THESIS ON INFORMATICS AND SYSTEM ENGINEERING C75

Towards Adaptive Web - Analysing and Recommending Web Users' Behaviour

TARMO ROBAL

TUT PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Computer Engineering
Chair of System Programming

Dissertation was accepted for the defence of the degree of Doctor of Philosophy in Computer and Systems Engineering on June 06, 2012.

Supervisor: Prof. Ahto Kalja,
Dept. of Computer Engineering, Tallinn University of Technology

Opponents: Prof. Juris Borzovs
University of Latvia, LATVIA

Prof. Hannu Jaakkola
Tampere University of Technology, FINLAND

Defence of the thesis: August 22, 2012

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for any academic degree.

/ Tarmo Robal /

Euroopa Liit
Euroopa Sotsiaalfond

Eesti tuleviku heaks

ISSN 1406-4731
ISBN 978-9949-23-331-1 (publication)
ISBN 978-9949-23-332-8 (PDF)

INFORMAATIKA JA SÜSTEEMITEHNIKA C75

Veebikasutajate käitumise analüüs ja soovitused adaptiivse veebi loomiseks

TARMO ROBAL

TTÜ
KIRJASTUS

To my family

ABSTRACT

The Internet has grown into a sophisticated set of resources providing an ever-increasing amount of information, leaving users to face data presented in diverse forms, and introducing problems of information overload and its successful retrieval. Alongside search engines, features of adaptive web, recommender systems and web personalization are seen as solutions to the aforementioned problems, since users are browsing the Web according to their informational expectations while having a sort of implicit conceptual model in their mind, shareable with other website visitors.

This thesis is dedicated to investigations on improving web information systems and providing web personalization on the basis of users' behaviour modelling. In particular, the thesis concentrates on anonymous ad-hoc web users and collective intelligence obtained from their behaviour.

Firstly, the problem domain is described and methods of obtaining necessary knowledge about users' behaviour are disclosed. An original log system and analyzer have been established for the task.

Secondly, a methodology to evaluate web information systems against usability issues on the basis of post-modelling users' actions is presented. Several metrics are introduced to identify problem areas and help to minimize the gap between the conceptual models applied by developers and exploited by actual users. Web improvements will be applied as strategic web adaptation.

Thirdly, a novel framework for computing recommendations for anonymous ad-hoc web users is presented. The recommendations are established on the basis of users' domain model and activity prediction model. Both of the models are based on new methods for users' behaviour modelling. The established recommendations allow to offer users personalized web experience recognizing their interests and intentions on the Web, deliverable as tactical web adaptation.

The proposed methods described in the thesis are seen suitable for websites and portals of general information, but not only. They can be applied also in systems where users are identifiable and thereby a lot more information about them becomes available. The experiments performed on web usage data have proven the feasibility and efficiency of the proposed methods.

KOKKUVÕTE

Internetist on tänaseks päevaks saanud kiiresti arenev mitmesuguste ressursside võrgustik, mis võimaldab ligipääsu üha suuremale informatsiooni hulgale. See protsess on aga jätnud kasutajad vastakuti olukorraga, kus informatsioon on esitatud väga erineval kujul ning valitseb teabe ülekĂĽllus. Lahendust antud probleemile nähakse nii veebi otsingumootorites kui ka adaptiivses veebis realiseerituna soovitussüsteemide ja veebi personaliseerimise näol. Viimane põhineb väitel, et kasutajad lehitsevad veebi lähtuvalt hetke infovajadustest ja teatud kontseptuaalsest arusaamast valdkonnast, milles infot otsitakse. See on aga osaliselt kattuv teiste veebikasutajate käsitlusega otsitava info valdkonnast.

Käesolev väitekiri käsitleb veebiinfosüsteemide parendamist ja veebi personaliseerimist läbi kasutajate käitumise mõistmise ja modelleerimise. Antud töö keskendub anonüümsetele veebikasutajatele ja nende veebikäitumise modelleerimisele eeltoodud eesmärkide saavutamisel.

Esiteks on antud ülevaade tööga haakuvast probleemistikust ja võimalustest koguda informatsiooni kasutajate veebikäitumise kohta. Selleks on realiseeritud vastav logimissüsteem koos andmete töötlemise ja analüüsimise süsteemiga.

