan. Transport Policy 13(5), 339–348 (2006)

A Lightweight Architecture for the Web-of-Things

Tor-Morten Grønli[1], Gheorghita Ghinea[2], and Muhammad Younas[3]

[1] Norwegian School of IT, Oslo, Norway
tmg@nith.no
[2] Brunel University, London, UK
george.ghinea@brunel.ac.uk
[3] Department of Computing and Communication Technologies,
Oxford Brookes University, Oxford, UK
m.younas@brookes.ac.uk

Abstract. The past few years have marked a shift in Web development as users have become accustomed to Web applications with dynamic content and enhanced user experience. New emerging protocols and standards seek to provide increased flexibility by making available new models of interaction to Web applications. One such application is the Web of Things. In this paper, we propose a new lightweight architecture for the Web of Things, based on RESTful approaches. We further show, through a proof of concept application, taking a smart city as its context, how new technologies can be combined to support our proposed architecture and application development for the Web of Things. We argue that the use of protocols and standards such as WebSocket, WebSocket API, Server-Sent Events and JSON, the JavaScript Object Notation, can make the vision of the Web of Things a reality.

Keywords: Web of Things, Architecture, Lightweight, Node.js, JSON, Smart city.

1 Introduction

Users have long been accustomed to consuming services offered over the Internet using web browsers. However, with Apple and Google's launch of the smartphone era in 2007-8, users have also started to consume services and acquire vast amounts of information when they are on the move. This trend has grown exponentially since then, with smartphones dominating today's cell phone markets. A separate but related trend has been that of an increasing number of hitherto standalone devices and sensors now being endowed with wired/wireless connectivity. Together, these two have given rise to the Internet of Things (IoT), a global information network consisting of trillions of smart objects in the world we live in [1].

Whilst the IoT has identified application areas such as supply chain management [2], healthcare [3,4], and workplace support [5], issues raised by the new infrastructure include managing and making sense of the massive amount of data generated

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 248–259, 2013.

[6,7], privacy and security [8,9], as well as user acceptance of an increasingly monitored and sensor-rich world [10], to name but a few.

In this paper, we propose a new lightweight architecture for the Web of Things, based on RESTful approaches. We further show, through a proof of concept application, taking a smart city as its context, how new technologies can be combined to support our proposed architecture and application development for the Web of Things. We argue that the use of protocols and standards such as WebSocket, WebSocket API, Server-Sent Events and JSON, the JavaScript Object Notation, can make the vision of the Web of Things a reality.

The rest of the paper is organised as follows. Sections 2 and 3 respectively describe the fundamental technologies and research challenges. The aim is to establish a rationale for the proposed approach. Section 4 presents the proposed architecture and section 5 illustrates an application scenario that's implemented within the proposed architecture. Section 6 gives implementation details and also presents experimental results. Section 7 concludes the paper.

2 Web of Things – Basic Concepts and Technologies

This section describes the basic concepts and technologies, which can potentially be used to design and develop Web of Things applications.

2.1 Internet of Things

In its current guise, the IoT is dominated by machine-to-machine communication. The next big leap in the evolution of this communication infrastructure will be when machine to smart object communication is facilitated. The first step along this road is to create smart devices or enrich everyday objects with smart communication capabilities, laying the premises for a smart environment. Smart objects can be recognized by:

- Sensors to measure light, temperature, position etc.
- Information (data) persistence
- Communication capabilities
- Machine-to-Machine communication

IoT embeds intelligence in the components enabling communication, exchange of information, recommendations, make decisions and provide services. Although this has gained significant research and industry interest, the key challenge is to make this widespread, commonly accepted and part of the global Internet. Currently, one such approach is described by Web of Things (WoT). WoT emerges from IoT and continues the vision of embedded technology and smart devices from IoT. This is taken a step further by incorporating the use of standard web technologies and thereby reusing existing, well-accepted standards and practices such as HTTP, URI and REST (REpresentational State Transfer).

Such RESTful technologies [11] have obvious advantages when compared against approaches based on the by now traditional Web Services Architecture (SOAP, WSDL, UDDI) – they are lightweight, loosely coupled, require relatively little computing resources and virtually no operating system support, and scale well. All of this means that one can embed tiny Web servers and, by using the REST architectural style (which is the same as that of the web, i.e. based on HTTP, URIs and HTML/XML) one can build scalable interaction models at the application layer [12, 13].

Thus, a scenario whereby smart objects on the Web start abstracting and describing their services using either XML or JSON (JavaScript Object Notation), which are both human and machine readable, can easily be envisaged. In so doing, such smart objects can be accessed and interacted with via Web browsers. Indeed, assuming a user with appropriate technological skills, there is no reason why such users cannot create mash-ups, combining real-world physical devices with virtual services made available over the Web – all combining towards a Web of Things [14, 15].

2.2 Cloud Computing

Cloud computing has received considerable attention in the software industry. It is an established architecture used for hosting services, virtual machines and web based applications. Cloud computing refers to the applications delivered as services over the Internet and the hardware and systems software in the data centres providing these services. The idea is built around an economy of scale, where the ultimate goal is to provide more resources, better scalability and flexibility for less money. Cloud computing indicates a movement away from computing as a product that is owned and towards computing as a service [16]. Service in this context is a concept that deals with the utilisation of reusable fine-grained components across a vendor's network. The cloud, as an open, distributed, resource-full platform, contains promising opportunities for creating a framework for wrapping WoT solutions. To this end Kovatsch et al. [17] proposes, in analogy with the thin client concept, that of the thin server. Here tiny servers embedded on physical devices use RESTful approaches and export only their elementary functionality to application servers running application logic. Thereby separating the application logic from the firmware and devices can export their functionality without hosting application logic, which is all found on the cloud. Application development is thus completely decoupled from the embedded domain.

3 Research Challenges

The future of WoT holds many promising ideas, and generates great research interest from researchers, businesses and industry. All application domains can benefit from

WoT solutions and areas such as disaster prediction; smart homes; medical applications; transport systems; smart cities and security systems are some of the most common examples. Some of the possibilities and research challenges of the WoT and cloud computing are highlighted in the following table.

Table 1. WoT and Cloud properties comparison

Web of Things requisite	**Counterpoint in Cloud Computing**
Dynamic demand of resources	Cloud elasticity
Real-time needs	Service Level Agreement and Quality of Service
Expected growth in application use	Scalability and infrastructure
Security of data	Cloud privacy and security elements
Open, interoperable communication	Cloud standards and architecture

Other and related areas represent as well research challenges for WoT and commonly mentioned are identification of objects, sensing of environment, information ownership, continued absence of standards and security/privacy [18] [19]. Although highly important and relevant, we would like to draw the attention to the somewhat less commonly discussed one level higher of abstraction. The application layer, which allows for the *use* of IoT data/information and which ties everything together in WoT solutions, will be the primarily focus. From Table 1 there are several important issues to deal with and we would like in this article to focus on growth in application use and interoperable communication. We do see all these properties as important, independent research issues with unique attributes. We chose to focus on architectural issues, interoperability and growth as we do think they are of essential value and need to be in place for the technology to gain sufficient importance and adaptation.

4 The Proposed Architecture

The architecture of the proposed framework is represented in Fig. 1. The architecture is multi-layered which includes*: Things Interface Layer, Things Management Layer, Things Service Layer,* and *User/Application Layer.*

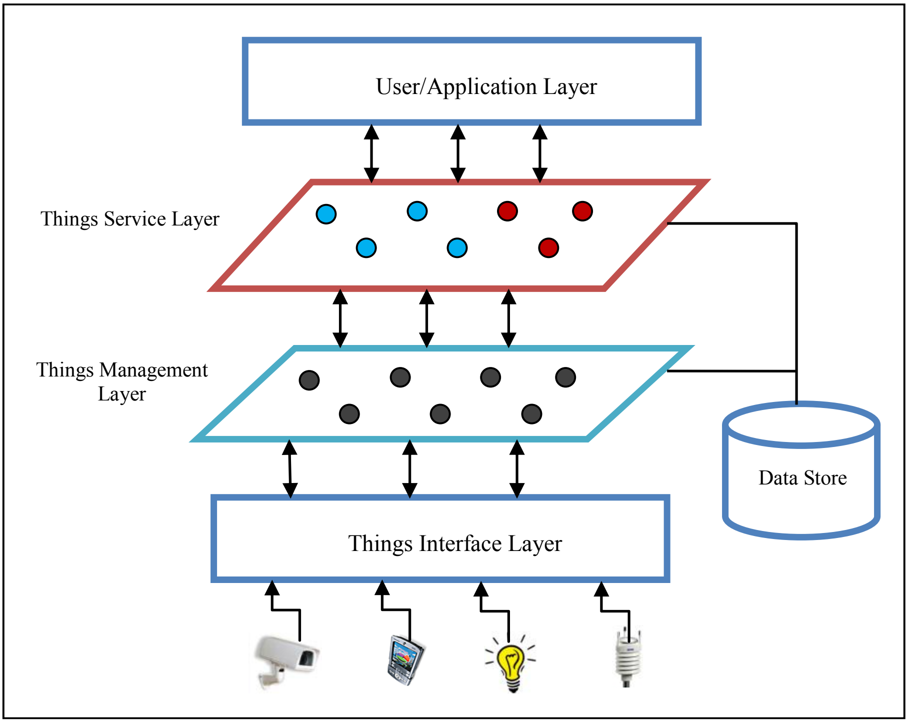


Fig. 1. Architecture of the Proposed Framework

Each of the layers is briefly explained in the following sections.

4.1 Things Interface Layer (TIL)

This layer represents the interface of the proposed system to the physical devices (things), such as interfaces to CCTV camera, mobile phones, light bulbs and sensor devices. There are various tools that provide interfaces to physical devices. In the proposed architecture we make use of Arduino[1] in order to enable communication between the physical devices and the computer systems, to create an interactive environment for communication with (smart) objects. Arduino, a lightweight single-board microcontroller, is an open source electronics prototyping platform which offers interface for communicating with and collecting information from a variety of sensing devices.

4.2 Things Management Layer (TML)

This layer represents the devices at a higher level of abstraction. In other words, it abstracts the data and functionality of the devices into programmable components

[1] `http://www.arduino.cc/`

such as web services. In the proposed architecture, we exploit the RESTful architecture of web services in order to transparently represent the data and functionality of the physical devices and to hide their underlying complexity and low level details. RESTful architecture is more appropriate to the characteristic of the devices in WoT as it is lightweight and loosely coupled. We enable information exchange through the use of JSON structured data, and business logic is applied through common software engineering components.

4.3 Things Service Layer (TSL)

This layer processes the data and functionality of the devices represented as RESTful services. For instance, it interacts with the user layer and receives user requests in order to provide them with required services. Further, this layer can also collect information from the underlying devices and store in the data store using appropriate format and structure such as JSON or XML.

In addition to the RESTful approach, we employ Node.js [4] in the web components. Node.js implements an event-based framework for writing scalable and efficient Web applications using JavaScript. Node.js is based on non-blocking I/O calls and its light footprint makes it an excellent candidate for real-time, distributed and data-intensive applications. We argue that given the resource-scarcity of IoT devices, a lightweight web component and programming model is necessary to process and manage RESTful web services.

4.4 User/Application Layer (UAL)

This layer represents user applications that interact with the TSL in order to make requests and consume services. For example, user request can be checking the temperature of a given location or adjusting lights in a room or building. Communication will go through the web services described in the TSL and they can be accessed from virtually any web component, device and implementation language.

5 Scenario

In a vibrant city environment we are constantly overloaded with information from various sources. As more and more objects are becoming smart and interactive there are constant feeds trying to communicate with us. Such communication is often achieved through connection with our mobile devices. The mobile device can through Bluetooth, RFID and NFC be an integrated part of the environment, consuming information from nearby posters, bulletin boards or integrated sensors. They can actively push and pull information facilitating for, i.e., commercial displayed information on devices tailored accordingly to context, location and interests. In such an environment with potential information overload, the information acquired needs to be precise and tailored. Various forms of distribution or filtering of this information networks are available, but we take this one step further to propose a proactive model for this by using our proposed architecture framework (Fig 2).

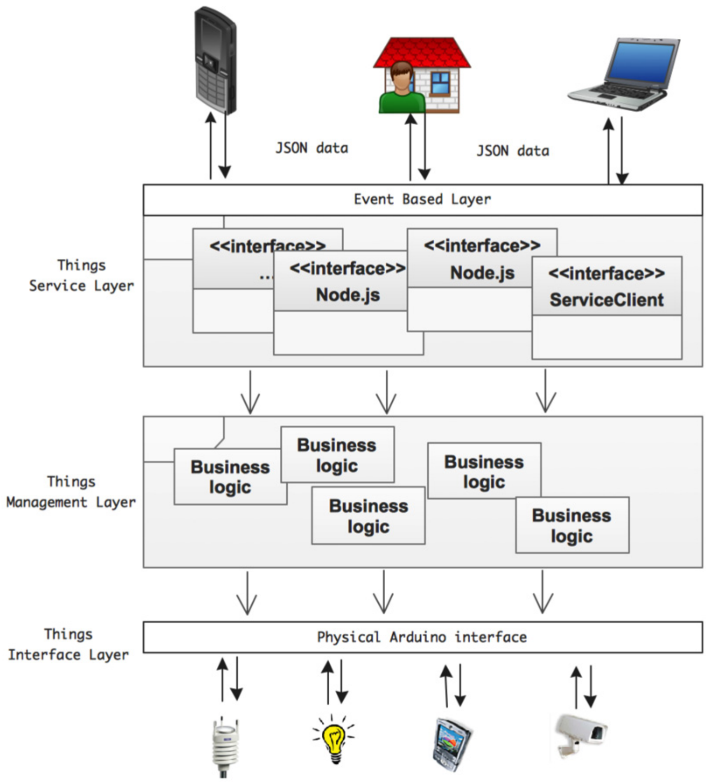


Fig. 2. Proposed Framework Architecture in Applied Scenario

By using this framework a user moving around in a city would end up with filtered, tailored and personally adapted information on his/her mobile device. General utilization could be: the customer needs to find the fastest route to a specific address; a recommendation for a particular shop to purchase the desired product from based on IoT information aligned with personal preferences; recommendation of shops to visit in order to fulfil the needs of the shopping list. Application usage and examples scenarios are numerous, but nevertheless it is important how this information is going to be made available. By applying a service, management and interface layer we provide three levels of abstractions allowing for service composition in each step. The physical Arduino devices will be wrapped in the interface layer, concealing all communication needed for physical IoT sensor access. Through the management layer business logic can be built and wrapped for access from the service layer. The service layer will then provide the physical IoT sensors available as services to end users/ applications. This will then facilitate taking advantage of a modern event based architecture communication from Node.js.

6 Framework Architecture Components

Given an environment consisting of a rapid growing number of smart objects that can sense, measure, record, transmit and provide services this brings to the table a whole new range of possibilities and challenges. Despite the WoT being excellent for making things smart, it still needs a wrapper to be able to offer aggregated data from its IoT sensors. Previous research has dealt with sensor communication and data aggregation as separate matters [6, 7, 20]. Earlier approaches have often used traditional client server programming model in a layered architecture fashion. Though nothing wrong with this, we see the possibility for a cleaner maintenance model, the architecture and easier options for dynamic growth based on our suggested Framework architecture. We build upon a popular technique to efficiently and flexibly manage concurrent interactions on the server in event-driven programming. This approach makes use of asynchronous procedure calls that are executed in the order they are created [20] and such event-driven programming provide performance and flexibility benefits over traditional synchronous approaches.

6.1 Node.js

The interest of JavaScript on the server has increased lately by the introduction of a new event-based framework named Node.js [21]. Node.js differs from other efforts by providing an event based framework intended for writing scalable Web servers by utilising JavaScript. It does not use the commonly used concurrency model based on threads, but employs an event model that makes input and output with network and files asynchronous and non-blocking [21]. This makes it suitable for new protocols and standards such as WebSocket and Server-Sent Events, which rely on having an open connection to the server. In order to facilitate for modern WoT application architecture, we have in our framework architecture demonstrated how this technology can be applied to gain an advantage.

6.2 Cloud as Backbone for WoT

Although there have been a considerable number of research efforts targeting IoT/WoT application areas, the same cannot be said in respect of the infrastructure capable of supporting this vision. We believe that the cloud is ideally placed in this respect. Whilst the literature mostly treats the issue from a high-level perspective [22], in this paper we take the effort one step further and show a detailed implementation map of WoT characteristics to cloud features. To this end, we use JSON, which enables the use of a shared format for expressing and transferring data between application endpoints, reducing the need for custom formats. In the next two sections, this will be further detailed when looking at application growth, interoperable communication and performance results.

6.3 Growth of Application Usage

To address the challenge of application usage we can rely on the cloud distribution model in terms of the possibility to scale resources up or down, according to traffic. Given that the administration of resources happens over web protocols, one would like to keep verbosity in message exchange to a minimum, at the same time as sending clear messages by using the JSON data format. The most effective way to transfer data in JSON is using arrays. A single message can thus be formatted in the following way:

[“MessageType”, ConnectionId, MsgID, “Message”, “TimeStamp”]

The use of a JSON array therefore facilitates for new WoT components to be easily included in communication, thus not slowing network growth and aiding scalability (in its absence, one would have to code specific parsers for each new data type used). This would provide support for infinite scaling of the solution when deployed to a cloud vendor. Results from a comparison between a Node server and a Jetty server, show how much more efficient Node is in handling concurrent requests (Fig 3).

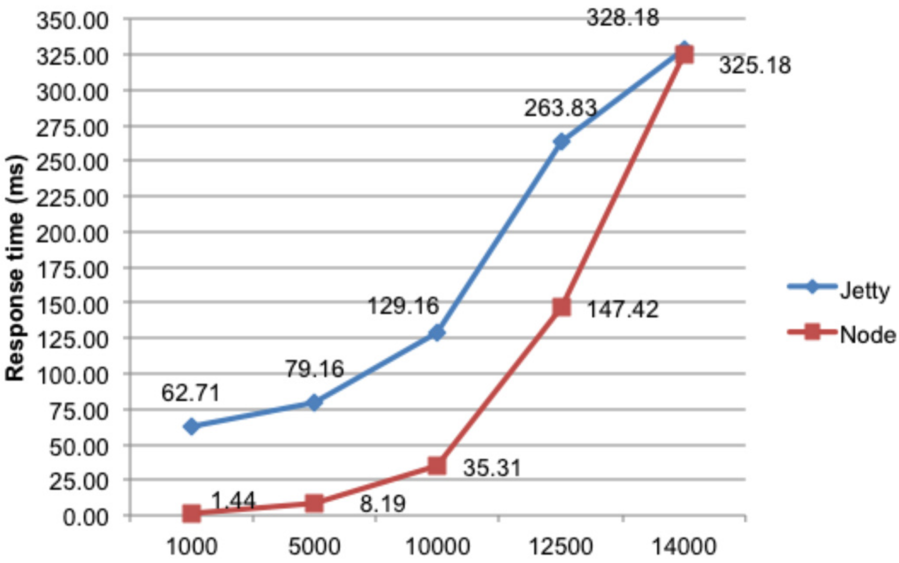


Fig. 3. Number of concurrent requests

From Fig 3 we can see that the response time for Node stays considerably below what the Jetty server uses until they both reach peak load at 14000 concurrent requests. Another perspective on growth of application usage will be the ability to add and remove IoT sensors as one pleases to the WoT network. In a vibrant city environment this will be a daily task as sensors come and go/turned on and off.

We propose a lightweight-programming model by using Node.js to communicate with different endpoints. This will be independent from proprietary sensor languages and will facilitate ad hoc composition of services in the cloud infrastructure. Fig 4 below shows a code snippet for using Node.js to read from a RFID sensor (details from the RFID setup library are excluded in this example).

```
function onRead(error, input2) {
  var dataChunk;
  var id;

  if (input2[0] != 0) {

    console.log(... pick data from input2 and format);

    dataChunk = input2[4];
    id = (data.splice(0)).splice(5, size);
    console.log(... log data extracted from id);

    }
  }
  [libraryFunction].read(onRead);
}
```

Fig. 4. Read from RFID sensor

6.4 Interoperable Communication

WoT application architectures can be improved by deriving a set of guidelines on how to partition the application between the server and client. There are significant drawbacks to using different languages on the client and server in modern Web based applications and that it prevents a uniform development experience. Also the use of a single programming language will bring benefits such as better flexibility, a common code base and more control of partitioning the applications. Fig 5 displays the use of Node.js for interpreting an IoT client sensor message call given in JSON.

```
var http = require('http');
var options = {
        host: 'localhost',
        port: 3000,
        path: '/load/debug',
        method: 'POST'
    };
var req = http.request(options, function(res) {
    console.log("Got response: " + res.statusCode);
    console.log('HEADERS: ' + JSON.stringify(res.headers));
    res.on('data', function(chunk) {
        console.log("Body: \n" + chunk);
    });
});
req.end();
```

Fig. 5. Interpreting a POST request

Applying Node.js for this task increases the utilization of common resources and a shared background. Using common standards enables interoperable communication both among WoT smart objects, but also among WoT solutions if desired. The lightweight footprint of Node.js enables focus on implementation logic and component interoperability.

7 Conclusion

The field of Web development has been characterized by a lack of coherent architectural concepts, partly due to the absences of standard protocols that support modern communication styles. As a result of this, systems specifically designed for real-time data delivery have been realized with proprietary technology in the industry, often violating established software engineering principles. The danger is that the vision of the Web of Things will also suffer from the same drawbacks. In this article, we address this issue and propose a new, lightweight architecture for the WoT.

Taking the case of a smart city, we implement a proof-of-concept application based on the proposed architecture, which uses modern technologies such as Node.js and JSON and show how IoT devices are accessed and managed through WoT services managed by a Cloud solution. We see this as an incremental step towards further eliminating the borders between smart objects and the data they generate on mobile clients. Future research should further explore the possibilities for strengthening and standardizing such a service environment, as well as addressing important issues such as privacy, security and real-time response needs.

Acknowledgement. We are grateful to Johan Andre Lundar's comments and work with Node.js.

References

1. Kortuem, G., et al.: Educating the Internet-of-Things Generation. Computer 46(2), 53–61 (2013)
2. Konomi, S., Roussos, G.: Ubiquitous Computing in the Real World: Lessons Learnt from Large-Scale RFID Deployments. Personal and Ubiquitous Computing 11(7), 507–521 (2007)
3. Bui, N., Zorzi, M.: Health care applications: a solution based on the internet of things. In: Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2011), Article 131, 5 pages. ACM, New York (2011)
4. Ghose, A., Bhaumik, C., Das, D., Agrawal, A.K.: Mobile healthcare infrastructure for home and small clinic. In: Proceedings of the 2nd ACM International Workshop on Pervasive Wireless Healthcare (MobileHealth 2012), pp. 15–20. ACM, New York (2012)
5. Efstratiou, C., Davies, N., Kortuem, G., Finney, J., Hooper, R., Lowton, M.: Experiences of designing and deploying intelligent sensor nodes to monitor hand-arm vibrations in the field. In: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys 2007), pp. 127–138. ACM, New York (2007)

6. Doody, P., Shields, A.: Mining network relationships in the internet of things. In: Proceedings of the 2012 International Workshop on Self-aware Internet of Things (Self-IoT 2012), pp. 7–12. ACM, New York (2012)
7. Ma, Y., Rao, J., Hu, W., Meng, X., Han, X., Zhang, Y., Chai, Y., Liu, C.: An efficient index for massive IOT data in cloud environment. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 2012), pp. 2129–2133. ACM, New York (2012)
8. Li, Y., Teraoka, F.: Privacy protection for low-cost RFID tags in IoT systems. In: Proceedings of the 7th International Conference on Future Internet Technologies (CFI 2012), pp. 60–65. ACM, New York (2012)
9. Kozlov, D., Veijalainen, J., Ali, Y.: Security and privacy threats in IoT architectures. In: Proceedings of the 7th International Conference on Body Area Networks (BodyNets 2012), pp. 256–262. ICST, Brussels (2012)
10. Jia, H., Wu, M., Jung, E., Shapiro, A., Shyam Sundar, S.: Balancing human agency and object agency: an end-user interview study of the internet of things. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp 2012), pp. 1185–1188. ACM, New York (2012)
11. Richardson, L., Ruby, S.: RESTful Web Services. O'Reilly Media, Inc. (2007)
12. Duquennoy, S., Grimaud, G., Vandewalle, J.: The Web of Things: interconnecting devices with high usability and performance. In: Proceedings of the 6th IEEE International Conference on Embedded Software and Systems (ICESS 2009), HangZhou, Zhejiang, China (2009)
13. Guinard, D., Ion, I., Mayer, S.: In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective. In: Puiatti, A., Gu, T. (eds.) MobiQuitous 2011. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 104, pp. 326–337. Springer, Heidelberg (2012)
14. Castro, M., Jara, A.J., Skarmeta, A.: Architecture for Improving Terrestrial Logistics Based on the Web of Things. Sensors 12(5), 6538–6575 (2012)
15. Guinard, D., Trifa, V.: Towards the Web of Things: Web Mashups for Embedded Devices. In: Proc. WWW 2009, Madrid, Spain (2009)
16. Khajeh-Hosseini, A., Greewnood, D., Smith, J.W., Sommerville, I.: The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise. Software: Practice and Experience 42(4), 447–465 (2012)
17. Kovatsch, M., Mayer, S., Ostermaier, B.: Moving Application Logic from the Firmware to the Cloud:Towards the Thin Server Architecture for the Internet of Things. In: Proc. of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012), Palermo, Italy (2012)
18. Miorandi, D., Sicari, S., De Pellegrini, F., Chlamtac, I.: Internet of Things: Vision, Applications & Research Challenges. Ad Hoc Networks 10(7), 1497–1516 (2012)
19. Hurlburt, G.F., Voas, J., Miller, K.W.: The Internet of Things: A Reality Check. IT Professional 14(3), 56–59 (2012)
20. Ganty, P., Majumdar, R., Rybalchenko, A.: Verifying liveness for asynchronous programs. In: ACM SIGPLAN Notices. ACM (2009)
21. Joyent Inc. Node.js - Evented I/O for JavaScript (2013), `http://nodejs.org/`
22. Alhamad, M., Dillon, T., Chang, E.: Conceptual SLA framework for cloud computing. In: 2010 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST), April 13-16, pp. 606–610 (2010)

Optimal Personal Area Network Coordinator Placement in Grid Topology

Muneer Bani Yasin, Maryan Yatim, Marwa Salaymeh,
Yaser Khamayseh, and Wail Mardini

Department on Computer Science, Jordan University of Science and Technology
Irbid, Jordan
{masadeh,mksalayma,yaser,mardini}@just.edu.jo,
mn.yatim@gmail.com

Abstract. Wireless sensor network (WSN) is a set of sensing devices that communicate with each other via a wireless medium. These sensing devices capture the sensed data, process it and send it to the main station which is called the Base Station (BS). Most of WSN applications are designed to transmit with low data rate, low energy consumption and at low costs. Zigbee is a new wireless protocol governed by the IEEE 802.15.4 standard; it is a technology of data transfer in wireless sensor networks. It is characterized with low energy consumption and it is designed for multi tunnel control systems. In this paper, a grid network topology is implemented according to 802.15.4/ZigBee standard using QualNet Simulator 5.2. By tryingdifferent Beacon Order (BO) and Superframe Order (SO) combinations, this paper aims to find the optimal Personal Area Network Coordinator (PANc) placement in grid topology based on energy consumption, end to end delay, and throughput. However, this paper proved that when PANc is placed at any angle of grid topology, then the maximum throughput is achieved, while when the PANc is placed at the center of grid topology, then the minimum energy consumption and end to end delay are achieved.

1 Introduction

Wireless Sensor Networking is an innovative area within the bold spectrum of wireless networks that are specifically designed to capturesensed data. These little pieces of information are extremely valuable to companies because they provide insights into events that might enhance their business processes. Sensor devices or nodes provide an advanced WSN platform. First of all, WSN combines what we call "Check, Track, and Trace" so it integrates different functional capabilities; Check means sensing the surrounding environments. Track provides the capability to locate objects the surrounding environment, and finally Trace is the ability of sensors to lock the history of their sensor measurements.

1.1 Wireless Sensor Networks

The rapid growth of wireless communication made the transition from sensor networks to wireless sensor networks (WSN) easier. Wireless sensor network is a set

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 260–273, 2013.

of sensing devices that communicate through a wireless links [1]. These sensing devices capture the sensed data, process it and send it to the main station, namely Base Station (BS). These sensing nodes or devices have two main components for controlling and sending information; first component is the microcontroller in which all communications and processes of the device are controlled. The second component is the transceiver which controls the sending and receiving of signals, and decides when exactly it's time to snooze, sleep or wake up. Since energy conservation and utilization is the main goal of WSN researches, transceiver is designed to switch to the appropriate mode: idle, sleep, and active. When the device is in the sleep mode then it is not wasting its power and energy, and thus prolonging the battery life-time.

1.2 IEEE 802.15.4 Standard

In the past decades, many protocols and applications were implemented and developed, all in which were competing to achieve the ideal solution of energy consumption, simplicity, scalability, medium utilization, low cost, and minimum delay. Unfortunately all proposed protocols were developed to ameliorate one special case of the enormous number of required applications, in other words, none of the proposed protocols is designed to achieve all applications requirements. Hence, there is a real need for a general protocol to achieve all the needed requirements, hence, the IEEE 802.15.4 standard was proposed.

Referred to its name, IEEE 802.15.4 standard is a Wireless Personal Area Network standard Proposed by the Institute of Electrical and Electronic Engineers. The IEEE 802.15.4 standard deals with the wireless Personal Area Networks (WPAN). Number 15 in IEEE 802.15.4 represents the 15th working group of WPAN, where number 4 refers to the task group. That means, task group 1 is Bluetooth, task group 2 is for coexistence with 802.11, while task group 3 is for high rate WPAN, and finally task group 4 is for low rate and high battery life, and that where ZigBee comes in.

1.3 ZigBee

ZigBee protocols are based on IEEE 802.15.4 and its task group specifications were approved in April 2003. Zigbee is a national working technology guided by the IEEE 802.15.4 standard. It is a simple point to point wireless technology for data transfer in in low energy consumption, and it also can be served for the multi tunnel control systems. Zigbee standard defines a multilayer structure of software, usually this is called Zigbee protocol stack. The IEEE 802.15.4 standard is specialized in the lower two layers, Physical Layer (PHY) and Data Link Layer (DLL), while ZigBee standard is specialized in the upper two layers. Therefore, and to achieve a comprehensive architecture, the name ZigBee has been connected to the IEEE 802.15.4 standard in one worldwide known expression which is the IEEE 802.15.4/ZigBee.

Zigbee has a unified standard for data transfer; all the devices of a protocol can interact. Zigbee ensures that networks remain operable in the conditions of a concen detained in quality between communicating nodes. A ZigBee device can operate with other ZigBee components to generate a large Mesh topology called Wireless Personal

Area Network (WPAN). A node or device in ZigBee networks can be a Full-function device (FFD) or Reduced-function device (RFD). The IEEE 802.15.4 distinguishes between Full Functional Devices (FFD) and Reduced Functional Devices (RFD). FFDs can be "coordinators" and can be the hop of the network, where all what can RFDs do is to connect to a FFD, also FFDs can implement the higher layers tasks.