Teiseks on esitatud veebipõhiste süsteemide hindamise metoodika nende kasutatavuse seisukohast. Antud metoodika põhineb kasutajate käitumise järelmodelleerimisel. Välja on pakutud mitmed mõõdikud läbi mille süsteeme hinnata ja tuvastada neis probleeme. Kõnealuse metoodika väljund võimaldab lähendada arendajate ja tegelike kasutajate käsitlust süsteemist, kohandades süsteemi just kasutajate nägemusele läbi vastavate strateegiliste muudatuste.

Kolmandaks on välja töötatud raamistik veebisoovituste leidmiseks anonüümsete veebikasutajate jaoks. Kõnealune soovitussüsteem tugineb kahele töös uudsena kirjeldatud metoodikale, mille väljunditeks on veebikasutajate valdkonnamudel ning kasutajate tegevuste ennustamise mudel. Soovitussüsteemi poolt leitud inforessursside hulk võimaldab veebikasutajatele pakkuda personaliseeritud veebikogemust lähtuvalt just konkreetse kasutaja huvidest ja kavatsustest veebiinfosüsteemis.

Välja töötatud meetodid on ennekõike suunatud veebisaitide ja portaalide parendamiseks ning nende kasutajatele personaliseeritud sisu kuvamiseks. Samas on need meetodid edukalt rakendatavad ka mitmesugustes teistes infosüsteemides, kus kasutajatel palutakse end eelnevalt identifitseerida, ning seeläbi omab süsteem juba rohkem informatsiooni kasutaja kohta. Uurimustöö käigus teostatud analüüsid ja eksperimendid on tõestanud kõnealuste meetodite teostatavust ja efektiivsust.

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to everyone who has contributed, supported and advised me during my PhD studies.

In particular, I would like to thank my supervisor professor Ahto Kalja for providing advice, guidance, and support throughout my PhD studies, research work covered in this thesis, and for encouraging me to become a member of the international research community.

I would also like to thank all the people I worked with during my PhD studies, especially the whole team at the Department of Computer Engineering. I would like to express my gratitude to Dr. Margus Kruus for his support and warm attitude when dealing with many practical and administrative issues. I would also like to acknowledge Helena Kruus, Janari Põld, Elmet Orasson, and Prof. Peeter Ellervee for their support and contribution.

I am deeply grateful to Dr. Hele-Mai Haav from Institute of Cybernetics at Tallinn University of Technology for her guidance and support.

I also want to acknowledge several organizations that have supported my PhD studies: Tallinn University of Technology, National Graduate School in Information and Communication Technology (IKTDK), Centre for Integrated Electronic Systems and Biomedical Engineering (CEBE), Centre of Research Excellence in Dependable Embedded Systems (CREDES), and Estonian Information Technology Foundation (EITSA).

I would also like to thank all my friends and colleagues not mentioned above.

Finally, I would like to thank my family for their continuous encouragement, patience and support. Thank you!

Tarmo Robal,
Tallinn, June 2012

LIST OF SELECTED PUBLICATIONS

Studies on Web Users Behaviour

Robal, T., Kalja, A., Learning from Users for a Better and Personalized Web Experience, *PICMET '12 Conference "Technology Management for Emerging Technologies"*, July 29 - August 2, 2012, Vancouver, Canada, PICMET: USA, 10p. [to be published]

Robal, T., Kalja, A. Applying User Domain Model to Improve Web Recommendations. Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 14p. [to be published]

Robal, T., Kalja, A. Web systems evaluation on users' behaviour modelling. *Databases and Information Systems V: Selected Papers from the Eighth International Baltic Conference, DB&IS 2008*. H-M. Haav,A. Kalja (Eds). Amsterdam, IOS Press, Frontiers in Artificial Intelligence and Applications, 187, 2009, pp. 41-52.

Robal, T., Kalja, A. Conceptual web users' actions prediction for ontology-based browsing recommendations. *Information Systems Development: Towards a Service Provision Society*. G. A. Papadopoulos, G. Wojtkowski, W. Wojtkowski, S. Wrycza, J. Zupancic (Eds). New York: Springer-Verlag, 2009, pp. 121-129.