ZigBee has 3 main components or devices; ZigBee coordinators (ZC) which are FFDs and supportsthe most capabilities, the most functionalities and the root of the network tree and can be used to bridge other ZigBee routers to other ZigBee networks. Each ZigBee network has one and only one ZigBee coordinator. ZigBee coordinators can store information about the network and topology map,it can function as a repository for 128 bit security keys. The second component is ZigBee router (ZR), which accesses an intermediate router that passes data from other in-devices to other in-devices. The routers function primarily with ZigBee coordinator. And finally ZigBee End Device (ZED) which has enough power and enough capability to communicate with its parent node which is ZigBee router. ZC can itself be a ZED.

ZigBee Topologies. A topology is a physical or logical structure of the network. This section defines 4 types of ZigBee topologies: Star topology, Peer-to-peer topology, and Mesh topology which can be formed as Grid topology.

Star Topology. In star topology, every node in the network communicates only with a coordinator device called Personal Area Network coordinator (PANc), and vice versa. There are no two nodes that can communicate with each other's directly. In star network, all traffic funnels to a central point, such as access point. Routing is simple in this topology with low overhead. Devices are limited in their ability to communicate with each other directly.

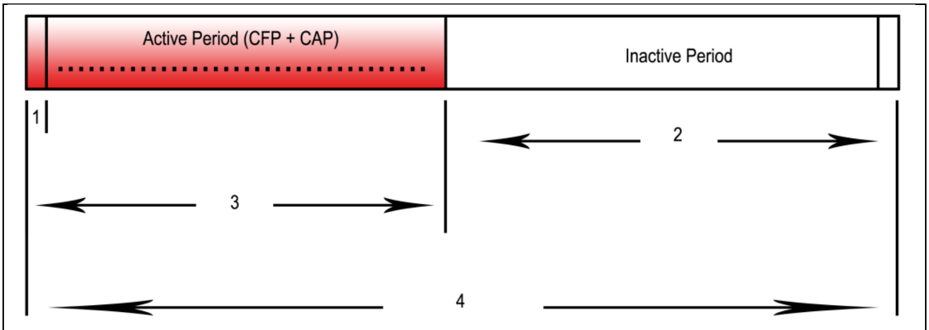
Peer-to-Peer Topology. In this type of topology, nodes are equivalent in responsibilities, and processing. Whereas, in client server networking certain nodes have certain responsibilities for serving and other nodes have the responsibility for receiving. In a peer to peer network, all nodes are the same; each node is actually acting like a client and a server. Peer to peer networking is also a lot more common in small networks.

Mesh Topology. Mesh network provides dynamicand reliable routing. One of the defining properties of the mesh network is that it's self forming; that means that the network nodes automatically establish routes to new nodes when they become available. Another property of mesh network is that it's self healing; that means if a device is taken out of commission, the devices that were routed through that node are able to reconnect to the network and are able to rejoin the network through existing nodes. The data of mesh network "hops" along the network, that mean the nodes pass the data to the next node or to its neighbor and its neighbor pass it to its neighbor and so on, until it reach its destination. In mesh topology, data has many alternative paths to reach its destination if something wrong occurred to the main path. But in contrast there are a lot of connections between nodes, which may cause a bandwidth limitation and also high cost.

Grid Topology. Grid topology is a special case of mesh topology where nodes are arranged regularly. In Grid topology node can be arranged regularly in one dimensional or two dimensional shapes. Regularity in Grid topology means that nodes are not placed in randomly; instead they have a regular shape to be placed in, like ring shape, and rectangular shape. Nodes in this type of topology can be either RFDs or FFDs depends on the scenario being implemented on the topology.

1.4 Superframe Parameters

In any wireless sensor network, nodes should first communicate with their coordinator in order to participate in the network. Many nodes start performing their tasks simultaneously on the medium or channel, so the coordinator should synchronize the duties of these nodes. Beacon frame is a new feature of IEEE 802.15.4 standard in which PAN coordinator identifies itself to the existing nodes so they can communicate and get their duties synchronized. As nodes are categorized into FFDs and RFDs, only FFD nodes can send beacon frames. As shown in Figure 1, beacon interval (BI) is divided into two parts; active period and inactive period. The active period in which Superframe Duration (SD) exists consists of 16 slots which can be also partitioned into Contention Free Period (CFP) and Contention Access Period (CAP). CFP refers to the period that has Guaranteed Time Slots (GTS) which are time slots preserved only for specific nodes to transmit on, so all the packets transmitted on these slots are guaranteed to be received, while CAP refers to the period in which nodes try to access the medium simultaneously without collision, so they need to compete in order to take the chance to send when the medium is idle and not busy. The second part of BI is the inactive period which refers to the period in which nodes and their coordinator sleep.



1: Beacon (First Slot) 2: Inactive Period 3: Active Period (CFP+CAP) 4: Beacon Interval

Fig. 1. Beacon Interval

The beacon frame that is sent out periodically from PAN coordinators is responsible for defining the length of each period length in the BI; the Beacon Order (BO) determines the BI while (SD) is expressed in the meaning of Superframe Order (SO). The duty cycle of each node is the time of which the nodes are active and not in the sleep mode, and it is determined by the values of (BO, SO).

The rest of this paper is organized as follows. Section 2 presents some related work. Section 3 and Section 4 illustrates the simulations scenarios, simulation results and results analysis. We conclude the paper in Section 5 and present the future work in section 6.

2 Related Work

- M. Hussnain et. al. [2] focused on evaluating different wireless sensor networks applications in agriculture and comparing 3 types of ZigBee topologies which are tree topology, mesh topology, and grid topology for precision agriculture. The comparison of the 3 topologies is based on 3 main factors which are throughput, end to end delay and the load of the network. The simulation of each topology is implemented on 1000 meters area for all topologies, while number of nodes for each topology is different; grid topology is implemented with one ZigBee coordinator and six end devices, while tree and mesh topologies are build with one ZigBee coordinator, three ZigBee routers, and six end devices. The reason behind that is that grid topology does not need ZigBee routers. The simulation results and statistics of this paper proved that a better throughput and network load can be achieved in WSNs when tree topology is used. The throughput of the network achieved its maximum in tree topology and its minimum in grid topology. Moreover, the end to end delay of tree topology is higher than the other two topologies by negligible percentage value. The reason behind the efficiency of tree topology in WSNs is its flexible structure; any new sensor can be added to the topology and also any damaged sensor can be removed without affecting or altering other nodes or network. Hence, this paper proved that tree topology in WSNs is the best to be deployed in precision agriculture.
- Fan Ouyanget. al. [3] evaluated and analyzed the packet delivery ratio and throughput for IEEE 802.15.4 Standard. They simulated the performance of these two parameters using NS2 simulator by trying number of (BO, SO) combinations with their values no more than 6. This paper considered duty cycle percentages of 50% and 100%. Results proved as predicted that by increasing the value of BO the throughput of the nodes increases. The reason behind that is when BO value increases then the packet drops absolutely decreases. Furthermore, packet collisions increase as number of nodes increase. However, this research enhanced IEEE 802.15.4 standard by allocating the Guaranteed Time Slots (GTS) dynamically.

3 Simulation

The performance of the topologies under study is evaluated using QualNet 5.2 simulation package. The main objective is to find the optimal PANc placement in grid topology in terms of throughput, end-to-end delay, and energy consumption under different BO, and SO combinations ((6,6), (7,6), (8,6), (8,7), (8,8)) which are considered the optimal (BO, SO) combinations for the needed parameters in WSNs [4]. Three scenarios are considered; first, PANc is placed at the center of the grid topology, then the topology was modified so the PANc is out of grid topology, and finally PANc is placed at one angle of the grid topology. Each one of the three scenarios is to be simulated 5 times to increase the accuracy of our work. The analysis of the three scenarios is calculated to get the statistics and evaluate results based on throughput, end to end delay, and energy consumption in its four modes (Energy Consumption in Transmit Mode, Energy Consumption in Receive Mode, Energy Consumption in Idle Mode, and Energy Consumption in Sleep Mode). Simulation parameters used in the three scenarios are shown in Table 1.

Table 1. Simulation parameters

Parameter	Value
Simulation Time	1000 seconds
Terrain Size	50 x 50 meters
Packet Size	50 bytes
Distance Between Nodes	5 meters
Packet Arrival Rate	1 second
Start Time	Starts from 10 seconds to 18 seconds
Application	Constant Bit Rate (CBR)
Energy Model	MicaZ
Antenna Height	0.08
Routing Protocol	AODV
Seed	1
Number of Experiments	5

3.1 First Scenario

In the first scenario, the PANc is located at the center of the grid topology as shown in Figure 2. The number of nodes and their types are identical in the three scenarios. There are eight RFD nodes and one node considered as PANc. This scenario is tested on five (BO, SO) combinations and the analysis and results derived out of this scenario are shown in Table 2.

3.2 Second Scenario

In the second scenario the PANc is located out of grid topology 5 meters far away from the nearest node as shown in Figure 3. Again, this scenario is tested on five

(BO, SO) combinations and the analysis and results derived out of this scenario are shown in Table 3.

3.3 Third Scenario

Finally, the PANc is located at one of the grid topology angles in scenario three as shown in Figure 4. Results derived out of this scenario are shown in Table 4.

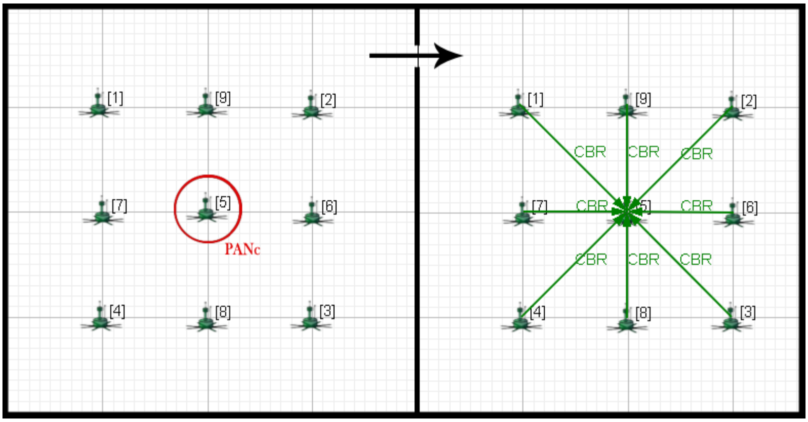


Fig. 2. Grid topology/ PANc at center

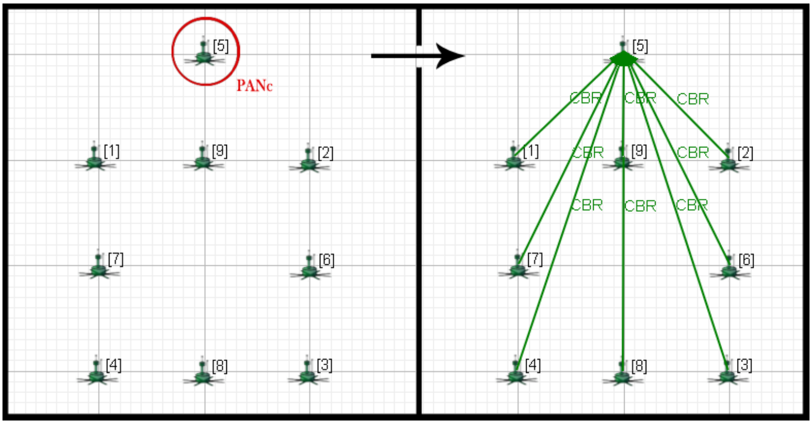


Fig. 3. Grid topology/ PANc out of topology

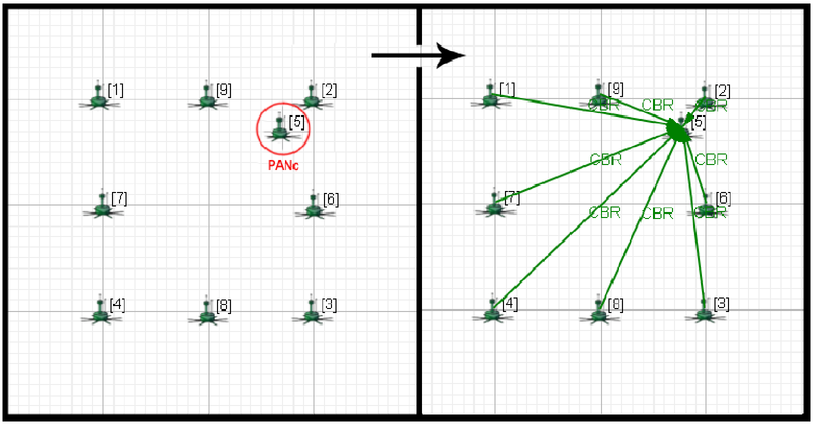


Fig. 4. Grid topology/ PANc at one angle of grid topology

Table 2. Results of PANc at center of grid topology

(BO, SO)	Average Throughput PAN/Center	Average Delay (sec) PAN/Center	Average Energy PAN/Center
(6,6)	2123.4	5.784028	2.6155
(7,6)	914.8	2.678864	2.6709
(8,6)	752.2	6.36383	3.3844
(8,7)	928.25	3.849015	3.3077
(8,8)	1955.4	3.76766	1.6931

Table 3. Resultsof PANc out of grid topology

(BO, SO)	Average Throughput PAN/Out	Average Delay (sec) PAN/Out	Average Energy PAN/Out
(6,6)	2118.8	5.422956	2.942277733
(7,6)	1161.4	3.537956	3.0655562
(8,6)	783.6	6.801628	3.607247267
(8,7)	1092.8	3.879382	3.496440267
(8,8)	1928.2	4.072978	1.881936467

Table 4. Results of PANc at one angle of grid topology

(BO, SO)	Average Throughput PAN/Angle	Average Delay (sec) PAN/Angle	Average Energy PAN/Angle
(6,6)	2133.6	6.099626	2.9470794
(7,6)	1148.4	4.1233	3.045094333
(8,6)	852.2	6.941414	3.514824
(8,7)	1113.2	4.967636	3.5100696
(8,8)	1954.2	4.070374	1.883083467

4 Simulation Results

As mentioned before, the simulation analysis andresults were derived in terms of three parameters; throughput, end to end delay, and energy consumption in its four cases (Energy Consumption in Transmit Mode, Energy Consumption in Receive Mode, Energy Consumption in Idle Mode, and Energy Consumption in Sleep Mode).

4.1 Throughput

Throughput is defined as the amount of data packets sent from sender to receiver in a specific amount of time which is seconds. Because a huge number of nodes or devices always try to access the wireless channel at the same time which leads to collisions, throughput is considered very important. Throughput can be analyzed by several factors like data packet collision, topology structure, and impediments that may occur between devices or nodes.In the three scenarios illustrated before, throughput was measured by calculating the average value for the batch of 5 experiments of each scenario and for each (BO, SO) combination as shown in Figure 5.