Robal, T., Kalja, A. A model for users' action prediction based on locality profiles. *Information Systems Development: Challenges in Practice, Theory and Education*. C. Barry, K. Conboy, M. Lang, G. Wojtkowski, W. Wojtkowski, Wita (Eds). New York: Springer, Vol. 1, 2009, pp. 169-182.

Robal, T., Kalja, A. Evaluations for improving web systems on the basis of users behaviour modelling. *Databases and Information Systems: Proceedings of the Eighth International Baltic Conference, Baltic DB&IS 2008*. H-M. Haav, A. Kalja (Eds). Tallinn: Tallinn University of Technology Press, 2008, pp. 229-240.

Robal, T., Kalja, A. Applying user profile ontology for mining web site adaptation recommendations. *Local Proceedings of the 11th East-European Conference on Advances in Databases and Information Systems (ADBIS 2007)*. Y. Ioannidis, B. Novikov, B. Rachev (Eds). Varna, Bulgaria: Technical University of Varna, 2007, pp. 126-135.

Robal, T., Haav, H-M., Kalja, A. Making web users' domain models explicit by applying ontologies. *Advances in Conceptual Modeling - Foundations and Applications : ER 2007 Workshops CMLSA, FP-UML,ONISW, QoIS, RIGiM, SeCoGIS*. J.-L. Hainaut (Ed). Auckland, New Zealand, Berlin: Springer, 2007, Lecture Notes in Computer Science, vol. 4802, pp. 170-179.

Robal, T., Kalja, A., Põld, J. Analysing the Web Log to Determine the Efficiency of Web Systems. *Seventh International Baltic Conference on Databases and Information Systems*. O. Vasilecas, J. Eder, A. Caplinskas (Eds). Vilnius, Lithuania, Technika, 2006, pp. 264-275.

e-Learning

Robal, T., Kann, T., Kalja, A. An ontology-based intelligent learning object for teaching the basics of digital logic. Proceedings of 2011 IEEE International Conference on Microelectronic Systems Education (MSE). IEEE Computer Society, 2011, pp. 106-107.

Robal, T., Kalja, A. Creating interactive learning objects with web services. Proceedings of the 20th EAEEIE Annual Conference. Valencia, Spain. IEEE Publishing 2009, 2009, pp. 1-6.

Robal, T., Kalja, A. Interactive Hands-On Tools as Learning Objects on Web Services. Proceedings of International Conference on Microelectronic Systems Education MSE'09. San Francisco, CA, USA. IEEE, 2009, pp. 73-76.

Robal, T.; Kalja, A. Making use of personalized web services in the study process. Proceedings: 11th Biennial Baltic Electronics Conference BEC 2008. Tallinn, Estonia, 2008, pp. 211-212.

Robal, T., Kalja, A. Enabling students contemporary ways of learning using e-supported courses. 19th EAEEIE Annual Conference, Formal Proceedings. The 19th European Association for Education in Electrical and Information Engineering (EAEEIE) Annual Conference. Tallinn, Estonia, IEEE, 2008, pp. 14-19.

Robal, T., Kalja, A. Applying e-environments in teaching the basics of digital logics. Proceedings of the IEEE International Conference on Microelectronic Systems Education MSE 2007. P. Kellenberger (Ed).San Diego, CA, USA. IEEE Computer Society Press, 2007, pp. 41-42.

Robal, T., Kalja, A. Moving studies to e-environments: a case study. Current Developments in Technology-Assisted Education. A. Méndez-Vilas, M. Solano Martín, J.A. Mesa González, J. Mesa González. Badajoz, Spain: Formatex, 2006, pp. 936-940.

Robal, T., Kalja, A. e-EDU - an information system for e-learning services. Selected Papers from Sixth International Baltic Conference DB&IS'2004. J. Barzdins, A. Caplinskas (Eds). Amsterdam: IOS Press, Frontiers in artificial intelligence and applications, vol. 118, 2005, pp. 288-298.