It's obvious in Figure5 that when (BO, SO) values are (6, 6) and (8, 8), the values of throughput for the three scenarios are approximately the same and have high values, while there is a clear difference in throughput when the values of (BO, SO) = (7, 6), (8, 6), (8, 7) in which PANc located at the center of grid topology achieved the minimum throughput, and the scenarios in which PANc was located at one angle of topology and Out of the topology achieved the maximum throughput. The Reason behind that is due to the amount of packet drops in each scenario and the overall load divided by the nodes; in scenario one where PANc is located at the center of the grid

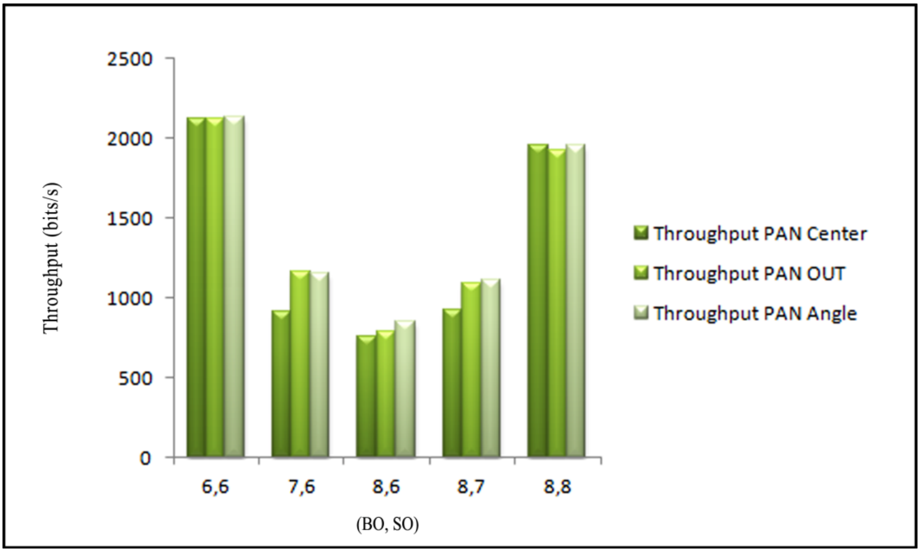


Fig. 5. Throughput results

topology, all the nodes are sending almost at the same signal power because the distance between nodes and PANc is the same, hence, the resulting throughput of each node is expected to be in the same range as others, while when PANc is located at one of grid topology angles, nodes which are located close to the PANc will cause a very high signal powers due to the small distance between them and the PANc, and thus decrease the probability of interference and packet loss in average, which at the end provides a high throughput and that's why scenario three achieved the maximum throughput.

Moreover, the values of (BO, SO) combinations affected the value of throughput for each scenario; throughput value reached its maximum when (BO, SO) values were (6, 6) and (8, 8) in all scenarios because in these two intervals the duty cycle of the nodes is 100% which means that all nodes are active during all the time and are sending packets all over the simulation time, while when (BO, SO) values = (7, 6), (8, 6), and (8, 7) the nodes are active 25-50% of the time, which means there are periods where nodes are sleep and the whole model is paused due to not sending nor receiving data packets. Furthermore, when the sleep period ends, all the nodes wake up and send packets at the same time, the case in which more probability for packet drops and collisions occur, thence, using these (BO, SO) values achieved less throughput in all the three scenarios.

4.2 End-to-End Delay

End to end delay measures the total time in which the packet is queued to be sent from the sender to the time the final bit of this packet reaches its destination. Figure 6 shows the results of end to end delay for the three scenarios in all (BO, SO) combinations. Scenario three in which PANc is located at one angle of grid topology

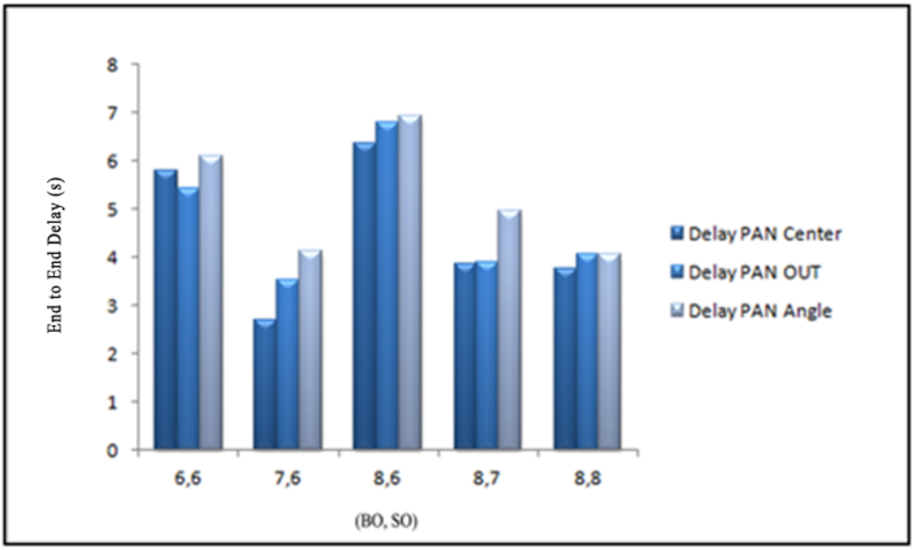


Fig. 6. End-to-End delay results

achieved the maximum end to end delay in all the values of (BO, SO) combinations. While scenario one when PANc is located at the center of grid topology achieved the minimum end to end delay except when the (BO, SO) value is (6, 6) as shown in Figure 6.

4.3 Energy Consumption

Energy consumption of a node in any network depends on four modes, transmit mode, receive mode, idle mode, and sleep mode. When nodes are in a sleep period the energy consumption in idle mode increases because there are no packets have been sent nor received in the running time, however, nodes wake up and are ready to send data they all wake up and send data at the same time which causes transmit mode energy consumption to increase, hence when duty cycle for nodes is 100%, which means that nodes are sending packets all over the simulation time, the overall average energy consumption is less than that of when node's duty cycle is 50% or 25%.

Figure 7 shows that when PANc is placed at the center of grid topology less energy consumption is achieved for all (BO, SO) combinations, and the minimum energy consumption is achieved when (BO, SO) is (8, 8). The reason behind that is the duty cycle; when SuperFrame parameters is equal then the node's duty cycle is 100% and is active all over the simulation time. Also, energy consumption in idle mode when (BO, SO) is (8, 8) achieved the minimum values between all other SuperFrame parameters combinations because transceiver is active all over the simulation time and is receiving packets all over that period, which means that there is no pause periods or sleep periods for the nodes in this interval. However, the reason that caused (8, 8)

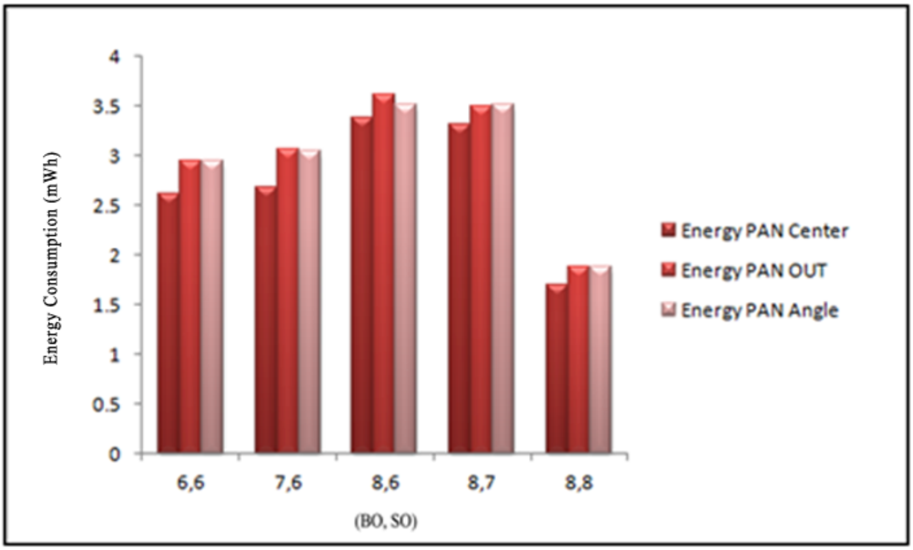


Fig. 7. Energy Consumption Results

combination to overcomes the (6, 6) combination while (6, 6) combination also means 100% duty cycle, is because when the (8, 8) combination is used then the packet drop possibility decreases because of the CAP in beacon frame, and hence the need for retransmitting a packet decreases which decreases the energy consumption.

5 Confidence Interval

Table 5. Results of confidence intervals for the three parameters

	Energy Consumption	End to End Delay	Throughput
PAN Angle	[2.396047, 3.564013]	[4.138438, 6.342502]	[943.9381,1936.702]
PAN center	[2.137691, 3.331073]	[3.145097, 5.832262]	[765.384, 1904.236]
PAN out	[2.998692, 3.598468]	[3.555042, 5.930918]	[830.2418, 1839.378]

6 Conclusion

This paper studies the optimal location for placing PAN coordinator in grid topology in terms of throughput, end-to-end delay, and energy consumption fordifferent (BO, SO) combinations, which are,((6,6), (7,6), (8,6), (8,7), (8,8)). Those combinations revealed to be the optimal (BO, SO) combinations in [4]. We simulated three scenarios of PANc placements using QualNet 5.2 Simulator, first the PANc is placed at the center of 8 nodes in grid topology, and then we placed the PANc at one angle of the grid topology, finally PANc is placed 5 meters out of grid topology. The simulation of each scenario is repeated 5 times to get more accurate results.

As the most important parameters in WSNs are end to end delay and energy consumption, according to the revealed simulation results, the best PAN coordinator placement is at the center of grid topology because this scenario achieved the minimum end to end delay and the minimum energy consumption at almost all the (BO, SO) combinations. However, the minimum end to end delay when PANc is placed at the center of grid topology is achieved at (BO, SO) equals (7, 6). On the other hand, the minimum energy consumption for this scenario is achieved when (BO, SO) equals (8, 8). Moreover, the simulation results show that when PANc is placed at one angle of grid topology, maximum throughput is achieved specially when (BO, SO) equals (6, 6) and (8, 8). Therefore, the placement of PANc in grid topology depends on the desired parameters of the network; if in some situation the throughput of a network is the most important feature, it is better to place the PANc at one angle of grid topology to get the best results. While, if the end to end delay and

Table 6. Final results of best PANc placement

Parameters	Best Place For PANc	Best (BO, SO) Combination
Maximum Throughput	PANc at one Angle of Grid Topology	(6, 6) (8, 8)
Minimum End-to-End Delay	PANc at Center of Grid Topology	(7, 6)
Minimum Energy Consumption	PANc at Center of Grid Topology	(8, 8)

energy consumption what matter, then the placement of PANc at the center of grid topology will certainly achieve the best results. Table 6 summarizes the results of this paper.

7 Future Work

As this research is based on one arrival rate which is 1 second, our future work aims to expand the study of this research on different arrival rate times, and to develop an adaptive algorithm that can dynamically provide the best choice of PAN coordinator placement based on the simulation results and the desired parameters of a given network.

References

1. Koubaa, A., Alves, M., Tovar, E.: A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks. In: IEEE WFCS (2006)
2. Hussnain, M., Sharjeel, M., Chaudhry, S.R., Hussain, S.A., Raza, I., Mirza, J.S.: Investigating Multi-Topological ZigBee Based Wireless Sensor Network in Precision Agriculture. Journal of Basics and Applied Scientific Research (print, 2013)
3. Fan, O., Xu, D., Zhang, X., Chen, C.: Modeling and analysis of energy consumption for ZigBee-based wireless sensor networks. In: 2012 31st Chinese Control Conference (CCC). IEEE (2012)
4. Salayma, M., Mardini, W., Khamayseh, Y., Yasin, M.: Optimal Beacon and Superframe Orders in WSNs. In: Proc. of the Fifth International Conference on Future Computational Technologies and Applications, FUTURE COMPUTING 2013, Valencia, Spain, May 27–June 1. IARIA (2013)
5. Bougard, B., et al.: Energy efficiency of IEEE 802.15.4 standard idense wireless microsensor networks: modeling and improvement perspectives. In: Proceedings of Design, Automation and Test in Europe, DATE 2005 (2005)
6. Chandane, M.M., Bhirud, S.G., Bonde, S.V.: Performance Analysis of IEEE 802.15.4. International Journal of Computer Applications (0975 – 8887) (2012)

7. Liu, X., Haenggi, M.: The Impact of the Topology on the Throughput of Interference-limited Sensor Networks with Rayleigh Fading. Notre Dame, IN 46556. IEEE CONFERENCE PUBLICATIONS, USA (print 2008)
8. Ortiz, A.M., Royo, F., Olivares, T., Orozco–Barbosa, L.: Intelligent Route Discovery for ZigBee Mesh Networks. IEEE CONFERENCE PUBLICATIONS, University of Castilla–La Mancha–Albacete (print, 2011)
9. Ergen, S.C.: ZigBee/IEEE 802.15.4 Summary (September 2004), http://pages.cs.wisc.edu/~suman/courses/838/papers/zigbee.pdf
10. Khan, S.A., Khan, F.A.: Performance Analysis of a ZigBee beacon Enabled Cluster Tree Network. In: Third International Conference on Electrical Engineering (2009)

Mobile and Federated Access to DSpace-Based Digital Libraries

Isaias Barreto da Rosa and David Ribeiro Lamas

Institute of Informatics, Tallinn University, Estonia
{ibr,drl}@tlu.ee

Abstract. In developing countries, people struggle to access print-based publication and also face difficulties in accessing ICT resources such as computer and Internet. Considering the much higher availability of mobile phones than computer in these regions, accessing library materials on mobile devices can be an alternative. On the other hand, in developed countries, the proliferation of smartphones and tablets also demands a proper access to digital libraries on mobile devices. This paper features a mobile application for accessing DSpace-based digital libraries, a popular open source digital library system. The development of this application followed a user centric design approach and aimed at bringing a solution characterized by relevance and novelty.

Keywords: Mobile digital libraries, DSpace, digital divide, developing countries.

1 Introduction

The distribution of scientific and academic publications started experiencing a profound revolution with the birth and development of digital libraries. Vast panoply of resources is now accessible theoretically at anytime and from anywhere, as far as, for instance, a computer with Internet access is available. Therefore, if in the past researchers would often have to travel to different cities and sometimes even to different countries in order to consult scholarly publications, today they can access a great panoply of resources from the comfort of their offices, their homes and theoretically from wherever they are. Moreover, e-books have become a dominant force in contemporary publishing with sales rising nearly 40% in 2010. In 2011, over 30% of book publishers issued more than three quarters of their titles as e-books. In April 2011, Amazon started selling more Kindle e-books than printed books. One of the main reasons for such growth is massive spread of mobile computing and e-reading devices, like dedicated e-book readers, smart phones and tablet computers.

In the developing countries, due to the tragic failure of the traditional distribution mechanisms, digital libraries are seen as the great hope [15]. However, despite the great benefits of DL for these countries, they also face several problems in building and using them. With the several basic problems they face such as food, drinking water, sanitation, etc., spending the existing scarce resources

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 274–287, 2013.

on building these libraries is not a priority [15]. Apart from that, these countries face several other hindrances in this process, such as lack of financial resources, low digital and information literacy rates, poor ICT infrastructure, low computer and Internet penetration rates, high costs of building digital libraries, etc [12].

The International Telecommunication Union (ITU) estimates that in these countries, while mobile phone penetration rate is currently 77,8%, there are 24,8% households with a computer and 24 people using Internet for each 100 inhabitants[1]. Thus, if digital libraries are made available on mobile devices, they could theoretically reach a broader range of people. Therefore, making digital libraries accessible on mobile phones and understanding the way they are perceived and accepted in this context, seem to be particularly important.

Such alternative mechanism of accessing library resources is also relevant for the developed world that does not face the problems that are commonly found in the developing countries. In these more wealthy countries, according to the aforementioned ITU statistics, households with a computer and Internet penetration rates are 74,2% and 70,2% respectively, and there are about 128 mobile phones for each 100 people. On the other hand, there is a strong trend towards the usage of mobile devices for Internet access. Thus, making digital libraries accessible over mobile phones seems also to bring important added value into the developed countries.

This paper presents a design of an alternative mechanism for accessing DSpace-based digital libraries, through mobile devices. DSpace is a free and open source DL system used around the world. Making it properly accessible on mobile devices might ameliorate the accessibility of library resources to many people, especially when we take into account that resources in these libraries are usually freely accessible.

2 DSpace Digital Library System

When choosing a DL system, a questions that is often raised is: should we use an open source or a proprietary DL system? However, as mentioned above, in developing countries the usage of free open source digital library systems is indeed a must [15]. There are several open source DL systems available: Archimede, DSpace, EPrints, Fedora, Greenstone, Invenio, Keystone, MyCoRe, OPUS, Streetprint, etc. Among them, DSpace is one of the most popular, with more than 1430 installations around the world. It is OAI-PMH complaint and its interface is translated into more than 20 languages. DSpace can be installed on Windows, Linux, or Mac OSX. Apart from the Greenstone version for iPod-Linux and Android powered mobile phones, and the DSpace theme for mobile devices which was released in 2012, the majority of these existing open source DL systems does not include any specific interface for mobile devices.

DSpace supports two possible user interfaces: the first one is based on JavaServer Pages (JSP) technologies and the second one, often called Manakin,

[1] `http://www.itu.int/ITU-D/ict/statistics/at_glance/KeyTelecom.html`, April 10 2013.

is based on the Apache Cocoon framework (XMLUI). Modifying the JSP interface is difficult and expensive. However, Manakin (XMLUI) uses a modular interface layer to enable institutions an easy customization of the DL interfaces according to their own needs and requirements. The DSpace theme for mobile devices released in 2012, took advantage of this Manakin's modular interface.

What is presented in this paper however, is not another DSpace theme for mobile devices, but a mobile application which enables search over one or several existing DSpace-based digital libraries at the same time and from a single point. The core idea is bringing the possibility of performing offline reading of DSpace resources on mobile devices (which is only possible with a native application and not with a mobile theme for DSpace), receiving reading suggestions, apart from searching several libraries at the same time. As mentioned above, such application can be valuable for users in developing countries in their struggles to access library materials in a digital divide context, and can also bring value added to the users in places where such challenges are not a major concern.

3 Mobile Access to Digital Libraries

There has been some researches on making digital libraries available on mobile devices [1], [9], [10]. A good number of such researches have focused on extending the existing traditional libraries, by making additional services available on mobile devices. An analysis of such researches enables us to understand some best practices regarding to developing a mobile interface to digital libraries. Thus, before venturing into making a digital library available on these devices, it is important to consider some relevant questions: what are the lessons previously learned from these experiences? What could be done in a better way? The analysis of such experiences enabled the identification of some key learnings [10]:

- The project should be kept as simple as possible;
- A specific web service should be created to provide data to the application instead of bundling images and metadata with the application; When the application is to be downloaded by third parties (such as App Store), it is important to take into consideration the corresponding requirements and processes [10] [11].
- It is imperative to understand the potential users beforehand, and therefore to know who they are, what their needs are, what functionalities they want, which mobile devices they use, etc.
- The full desktop site should not necessarily be made available to mobile users. Typically only a subset of functionalities is made available according to user's needs. However, the response they get from the mobile site or application should not be incomplete when compared to the desktop version.
- When offline access is an important prerequisite or some mobile specific features are needed (such as camera, user location, offline reading, etc.), building a mobile application is recommended in detriment of a mobile Web site.

– Considering the specificities of mobile environments and equipments, a very special attention should be given to the usability and interaction design.

As previously mentioned, some traditional libraries have made part of their services available to users via mobile web sites, aiming at bringing added value to the users of these libraries. Apart from these initiatives, there are also some existing artifacts, not necessarily developed by traditional libraries, that allow users to access publications, references and other resources on their mobile phones. Some of these artifacts are [14]:

– Mendeley - enables users to manage their references and to carry their personal digital library on mobile devices. It can synchronize with the web version as well as desktop version of the application. Among other functionalities, users are able to see the metadata and to read publications on mobile devices.
– Papers - this commercial system allows users to manage references, to search papers on major databases such as ACM, IEEE Xplore, Google Scholar, etc. Downloading and offline reading possibility are also available along with other functionalities.

There are also other initiatives aiming at making library resources available over mobile phones: the development of a version of the Greenstone digital library system for iPod-Linux and Android powered mobile phones, enabling the installation of the entire digital library on these devices which might work as a server for the DL over a local area network; the release in 2012 of a mobile theme for DSpace-based digital libraries, enabling a better access to this system, when using mobile devices. However, it is important to highlight a big difference between Greenstone, DSpace and the above mentioned tools (Mendeley and Papers): while Greenstone and DSpace are digital library systems and therefore library-side tools, Papers and Mendely are user-side tools, used for searching, references management, etc.

These and other existing mobile artifacts are not directed towards profiting from popular open source-based digital library systems such as Fedora, ePrints, etc. Apart from Greenstone (and lately, DSpace), they do not provide mobile interfaces to any digital library based on these open source systems which are very popular all over the world. The majority of such digital libraries make contents available according to the Open Access Philosophy which advocates free access to scientific publications for all. Given the developing countries context of shortage of financial resources and difficulties in accessing printed-based publications, such resources are particularly relevant for them [2]. Therefore, considering that the usage of open source digital library systems is a must in developing countries [15] and taking into account the great advantage brought by the Open Access resources in these regions [2], facilitating the access to such DLs seems to be really important. Mobile access can bring such contribution.

With the release of a mobile theme for DSpace a few months ago, mobile access to this digital library system becomes much easier since users can use their mobile phone browsers to access papers and other materials available, in a more user friendly manner then when using the desktop web version of these libraries on their mobile devices. However, an eventual native application would provide a different set of advantages that cannot be found in mobile web. For example, such applications would be installed on the mobile phone or tablet, enabling offline reading of downloaded materials and the usage of specific device features such as camera and user location. This offline reading possibility is particularly relevant in developing countries where Internet access is often a real challenge.

Thus, this paper presents a design of a native mobile phone application (for Adroid and IOS devices) that enables on one hand, a mobile access to the digital libraries that are based on a popular open source system (DSpace), and on the other, provides several additional functionalities, such as offline reading, automatic reading suggestions according to user's profile, sharing of resources among users, etc. With this application, users have the possibility of potentially using all the existing DSpace-based digital libraries at the same time, from a single point and perform searches on one as well as on all of them simultaniously.

4 Designing the Mobile Application

In order to design this application, a specific set of procedures was followed so that the result could be valuable to the users and reflect their needs and expectations. Thus, a loop of 4 stages of design research activities was carried out, as advocated by [8]:

1. Contextual Inquiry - defines the context and the preliminary design challenges. It involves interacting with different representatives of potential users groups and writing fictional narrative descriptions about them. These narratives, called "personas", contain detailed typical characteristics, representing distinct grouping of behavior that were identified during the research [5].
2. Participatory Design - defines the context and the preliminary design challenges. Stories or scenarios are defined in order to imagine ideal user/system interaction. These scenarios are used for defining system requirements.
3. Product Design - defines the use cases and basic interaction. User stories on the usage of the system are created, as well as a prototype.
4. Software prototype - At this stage, a number of prototypes are delivered until a feature-rich application.

The following subsections describe the stages followed in this design process.

4.1 Contextual Inquiry

The core idea related to the development of this application is making library resources (that are available in DSpace-based digital libraries) more accessible to users, especially in a digital divide higher education context, where there are challenges on accessing printed publications as well as ICT resources. Thus, a user study was carried out in a developing country scenario characterized by difficulties in accessing printed publications as well as low Internet and computer penetration rates. This study was carried out at the University Jean Piaget in Cape Verde, and aimed at understanding who the users were, what their needs were, which functionalities they would like to see available, etc.

Cape Verde is a 10 islands archipelago with 491875 inhabitants, 17,2% illiteracy rate and a US$3737 GDP per capita[2]. 32% of the Cape Verdean population has regular Internet access and 79.2% own a mobile phone. However, computers are only available at 12.3% of homes [6]. The country faces serious problems in accessing both printed and digital publications. There is no national network for book trade and the first digital library in the country was built at the University Jean Piaget of Cape Verde in 2008, using DSpace.

For collecting information on user requirements, devices and needs, a survey was conducted among the potential users at the University Jean Piaget, where there was already a DSpace-based digital library available with the members of the University's academic community as the main target public.

The survey took place between January and February 2012, and involved 312 potential users among the University's 2100 students and 180 lecturers. People from all areas responded to the questionnaire which was divided into four main parts: (1) ICT access; (2) usage of mobile devices; (3) accessing online academic materials; (4) expectations. Some key findings of the survey are summarized below:

- In general, users did have access to computer (86.5%) and Internet (69.7%), at least at the University; almost all of them (97.4%) owned a mobile phone; iPhone was the most popular smartphone among them, and was owned by 16% of the respondents to the survey.
- The majority of users (75%) did not own a smartphone, but within 6 months time, about 69% of them would probably own a smartphone, an iPod touch or a tablet, since they had the intention of acquiring at least one of these devices.
- 78% of the users did not have the habit of reading academic materials on their mobile phones due mainly to unavailability of WiFi/mobile network, slow connection speed, application unavailability and small screen size.
- 73.7% of the users would use a digital library on a mobile device. They would like the materials to be available in PDF, text, and HTML formats, with the possibility of performing offline content reading.

4.2 Personas

Apart from this survey, additional information on the DL usage was also taken into consideration so that a better understanding of the potential users could be reached: a previous user study on the existing DSpace-based digital library at the University [13], the updated DSpace usage statistics, the updated Google

[2] `http://www.ine.cv/censo/censo2010.aspx`, April 23 2013.

Analytics statistics on the DL, ICT usage indicators in the country [6], etc. Such user study and information enabled a better understanding of the potential users. Therefore, after analyzing all these studies and data, 4 groups of users were identified and 4 personas were created (two primary personas and two secondary personas) along with their corresponding descriptions. Persona descriptions are used mainly to contextualize the usage scenario and to narrow down the scope of the solution which is proposed [7]. Some key characteristics and requirements of these personas are:

1. Bachelor student 1 - 21 years old; does not have Internet access at home; owns a smartphone; wants to search the DL and download materials for offline reading.
2. University lecturer - 34 years old; has computer and Internet access at home as well as an Internet connected smartphone; wants mainly to perform searches on a single as well on several DSpace-based DL at the same time (federated search); is interested in receiving automatic reading suggestions and on indicating reading materials to students.
3. Bachelor student 2 - 24 years old; has computer and Internet access at home as well as smartphone; is mainly interested in searching thesis and dissertations for reading.
4. Foreign user - 24 years old; regular smartphone user for searching and reading materials on the go; wants a reading list of downloaded materials and the possibility highlighting texts and taking notes for each material that has downloaded.

4.3 Scenarios and User Stories

After creating the personas, and following up with the design methodology advocated by [8] and [5], persona-based scenarios were created to describe how users would interact with the system. Such scenarios are in fact brief narratives of one or more personas using the system to achieve a certain specific goals [5]. For this purpose, the characteristics and requirements of the personas are the main constraints. Three usage scenarios were created:

- *Pedro is currently working on his Bachelor thesis. The libraries he knows are very limited on sources and he does not have Internet access at home so that he can search and download materials online. Since he is now at the University where there is free WiFi Internet access, he wants to search for such materials and to download them for offline reading at home, so that he can work on his thesis.*
- *Antonio is a University lecturer. There are not many printed sources at the University where he works and his students have difficulties on accessing Internet at home. He needs to search good articles, thesis and other publications on several digital libraries, take some notes and recommend some reading materials to his students along with his notes for each resource.*
- *Manuela enjoys using her smartphone even for reading. She has even subscribed the Internet access service on her mobile phone. She was given an*

assignment at the University where she studies and she wants to search for credible materials in Portuguese but across several digital libraries while she is traveling home in the bus. She wants to read some materials download them for further readings, take notes and share these resources along with the corresponding notes, with other colleagues.

These scenarios along with the aforementioned personas, were then validated through interviews with potential users. After that, possible user stories were created. A user story contains a written description of what is necessary to have in a system, that might represent a user need or a description of system functionalities [4]. User stories are usually written from the user's point of view. A possible format is describing the user needs in a format such as: "As as [role] I want [goal/desire] so that [benefit]". The user stories identified in this project are:

1. *As a user I want to search digital library materials so that I can download them and do offline reading when no Internet connection is available.*
2. *As a lecturer I want to search digital library materials on the topics I'm supervising so that I can suggest trustable sources to my students.*
3. *As lecturer, I want to receive automatic reading suggestions so that I can be aware of interesting materials in my area of expertise.*
4. *As a user I want to see citation information so that I can properly cite the resources I read, on my thesis and academic assignments.*
5. *As a lecturer I want to send lists of reading suggestions to my students so that they can read them for their thesis and assignments.*
6. *As a user want to search and read digital library materials on the go, so that I can save time.*
7. *As a user I want to search information on several digital libraries at the same time in Portuguese language, so that I can find more resources.*
8. *As a user, I want to share my reading notes with others, so that they can know my impressions about the corresponding resource.*

5 The Mobile Application

Given the personas, scenarios and users stories, a set of functionalities was defined, and sketches, wireframes and mockups were created. A number of prototypes were then developed, formative evaluation sessions were conducted with expert users, including a librarian, and adjustments were made according to the results of these evaluations. The key functionalities of the application were therefore defined during the design process and then implemented:

– Search for resources. m-DSpace allows users to search either in a single DSpace-based digital library or in several libraries at the same time. Therefore, from this single application, users can access resources available in several DSpace-based digital library, without the need of accessing each one's web site individually.

- Download materials for offline reading. When searching for sources, users can access the metadata about these resources, and eventually download the full materials for further reading. The downloaded materials and the corresponding metadata are stored in a database on the mobile device.
- Manage the reading list of downloaded materials, including rating, flagging and deleting. While reading offline, users are able to highlight texts, and take notes. The reading list should be organized by users and in this process they can insert the downloaded materials into the collections they create on their devices.
- Send reading suggestions to others users. Users can share one or several resources with others by sending emails containing links to the resources. They have the option of sharing their notes on the resources. Social networks such as Facebook and Twitter can also be used for this purpose. Users also have the possibility of sharing their reading list items with neighboring friends over a WiFi network. Therefore, if one user has a certain resource on his device, there is no need for his neighboring friend to go online, spend bandwidth and download the same material. He can simply make this material available to his friends over a WiFi network.
- Export resources. Users can export their reading list items to other applications (Kindle, Evernote, Dropbox, etc.) they might have in their devices.
- Receive automatic reading suggestions provided by the back-end service, according to the user profile and the usage statistics of available materials. Since when signing up users provide the system with fields of interest, and other profile information, m-DSpace back-end service can send regular reading suggestions according to their needs and taking into consideration other indicators such as ratings, usage statistics, etc.

Apart from facilitating the process of finding and sharing resources, these recommendations and reading suggestions can also encourage more trust on the system. The absence of trust (untrustworthy systems) can induce inefficiencies, encourage protective and unproductive actions and complicate interactions [3]. Users sometimes miss important system features which might prove useful to them, because they tend to protect themselves from software that seem inefficient, that are untrustworthy or from unsafe information sources. By including these two functionalities in the system design, we also intend to explore this trustful feature in supporting users interactions (either interpersonal or system based relationships). From our perspective, these functionalities address the interpersonal relation perspective and the system user relation perspective: from the interpersonal relation side, users can select the resources they find important and share them with others; from the system side, the recommendations will be based on user profile, access statistics and others user's inputs on a particular resource.