Robal, T., Kalja, A. e-EDU - an information system for e-learning services. Acta Universitatis Latviensis, vol. 672, 2004, pp. 469-480.

e-Government and Services

The following articles bridge the research presented in this thesis and the future potential applicability of the described methods for the Estonian Governmental Portal (eesti.ee) to deliver its visitors personalized view to portal content. Negotiations on these topics were initiated in February 2012.

Kalja, A., Põld, J., Robal, T., Vallner, U. Modernization of the e-government in Estonia. *PICMET '11: Proceedings Technology Management in the Energy-Smart World.* D. Kocaoglu, T. Anderson, T. U. Daim (Eds). Portland, OR, USA. PICMET, 2011, pp. 3151 - 3157.

Kalja, A., Robal, T., Vallner, U. Towards information society: Estonian case study. *Proceedings of PICMET '09: Technology Management in the Age of Fundamental Change.* D. Kocaoglu, T. Anderson, T. U. Daim (Eds). Portland, Oregon USA. PICMET, 2009, pp. 3218-3225.

Kalja, A., Kindel, K., Kivi, R., Robal, T. eGovernment Services: How to Develop Them, How to Manage Them? *Proceedings of PICMET '07: Management of Converging Technologies: PICMET '07 Portland International Center for Management of Engineering and Technology*. D. Kocaoglu, T. Anderson, T. U. Daim (Eds). Portland OR, USA, IEEE, 2007, pp. 2795-2798.

LIST OF ABBREVIATIONS

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CIM	Conceptual Interest Model
CLF	Common Log Format
CMS	Content Management System
CSS	Cascading Style Sheets
DAML	DARPA Agent Markup Language
DB	Database
DBMS	Database Management System
DCE	Department of Computer Engineering
DOM	Document Object Model
HCI	Human–Computer Interaction
HTML	HyperText Markup Language
HTML5	HyperText Markup Language redaction 5
HTTP	Hypertext Transfer Protocol
IC	Interest Concept
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IR	Information Retrieval
IS	Information System
ISP	Internet Service Provider
JSON	JavaScript Object Notation
LMS	Learning Management System
LRU	Least Recently Used
OIL	Ontology Interchange Language
OS	Operating System
OWL	Web Ontology Language
OWL-DL	OWL Description Logic variant
PE	Prediction Engine
PHP	Hypertext Preprocessor
RDF	Resource Description Framework
RDFS	RDF Schema

RIC	Referred Interest Concept
ROI	Return on Investment
RS	Recommender System
RSS	Really Simple Syndication
SEO	Search Engine Optimization
SOAP	Simple Object Access Protocol
TSP	Time Spent on Pages
TUT	Tallinn University of Technology
UDDI	Universal Description Discovery and Integration
UDM	Users’ Domain Model
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VLE	Virtual Learning Environment
W3C	World Wide Web Consortium
WIS	Web Information System
WPM	Words per Minute
WSDL	Web Services Description Language
WUM	Web Usage Mining
WWW	World Wide Web, also known as the Web