The application was initially developed for iPhone, since this was the most used smartphone among the personas. However, the dynamic of Android powered devices are undeniable: in 2011 for example, the Android powered mobile phones had a 244% annual growth and represented 48,8% of global smartphone shipments[3]. Taking this factor into consideration, an Android version of m-DSpace was also developed. Figure 1, shows two views of the application (reading list view and metadata view), for Android and IOS.

When building a mobile application, a specific web service should be created to provide data to the application instead of bundling images and metadata with the application [10]. Thus, for building m-DSpace, a back-end application was developed. This service receives connections from the user's device, performs the requested operations and sends the results back. Some of the key functionalities of this back-end application are: (1) searching one or several Dspace-based DLs; (2) creating automatic reading suggestions according to the user profile and usage statistics; (3) creating the usage log which is then used for producing the usage statistics and reading suggestions, etc. Therefore, the architecture of m-DSpace includes three main levels: mobile application level, back-end application level and DSpace level (figure 2). The mobile interface interacts with the back-end service and the back-end service interacts with the DSpace-based digital libraries.

Fig. 1. IOS and Android versions of m-DSpace

In order to enable the interaction between the user's devices and the back-end application, a XML-RPC API was developed. Thus, the services provided by the back-end application can also be used by any other application through the usage of this API which includes functions for creating accounts, logging on, searching, managing reading list, receiving reading suggestions, etc. The entire back-end server was implemented in Python3 using postgreesql database.

[3] `http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011`, April 29, 2013.

The database of the back-end application was initially populated with the names and URLs of all the DSpace-based digital libraries that were published in the dspace.org web site. After that, the metadata of each one of these libraries were automatically harvested by the back-end application and inserted into the database. This process was accomplished using two different approaches: for all those libraries where the OAI-PMH was enabled, the metadata were fetched using this protocol. But for those where OAI-PMH was not enabled, HTML parsing was used for metadata harvesting. The DSpace's browsing functionality over the web interface was used and the results were automatically parsed using the XPathQuery library.

The different libraries are regularly scanned for new records and eventual changes on the existing records. The back-end system's database is updated regularly using either the OAI-PMH or HTML parsing. Detecting the records that were changed in a library since the last update, using OAI-PMH is a relatively easy process. However, performing such task using the DSpace browsing interface over the web along with HTML parsing is more time consuming.

The user has the possibility of searching for resources in a single library or in all libraries at the same time. When a single library is chosen, the back-end application performs the search using the regular DSpace web search functionality for the selected library. Thus, in this case, the search is performed according to the indexes of that library and a full-text search might be performed (if the library is full-text indexable). However, when the user chooses to search over all the libraries at the same time, such search is performed on the metadata that were indexed by the application, the repetitions are eliminated, the results are sorted and sent to the user's device.

Apart from being available as native application for Android and IOS, a web version of m-DSpace was also developed, enabling the user to perform the same type of operations available on mobile devices, on any Internet connected machine with a web browser. This web version contains all the materials that are

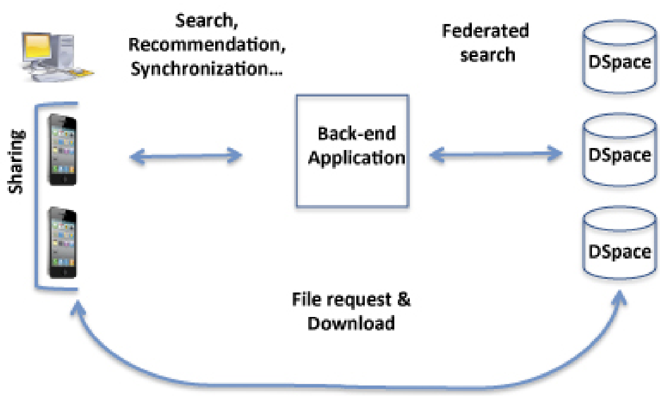


Fig. 2. System Architecture

available on the mobile device. Therefore, all the items that are available in the reading list (including the collections that the user can create to organize his reading list), can also be accessed and managed by the user on the web version. These materials are accessible for reading and management on the mobile device when the user is offline and when he gets connected to the Internet, any eventual change to this reading list is then synchronized with the one which is also stored by the back-end application. However, the files of each one of these resources are only stored on the mobile devices. The back-end application contains the metadata and a link to the original source, i.e. to the library where they were originally found. If the user wants to read the resource over the web, he will have to access it through that link. This approach helps saving bandwidth and storage space. Figure 3 shows the output of a search performed over the web Interface.

As previously mentioned, the core idea of this project is bringing a contribution to minimizing the issues of accessing library materials in digital divide contexts, especially in developing countries where there are problems, on one hand, related to the failure of the traditional mechanisms for distributing printed publications [15], and on the other, there are challenges related to accessing ICT resources. However, this system can also be used in regions where these issues are not a major concern. Thus, some improvements to this system are currently being implemented: a version specifically designed for tablets (Android and IOS) as well as a desktop version. This desktop version which is being developed in Java, will be able to provide all the functionalities that are available on the mobile version. Therefore, the reading list (including metadata and full-text files) will also be available on the user's computer, apart from the web and mobile devices. But the desktop user will also be able to benefit from other functionalities such as exporting metadata to reference management systems. The m-DSpace's

Fig. 3. m-DSpace Web

architecture (figure 2) with a back-end application and an API for performing all the most relevant operations, accelerates tremendously the process of developing new interfaces and functionalities.

6 Conclusion

This paper has presented an alternative way of accessing DSpace-based digital libraries, using mobile devices. Such access mechanism seems to be relevant for both, developing and developed world. The possibility of downloading resources for reading on mobile devices, the federated search over distinct digital libraries, the possibility of sharing resources among users, the possibility of receiving automatic reading suggestions, are some of the functionalities of this system. The next stages of this project include an evaluation on the level of acceptance and usage of this solution in a higher education context of a developing country scenario. Therefore, important aspects that should be taken into consideration when making digital libraries available in mobile devices should be unveiled, including the way users perceive and accept such alternative access mechanism.

References

[1] Bainbridge, D., Jones, S., McIntosh, S., Jones, M., Witten, I.H.: Portable digital libraries on an ipod (2008)
[2] Chan, L., Costa, S.: Participation in the global knowledge commons: Challenges and opportunities for research dissemination in developing countries. New Library World 106(1210-1), 141–163 (2004)
[3] Constantine, L., Dubois, E., Pohl, K.: CAiSE 2006. LNCS, vol. 4001. Springer, Heidelberg (2006)
[4] Cooper, A.: An Insurgency of Quality. Code Freeze (2009)
[5] Cooper, A., Reimann, R., Cronin, D.: About Face 2.0: The essentials of interaction design, vol. 3. Wiley Publishing, Inc., Indianapolis (2007)
[6] ITU: Measuring the Information Society. International Telecommunication Union, Geneva (2012)
[7] Laanpere, M., Tammsaar, K., Sousa, S.: A Case Study on Using Social Media for e-Participation: Design of Initiative Mapper Web Service. In: Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance, pp. 289–292. ACM, Tallinn (2011)
[8] Leinonen, T., Toikkanen, T., Silfvast, K.: Software as hypothesis: research-based design methodology. In: PDC 2008 Proceedings of the Tenth Anniversary Conference on Participatory Design 2008, pp. 61–70. Indiana University, Indianapolis (2008)
[9] Marsden, G., Cherry, R., Haefele, A.: Small screen access to digital libraries (2002)
[10] Mitchell, C., Suchy, D.: Developing Mobile Access to Digital Collections. D-Lib Magazine 18(1/2) (2012)
[11] Nielsen, J.: Usability of Mobile Websites and Applications - Design Guidelines for Improving the User Experience of Mobile Sites and Apps (2011)
[12] Rosa, I., Lamas, D.: Digital Libraries in the developing countries. In: Kommers, P. (ed.) Proceedings of IADIS International Conference on e-Society, pp. 502–504. IADIS, Libon (2007)

[13] da Rosa, I.B., Lamas, D.R.: Uso de Bibliotecas digitais versus tradicionais em contextos de info-exclusão. In: Sistemas e Tecnologias de Informação. Actas da 4 Conferência Ibérica de Sistemas e Tecnologias de Informação, pp. 331–336. AISTI FEUP UFP, Porto (2009)
[14] Vaidhyanathana, V., Moorea, M., Lopera, K.A., Schaika, J.V., Goolabsingha, D.: Making Bibliographic Researchers More Efficient: Tools for Organizing and Downloading PDFs, Part 1. Journal of Electronic Resources in Medical Libraries 9(1), 47–55 (2012)
[15] Witten, I.H.: Digital libraries for the developing world. Interactions 13(4), 20–21 (2006)

Demo: Professor2Student – Connecting Supervisors and Students

Feidias Ioannides[1], Georgia M. Kapitsaki[1], and Nearchos Paspallis[2]

[1] Department of Computer Science, University of Cyprus,
75 Kallipoleos Str., Nicosia, Cyprus
{fioann04,gkapi}@cs.ucy.ac.cy
http://www.cs.ucy.ac.cy

[2] School of Computing and Mathematics, UCLan Cyprus,
12-14 University Avenue, Pyla, Larnaca, Cyprus
npaspallis@uclan.ac.uk

Abstract. The wide spread of mobile platforms has brought a wide range of applications for the nomadic user addressing different domains. Universities constitute an ideal environment for the creation and introduction of novel applications that can facilitate the interaction among the members of the university community. In this demo we introduce a context-aware mobile solution that addresses the communication between academic faculty members or research associates and their supervised students. Professor2Student integrates a number of tools that allow an easier and closer collaboration between supervisors or academic advisors and students building on widely adopted protocols and a context-aware middleware for the Android platform.

Keywords: Mobile applications, Context, Location-awareness.

1 Introduction

The wide spread of mobile platforms (e.g., Android, iOS, Windows Mobile) has shifted the interest of providers at offering nomadic users with a wide range of applications. A significant portion of the created market offers location-based or context-aware applications [1] exploiting the sensors embedded on mobile terminals (e.g., accelerometer) [2]. At the same time, different communities can profit from specialized applications for their members. The university community with the interactions among different groups is an ideal environment for this kind of specialized mobile applications. In this framework initiatives from different universities around the globe can be found, such as the mobile applications offered by the University of Alabama, the Texas A&M University, the Ohio State University and the Massachusetts Institute of Technology. These applications cover different aspects of the university life: information for the university, registration to courses, information and notifications on bus schedules. In the same spirit as the aforementioned applications - but for a different aim - we have created the application *Professor2Student* for the Android platform

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 288–291, 2013.

targeting the communication between specific groups: students and professors (or research associates acting as thesis supervisors or academic advisors). The application contains useful tools that render the communication between users fast and reliable by building on existing protocols, whereas it exploits the following context information for its use: professor/supervisor location and student academic status. In this demo we will showcase different uses of *Professor2Student* and allow users to interact with the application.

2 Professor2Student and Demonstration Example

Professor2Student is composed of a number of modules with each module offering a specific application functionality. Module interconnection is depicted in the architectural diagram of Fig. 1. Really Simple Context Middleware (RSCM) [3] offers an extensible abstraction layer for accessing context resources while also providing a mechanism for context-based reasoning. It consists of the RSCM library which is used at development time, and the RSCM runtime which is installed alongside the application. The RSCM runtime consists of a service that runs in the background and manages the activation and communication between the context producers (plugins) and the context consumers (applications). Context plugins are individual, reusable components used to produce context data directly from underlying sensors or by reasoning on existing, primitive data. In *Professor2Student*, RSCM is used to provide the position of the professor. Extensible Messaging and Presence Protocol (XMPP) [4], also referred to as Jabber, is a protocol for real-time communication. XMPP is mainly used for instant messaging but offers also presence detection and data transfer. *Professor2Student* uses XMPP for instant messaging between professors and students, as well as in the sharing presence information (chat module). In order to store and retrieve data to and from our server's database an implemented RESTful web service is used (database connector module). Encoded SQL queries are sent through HTTP POST requests, whereas the results are returned serialized to JavaScript Object Notation (JSON) format.

In the demonstration of *Professor2Student* we will present different cases of interaction between a supervisor and a student. Consider for instance Kostas supervised for his diploma thesis by Prof. Andreou. Fig. 2 shows the use of the system by the supervisor for the following actions: creation of thesis completion stages, meeting arrangement and location detection. Viewing the application use from the professors perspective, Prof. Andreou selects initially the *Stages* tab. Stages functionality gives the supervisor the possibility to define thesis stages that the student needs to complete in specified time intervals in order to follow the student progress. In order to create a new stage Prof. Andreou chooses from the contact list the student he wishes to create the thesis step for, selects the date of the thesis step, writes a brief description and submits the step. To arrange a first meeting with his student, Prof. Andreou uses the functionality *Arrange meeting* of the *Meeting* tab. Prof. Andreou would like his supervised students to know when he is available at his office, so he decides to set his office position

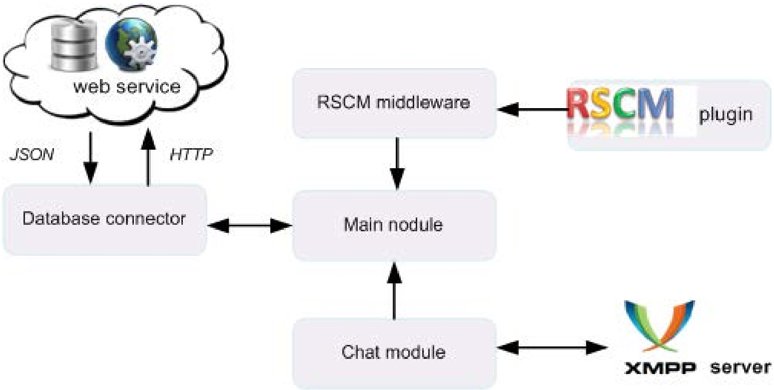


Fig. 1. Professor2Student architecture

by using the *Set office position* button. That position is saved and used later for comparison with the actual professor's position. For further guidelines Prof. Andreou can exchange instant messages with Kostas from the *Chat* tab. Viewing *Professor2Student* from the students perspective, as soon as Kostas opens the application he notices in the notification bar notifications about a new meeting proposal and a new thesis stage (Fig. 3). Kostas can click in the notifications to see more details and respond to meeting proposals or be redirected to the chat room with the supervisor. Although not visible in the figure, Kostas can also click the position button in the action bar to view whether or not the professor is in his office.

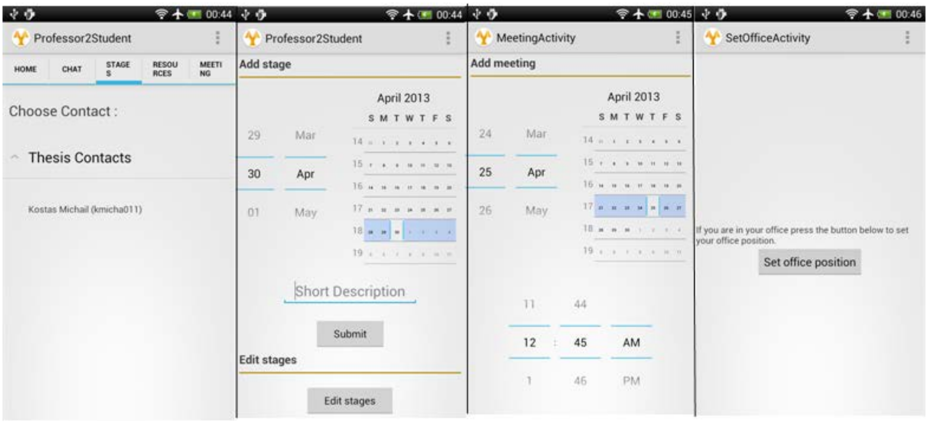


Fig. 2. Professor2Student scenario from the supervisors perspective

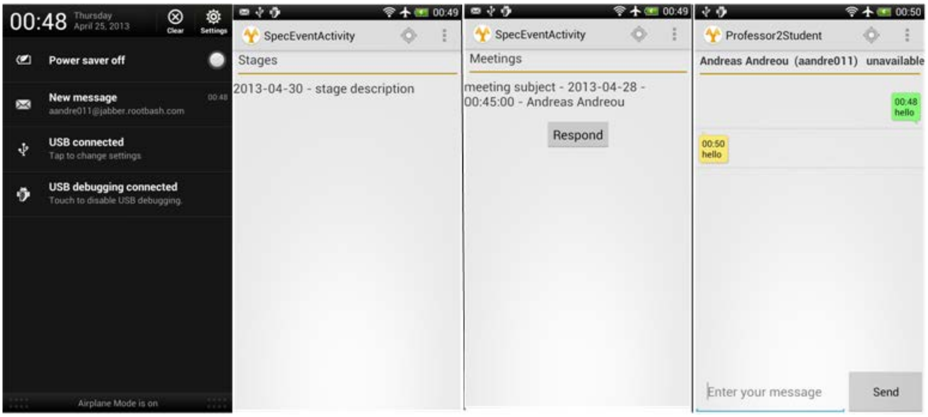


Fig. 3. Professor2Student scenario from the student's perspective

3 Conclusions

In this demo paper we have presented the *Professor2Student* application targeting the interaction between supervisors or academic advisors and students. The application consists of a number of tools, such as instant messaging and resource sharing that are built by the integration of widely employed protocols and middleware, namely XMPP for messaging and RSCM for context abstraction. The demo aims at demonstrating the usefulness of *Professor2Student* that is currently being used and assessed by the community of the University of Cyprus, whereas it is also available on Google Play.

References

1. Abowd, G.D., Dey, A.K.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999)
2. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. IEEE Communications Magazine 48(9), 140–150 (2010)
3. Paspallis, N.: Really Simple Context Middleware (RSCM) (2012), `https://code.google.com/p/rscm/`
4. Saint-Andre, P., Smith, K., Tronçon, R.: XMPP: The Definitive Guide: Building Real-Time Applications with Jabber Technologies. O'Reilly Media (2009)

Proximity: A Real-Time, Location Aware Social Web Application Built with Node.js and AngularJS

Ioannis K. Chaniotis, Kyriakos-Ioannis D. Kyriakou, and Nikolaos D. Tselikas

Department of Telecommunications Science and Technology,
University of Peloponnese
Tripoli, Greece
{tst08037,tst07086,ntsel}@uop.gr

Abstract. We demonstrate Proximity, a real time social web application built with avant-garde open source tools that make true real-time web communications, cross-platform compatibility, rapid development and service efficiency, possible.

Keywords: node.js, AngularJS, javascript, websockets, real-time social web.

1 Introduction

People nowadays communicate and interact online through social media and web enabled devices are becoming extremely commonplace, with mobile devices leading the way [1]. This makes it easier for one to keep in touch with acquaintances and follow interesting content. However, this perspective is not the most suitable for meeting people one doesn't already know, or attract attention without perpetual content input. We observed a lack of simplicity and of forthwith communication features in current social applications, as they tend to be bloated with features and content.

The Social web depends on mobile and web-based technologies, to create highly interactive platforms through which, individuals and communities share, co-create, discuss and modify user-generated content [2]. Two of the most successful examples of the real-time web are Facebook's newsfeed and Twitter. They both achieve a real-time effect by utilizing HTTP polling techniques [3,4]. A closely related revolution has been the emergence of websockets as part of web standards [5], which demonstrate superior efficiency compared to any HTTP polling techniques [6].

Proximity is a social web application for meeting new people, close to the user, in real-time, utilizing websockets. It just uses the current user location as input, as well as a picture the user chooses to represent himself/herself. It was designed to be swiftly developed, modular and offer high performance and interoperability across devices. It utilizes modern concepts that could benefit current and future social applications. These design aspects are presented in this paper, followed by simulation benchmarks.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 292–295, 2013.

2 System Architecture and Measurements

Fig.1. provides a schematic representation of the system architecture. The front end is structured with AngularJS [7]. Video calls are established using webRTC, which enables peer-to-peer connections behind NAT/firewalls without plugins [8]. At the back end, which is powered by the node.js framework [9], socket.io and express.js provide socket and web services functionality, respectively [10-11]. The entire network traffic is relayed through an instance of HAProxy [12], for security and throttling reasons.

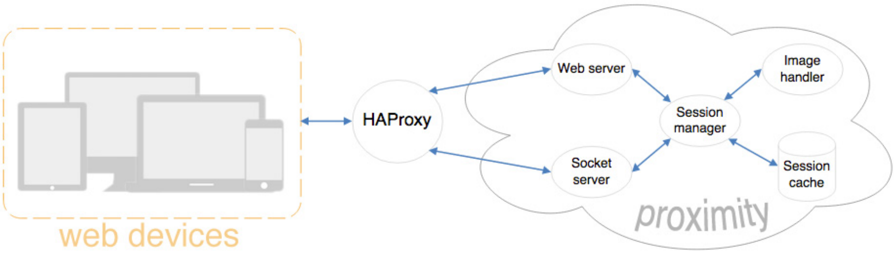


Fig. 1. System architecture

A series of tests were performed to evaluate server response time, resource utilization and stability under different load scenarios. The testbed is based on Intel Core i7 920 @ 2.6GHz with 4GB DDR3 @ 1066 MHz, Ubuntu 12.10 kernel 3.5.0-17-generic, while jpeg images used, were 431.1KB @ 1280x1024. The test module creates a client pool, simulates the login procedure and a messenger session. A launcher spawns test processes recursively, with the number of concurrent connections and the arrival rate of new users' connections fluctuating as parameters.

Concurrency was evaluated for up to 1300 concurrent connections, with new connection interval set to 500ms. As depicted in Fig.2, the memory required for new connections drops almost exponentially, proving node.js can easily scale.

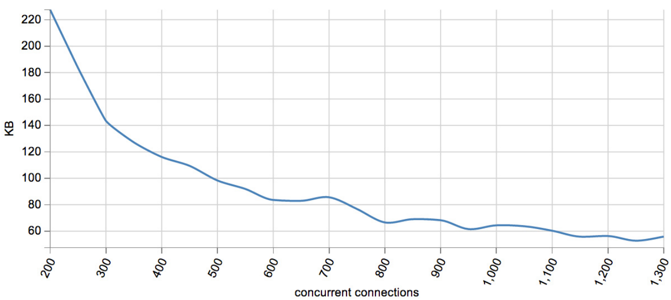


Fig. 2. Required Memory per connection vs. Concurrent connections

For connection rate measurements, new connection interval ranged from 500 (2 new connections/sec) to 90ms (11 new connections/sec), with a fixed pool of 500 concurrent existing connections. Login process time took values between 204 and 217 ms during the whole range of concurrency tests, but increased significantly, when the interarrival time between new connections, crossed the threshold of 150 ms (Fig.3).

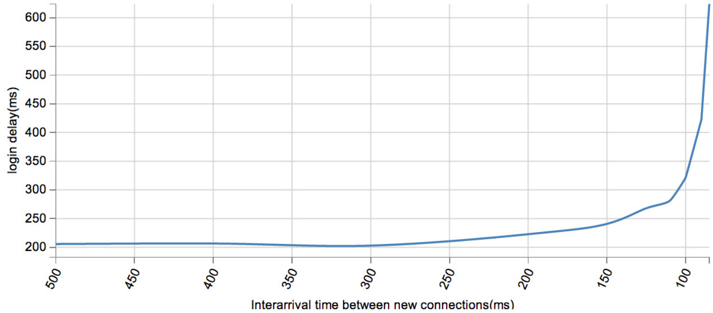


Fig. 3. Login delay vs. interarrival time between new connections

3 Demonstration

Attendees will be asked to participate at the demonstration. The application flow breaks down into three states, i.e. *login*, *online* and *busy*, respectively.

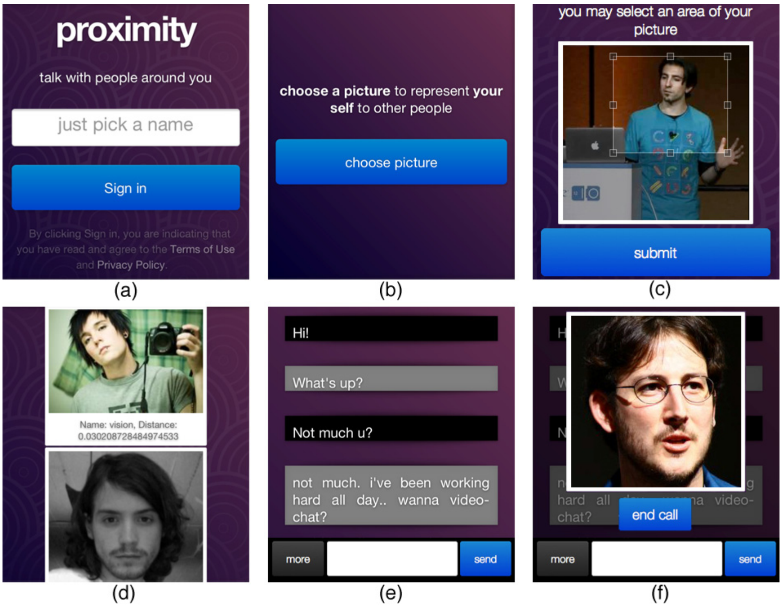


Fig. 4. Proximity screenshots

During *login* state (Fig.4.a-c) the users are prompted to enter a name, a picture, which can be cropped, or enhanced with effects, and give access to geolocation API. The users are then presented in *online* state, with a list containing other people sorted by distance proximity. Thereafter they can select, or be selected by other *online* users (Fig.4.d). When a two-sided selection occurs, the users are displayed in *busy* state; they are transferred to a private room and can either send text messages to each other (Fig.4.e), or start a video call (Fig.4.f). When a messenger or video session gets terminated, both users go back to *online* state.

4 Conclusions/Future Work

Writing in JavaScript at both the front and back ends, enables the porting of client side modules to the server side and vice versa. Socket.io's event driven approach and AngularJS' two-way data binding, reduce significantly the SLOC. Although WebRTC is still under heavy development and not fully standardized at the time, it's features surpass the implementation difficulties. The result is a modular, easy to debug and maintain, real-time web application featuring great performance, as presented.

Future work will include clustering support, in order to apply stress tests and evaluate Proximity, in a more reliable and also realistic environment. Finally, our future plans include further improvements on technicalities, such as to use Redis key-value store for session storage and expose an API, rendering Proximity as efficient as possible.

References

1. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update (2012–2017), http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html
2. Kaplan Andreas, M., Michael, H.: Users of the world, unite! The challenges and opportunities of social media. Business Horizons 53(1), 59–68 (2010)
3. Tornado: Facebook's Real-Time Web Framework for Python, https://developers.facebook.com/blog/post/301/
4. Real-Time Delivery Architecture at Twitter, http://www.infoq.com/presentations/Real-Time-Delivery-Twitter
5. The WebSocket Protocol, http://tools.ietf.org/html/rfc6455
6. HTML5 Web Sockets: A Quantum Leap in Scalability for the Web, http://www.websocket.org/quantum.html
7. Angular.js conceptual overview, http://docs.angularjs.org/guide/overview
8. WebRTC 1.0: Real-time Communication Between Browsers, http://dev.w3.org/2011/webrtc/editor/webrtc.html
9. Node.js, http://nodejs.org/
10. Express.js, http://expressjs.com/
11. Socket.io, http://socket.io/
12. Kaushal, V., Bala, A.: Autonomic Fault Tolerance using HAProxy in Cloud Environment. Int. J. of Advanced Engineering Sciences and Technologies 7(2), 54–59 (2011)

Towards a Mobile Technical Customer Service Support Platform

Michael Fellmann[1], Deniz Özcan[1], Michel Matijacic[2], Gerald Däuble[2], Michael Schlicker[3], Oliver Thomas[1], and Markus Nüttgens[2]

[1] University of Osnabrück, Information Management and Information Systems, Katharinenstraße 3, 49074 Osnabrück, Germany
[2] University of Hamburg, School of Business, Economics and Social Sciences, Department of Socioeconomics, Von-Melle-Park 9, 20146 Hamburg, Germany
[3] INTERACTIVE Software Solutions GmbH, Universität Campus Nord Scheer Tower, 66123 Saarbrücken, Germany
firstname.lastname@uni-osnabrueck.de,
firstname.lastname@wiso.uni-hamburg.de,
firstname.lastname@interactive-software.de

Abstract. Service technicians in the Technical Customer Service (TCS) domain have to deal with different duties at the point of service in a short time. To tackle these challenges, information systems have to be developed to support their everyday work. In the joint project EMOTEC we aim to improve the efficiency of service technicians by intelligent mobile assistant systems. Key component of our system is an integration platform for effective data storage and -integration that facilitates repair processes. The platform under development provides flexible support and has the potential to both increase the productivity and the empowerment of the service technician.

1 Introduction

For many manufacturers the TCS became a major value-adding resource. Each and every service technician is the cornerstone of this value delivery. They have to go beyond just complex technical issues but also need to deal with business aspects of spare parts logistics, guarantees and billing as well as financing and efficiency analyses. For customer loyalty their competence is of vital importance. At the mobile “Point of Service”, case-related expert and context information is needed immediately [3]. Thus, implicit expertise and individual experience need to get externalized and connected with recorded service information. The increasing importance of servicing makes its engineering and management to a necessity for staying competitive in the service industry. To manage these services, the right information systems need to be developed and have to be adopted since the adoption of information technology has an impact on the productivity of enterprises [4]. In this way, the EMOTEC-demonstrator offers professional support via intelligent mobile assistant systems intended to increase efficiency and empowerment. By the latter, we mean the degree of autonomy and self-determination of the service personnel to plan and perform tasks.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 296–299, 2013.

2 Use Cases of the Platform

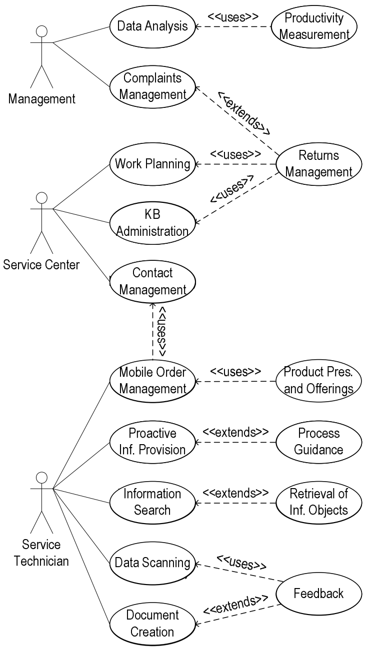


Fig. 1. Use Cases

Towards the more empowered and productive service technician an appropriate mobile assistance system is needed to support the various TCS processes. Unfortunately, the development of such systems is a complex and complicated task since diverse requirements have to be considered spanning technical aspects such as interfaces or integration technology [2]. In order to reduce the complexity of the system design task, we have identified 16 use cases as the nexus of our development effort (cf. Fig. 1). The use cases have been identified (a) by observing the daily work of service technicians and their usage of mobile TCS support systems during 77 complete service process executions and (b) by examining the results of a structured literature review on requirements for and applications of mobile systems in the TCS domain. The use cases have been documented using the template provided by Cockburn [1]. As it can be seen from Fig. 1, the system has to serve three distinct stakeholders: The *Management*, the *Service Center* and the *Service Technician*. Some of the use cases make use of other use cases indicated by an <<uses>>-relation, whereas others are specialized variants indicated by <<extends>>. In the remainder, we focus on the description of the architecture and features of the platform developed to support the use cases.

3 Architecture of the System

The EMOTEC-architecture (cf. Fig. 2) consists of three parts: the *Application Environment*, the *Integration Platform* and both *Mobile and Stationary Clients*. In the application environment, different enterprise application systems could be used as a source, e.g. *Enterprise Resource Planning Systems*, *Customer Relationship Management Systems*, *Product Lifecycle Management Systems*, *Business Intelligence Systems* or *Condition Monitoring Systems*. The *Integration Platform* plays a central role in the underlying architecture since it integrates the data from the different applications. Also, it provides functions for the analysis of data, proactive functions for the intelligent use of data over the whole service process as well as collaborative functions to easily collaborate with other service technicians. The integration platform has a *Productivity Measurement* module, which measures Key Performance Indicators (KPI) for every service process. Furthermore, an *Application Programming Interface* (API)

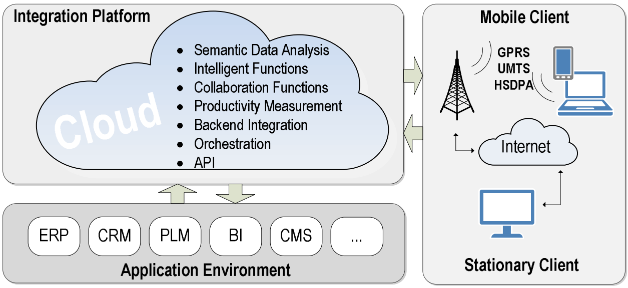


Fig. 2. EMOTEC-Architecture

provides the possibility to communicate with other software components. As a unique characteristic, the integration platform can be used as a cloud service over the internet. The third part of the architecture is the presentation layer. Both service technicians and the back office could use services from the integration platform over a dedicated Graphical User Interface (GUI). At the "Point of Service", the service technician could use a *Mobile Client* or a *Stationary Client* to interact with the integration platform. All information the service technician needs for his/her every day work can be provided by that application. On the opposite side the back office works with the same application, but with different permissions.

4 Technical Customer Support Features

In order to achieve an intuitive and flexible access and usage of the platform, a "Mobile App" on a tablet computer for mobile access or a web application on the laptop for stationary access can be used. The mobile client supports online/offline use and runs on top of Android or iOS thereby re-using features such as GPS, navigation, camera and address book. The web application has been developed using the modular framework "RedPanda" that integrates with ERP-systems via a REST-API or Web Services. Fig. 3 shows several combined screen shots of sample applications.

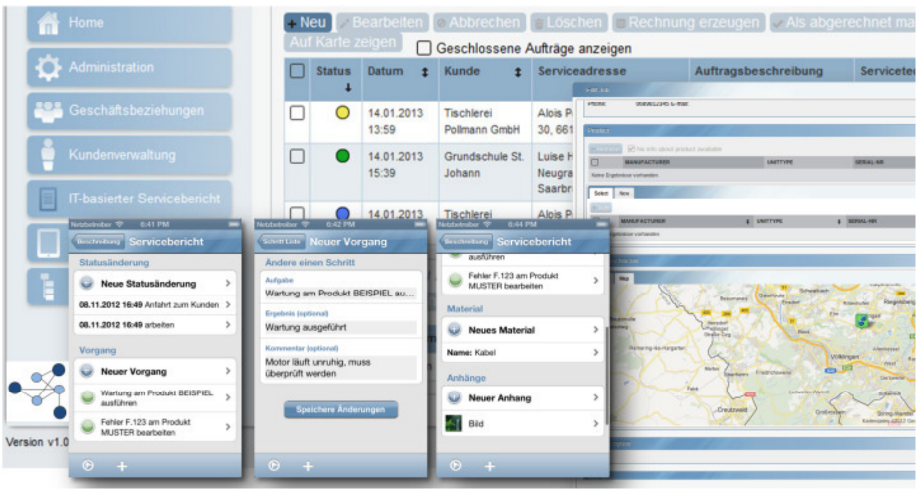


Fig. 3. Stationary Web-based Client (Back) and Mobile Application Screens (Front)

In the following, we briefly describe a quick tour through our platform touching on selected use cases. The service technician can create a new request or he can be assigned to an order by another service technician. If a service order is assigned to a service technician, he can access the order on his mobile client where the relevant information for the service process is displayed according to the use case *Proactive Information Provision*. The service technician has the option to accept or reject the service order. By doing so, he changes the processing status from “assigned” to “accept” or “reject”. The application corresponding to the use case *Feedback* captures feedback data to be processed in the service center. This gives the service technicians the opportunity to send the service center important information concerning the service process, the service object or the customer. When the service technician has finished his work, a service report can be created which is highly automated with predefined information in conjunction with context-sensitive fill-out-assistants for forms (cf. Fig. 3 bottom right mobile application screen) which implements the use case *Documentation Creation*.

5 Future Work

Currently, the EMOTEC-platform is thoroughly evaluated within the joint project and its value partners belonging to the domain of industrial trucks. Moreover, it has been presented at the well-known fair “Hannover Messe” in Germany in April 2013 to gain insights from potential stakeholders within the technical customer service area (e.g. technicians, managers, etc.). This feedback which has been quite positive will be incorporated into the further development of the platform. To validate the usability of the system which is important for technology adoption, eye-tracking tests are planned. Moreover, show cases of using the platform are planned by the end of 2013. This will enhance the understanding of how to purposefully apply the platform.

Acknowledgement. This contribution originates from the research project EMOTEC (Empower Mobile Technical Customer Services), funded by the German Federal Ministry of Education and Research (BMBF), ref. no. 01FL10023.

References

1. Cockburn, A.: Writing effective use cases. Addison-Wesley, Boston (2001)
2. Fellmann, M., Reinke, P., Matijacic, M., Schlicker, M., Thomas, O., Nüttgens, M.: Customer-oriented configuration and specification of mobile application systems for improved TCS productivity and empowerment. In: Proc. of the XXII. Int.l RESER Conf. on Services and Economic Development, Bucharest, Romania, September 20–22, p. 126 (2012)
3. Fellmann, M., Hucke, S., Breitschwerdt, R., Thomas, O., Blinn, N., Schlicker, M.: Informationssystemarchitekturen zur Unterstützung technischer Kundendienstleistungen. In: Bernstein, A., Schwabe, G. (eds.) Proc. of the 10th Int.l Conf. on Wirtschaftsinformatik (WI 2011), Zurich, Switzerland, February 16-18, vol. 1, pp. 252–261 (2011)
4. Kwon, M., Stoneman, P.: The Impact Of Technology Adoption On Firm Productivity. Economics of Innovation and New Technology 3, 219–233 (1995)

A Personal Mobile Academic Adviser

Vangel V. Ajanovski

Faculty of Computer Science and Engineering,
Ss Cyril and Methodius University, Skopje, Macedonia
vangel.ajanovski@finki.ukim.mk
http://www.finki.ukim.mk/en/staff/vangel-ajanovski

Abstract. The semester enrollment is one of the most important university processes. The students submit an enrollment application with a selection of courses, that goes under review by the academic advisers. The introduced virtual academic adviser aims to alleviate typical problems in the process and enables the student to make an informed choice, aided by measures of success and social navigation towards more streamlined selection. An estimated future study plan until graduation is given to the student, based on the average speed of studying. The adviser enables the student to perform what-if analysis, change the load, switch to another program, even try a migration to a new institution. The application was tailored to the requirements of students while mobile, which is of special importance due to the existance of concurrent administrative processes happening at various locations even outside of campus.

Keywords: course enrollments, academic performance, student information systems, course recommendations.

1 Introduction

One of the major administrative processes at universities – the semester enrollment, starts few weeks before the start of each following semester and usually ends a week after the start of the courses. In this short period all the students have to submit an enrollment application for the next academic semester with a selection of courses. The application is then reviewed by the academic adviser and approved for enrollment. Each student can choose a list of courses to enroll and is free to decide on the number of courses and thus, on the total work load for the following semester.

Ever changing requirements, mandate more and more free choice in the study programs. Currently, we have study programs where only 50% of the courses are obligatory, while the rest may be chosen from other programs in the same institution or even from any program active at the university. More choice means more directions where a successful career could lead. On the other hand the choice introduces an important side-effect – more responsibility at the hands of the student, on making a wiser choice. In a country where the unemployment rate is more than 30%, and where one is never sure if he will be able to have a successful start of the career and earn his living, such choices can make a significant psychological burden.

F. Daniel, G.A. Papadopoulos, P. Thiran (Eds.): MobiWIS 2013, LNCS 8093, pp. 300–303, 2013.

2 The Role of the Academic Adviser and Introduction of the Virtual Adviser

The role of the academic adviser is to help and guide the students towards their interests, by making the most adequate course selections and in that way leading the students to a successful completion of their studies. The academic advisers have an important place in the process, and they cannot be appointed from all of the staff, since offering a relevant guidance to the student requires having a more extensive academic and professional background.The main reason for this is the knowledge about the study matter and the intricacies and interdependencies. Fig. 1 presents a graph of dependencies between courses, showing only two types of interdependencies, and for only one study program, whereas many departments can have even 10 or more different study programs. Also there can be other types of prerequisites for the courses. The full graph of associations between courses at our faculty has more than 8000 edges.

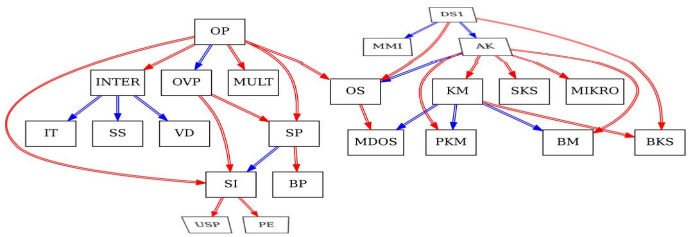


Fig. 1. Two types of dependencies in a 3 year study program

An academic adviser will have to know and understand the reasons for the complexities and what kind of effect they have on the choices of each student. To ease the whole process of enrollment we have built an Integrated Studies Information System (ISIS) [1]. In this system the students send their enrollment applications online, and they are immediately given an analysis of satisfied and unsatisfied course interdependencies. Still this left much room for improvement. In a previous study we explained the full background behind the need of a true academic adviser that will be able to help each and every student and we proposed the introduction of virtual academic adviser [2] in the ISIS system. Despite being only a simple component in the online system that would count and sum credits, and let the student make his decisions, it would still be able to give the student more personal guidance in the process, than any real adviser. In the first iteration it gave the student a dashboard for monitoring his own succes, showing a map of all of the previously enrolled semesters and the courses selections in each semester – visually distinguishing the successful courses from unsuccessful ones. It also offered a view into the future, a forecast of semesters to come and a proposed plan of studying until the graduation, taking into account all the necessary prerequisites.

The latest version of the virtual academic adviser can be seen in Fig. 2. The lower part represents the past and the upper part gives the student an estimate of the future until graduation. The semesters in the future are mapped according to the average speed of studying of all students. Each row represents a semester, and each box in the row is a course enrolled in that semester.

The semesters are ordered from recent to historical in downwards direction. Each box shows: the name of the course, certification that the student was present at lecture hours, final grade of the student, and ECTS credits per course. The boxes are colour coded: green boxes represent currently active courses (in the on-going semester), orange boxes are courses where the lectures have finished but the student did not have a chance to take an exam yet, white boxes represent courses that were finished successfully, and red boxes represent courses that the student has failed.

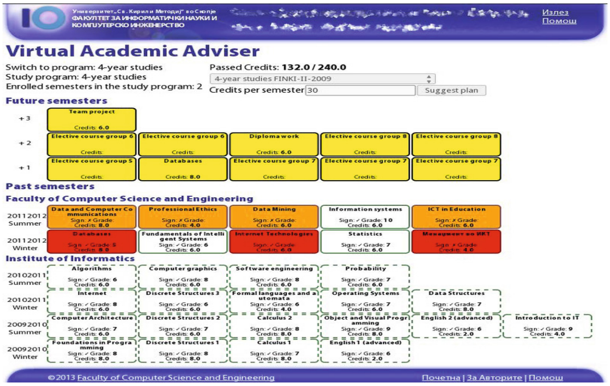


Fig. 2. The latest version of the Virtual Academic Adviser

The system takes into account all interdependencies and course prerequisites and will propose a *realistic plan*. Whether this plan will succeed depends only on the ability of the student to follow and stay through with the plan.

This dashboard gives the student measures of his success in comparison to other students. The student can compare the speed in acquiring credits to averages of per generation. The system also visually indicates courses that are critical – where there was a significant amount of failing grades and where students had to re-enroll a course after failing. The virtual academic adviser also gives some controls to the student to run what-if analysis and see a new plan according to a change of program, or speed of studying. The system immediately computes the new forecast and stores it so the student can discuss it with the adviser.

The mobile access is crucial feature of the system, because the enrollment process coincides with the exam period for the previous semester. The students need to have access to the plans when they are on the go, so they can discuss the proposal with their colleagues and advisers. The system needs to be completely online due to the large database of course associations that requires massive computational power to process and everyday changes in grades and measure of success (since the process coincides with the end of exam period). Custom built applications are not suitable since this system is not used everyday, only in three week long periods, two times a year. Several features have been implemented to enable the usage when mobile and improve the performance:

- Single GUI for mobile, tablet and desktop users. Many students have old devices with only a simple browser
- The layout is computed dynamically per user – it resizes according to browser type, resolution, portrait/landscape. The figure shows the layout prepared for a 7" tablet in 800x480 resolution (bottom of Fig.2 is visible after scrolling)
- The plans can be precomputed and stored. So repeated execution does not affect performace unless the parameters have changed. Students may start the planning on a desktop, and later they can see the plan and discuss it with colleagues, while on the go.

3 Work towards a Socially Adaptive Virtual Adviser

Still, the process is not more personalized than before. The students' applications are again processed automatically without regards to personal requests. The system gives more tools at the hands of the students, increasing their responsibility of a smart decision. We have already started work towards introduction of a recommendations systems [3], based on general recommendations algorithms (such as [4]) but integrated with ACM and AIS curriculum guidelines [5] in order to lead the student to better accomplishment of his career goals.

References

1. Ajanovski, V.: Information System of the Institute of Informatics: IS for Students by the Students. In: 7th International Conference for Informatics and Information Technology, p. 129. Institute of Informatics, Skopje (2010) ISBN 978-9989-668-88-3
2. Ajanovski, V.: Towards a Virtual Academic Adviser. In: 8th International Conference on Informatics and Information Technology, pp. 146–149. Institute of Informatics, Skopje (2011) ISBN 978-9989-668-90-6
3. Ajanovski, V.: Personalized Adaptive System for Term Enrolments Based on Curriculum Recommendations and Student Achievement. In: Proceedings of the IADIS International Conference Information Systems 2013, pp. 342–346. IADIS Press (2013) ISBN 978-972-8939-83-0
4. Aher, S.B., Lobo, L.: Applicability of data mining algorithms for recommendation system in e-learning. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp. 1034–1040. ACM (2012) 978-1-4503-1196-0/12/08
5. Topi, H., Valacich, J.S., Wright, R.T., Kaiser, K.M., Nunamaker Jr., J.F., Sipior, J.C., De Vreede, G.J.: Curriculum guidelines for undergraduate degree programs in information systems. In: Association for Computing Machinery (ACM) and the Association for Information Systems, AIS (2010), `http://www.acm.org/education/curricular-recommendations` (retrieved March 4, 2013)

Author Index