TABLE OF CONTENTS

Chapter 1 Introduction 23

1.1 Motivation.. 23

1.2 Problem Formulation 25

1.3 Thesis Contribution.. 26

1.4 Thesis Organization 27

Chapter 2 Preliminaries 29

2.1 Data Collection 29

2.2 Web Usage Mining 31

2.3 Adaptive Web 33

2.4 Web Personalization 34

2.5 User Profile 37

2.6 Ontologies Represent Domain Knowledge 38

2.7 Privacy Issues Concerned with Personalization.............................. 41

2.8 Related Works.. 42

2.9 Chapter Summary 48

Chapter 3 Logging Web Events 49

3.1 Introduction.. 49

3.2 Collecting Web Events 51

3.2.1 Web Server Logs 52

3.2.2 Hosted and Third-Party Analytics Solutions 54

3.2.3 Other Issues to Consider with Web Events Logging.................. 55

3.3 Overview of the Main Research Environment................................ 56

3.4 Developed Special Log System and Log Analyzer.......................... 58

3.4.1 Log System Architecture and Implementation 58

3.4.2 Log Analyzer System 62

3.5 General Analyses on Log Data 65

3.5.1 Viewing Experience: Browsers, OS's, and Screen Size.............. 65

3.5.2 Source of Visitors 68

3.5.3 Access Trends on Timescale of 24-7-12 70

3.5.4 Menu Structure Level and Page Load Analysis 70

3.5.5 Search Combinations in Sessions 72

3.5.6 Correlation Study on Session Parameters.................................... 73

3.6 Discussion.. 74

3.7 Chapter Summary 75

Chapter 4 Evaluating and Optimizing WIS's on the Basis of Users' Behaviour 77

4.1 Introduction.. 77

4.2 Indicators of User Interest in WUM 78
4.3 Metrics of WIS Evaluation on Users' Behaviour 83
4.3.1 Steps of Data Processing for WIS Evaluation 83
4.3.2 The Locality Model 84
4.3.3 Pages of Interest.. 87
4.3.4 Void Operations in Navigation Paths 91
4.3.5 Multi-Sequential Identical Operations in Navigation Paths 93
4.3.6 Cyclic Operations in Users' Navigation Paths 94
4.4 Chapter Summary 97

Chapter 5 Learning Web Users' Domain Models from Browsing Behaviour 99
5.1 Introduction.. 99
5.2 Extracting User Preference Profiles From User Behaviour Data... 101
5.3 Web Ontology 103
5.4 Establishing User Profiles Ontology... 105
5.4.1 Predefined User Profiles 105
5.4.2 Extracted User Profiles 106
5.5 Automatic Classification of User Profiles 107
5.6 Applicability of the Method... 109
5.7 Chapter Summary 111

Chapter 6 Predicting Web Users' Actions.. 113
6.1 Introduction.. 113
6.2 A Sequential Model for Users' Action Prediction 114
6.2.1 Using the Sequential Model for Predicting Users' Actions....... 116
6.2.2 Experimental Results 117
6.3 A Conceptual Model for Users' Actions Prediction 121
6.3.1 Using the Conceptual Model for Predicting Users' Actions 123
6.3.2 Conceptual Distance of Requested Pages................................. 124
6.3.3 Experimental Results and Comparison of the Two Models 125
6.3.4 Discussion.. 128
6.4 Chapter Summary 130

Chapter 7 Providing Recommendations for Anonymous Ad-Hoc Web Users 131
7.1 Introduction.. 131
7.2 Web Recommendations Based on User's Behaviour.................... 133
7.3 Empirical Study on Refining Predictions for Web Recommendations 136
7.4 Exploiting the Recommendation Result 138
7.5 Chapter Summary 140

Chapter 8 Other Developed Web Information Systems....................... 143
8.1 Introduction.. 143
8.2 A Lightweight Learning Management System e-EDU................. 144
8.3 Learning Objects as Interactive Hands-On Tools 146
8.4 Chapter Summary 148

Chapter 9 Conclusions and Future Work 149
9.1 Conclusions 149
9.2 Future Work 152

References 153

Appendix A Log Analyzer System 167

Appendix B TSP Impact Study 170

Appendix C Example Web Ontology 171

Appendix D OWL2 Description Example 172

Appendix E General Prediction Algorithm... 173

Appendix F Recommendation Algorithm 174

Appendix G Learning Objects on Web Services.................................... 175

Chapter 1
INTRODUCTION

Today, the Internet provides us with an enormous ever-growing source of information and has become a crucial part of our everyday lives. At the time of its initiation, nobody could imagine its success and influences on the human kind. The World Wide Web, to which we commonly refer just as the Web, has gained tremendous popularity. We use the Web daily to search for information, read news, shop online, buy theatre and movie tickets, view and download digital media from books and magazines to music and videos, use online banking, and communicate with friends and colleagues through various channels, including holding video conferences. Even more, the advances of technologies allow citizens in many countries to relate to their governments using various e-government services starting with electronic tax declarations, digital signatures and ending up with e-voting, e-census, and sophisticated systems of e-health. The diversity of platforms and availability of the Internet has enabled a wide, not to say an endless range of online opportunities available to us and future generations.

1.1 Motivation

Over the time the Web has grown into a large distributed space of information resources in various forms, covered by billions of pages. While at the beginning the information was presented mainly in the form of simple static hypertext documents, the situation has dramatically changed and now we can talk about complex web information systems (WIS) – systems that use web technologies to deliver information to users and communicate with other systems [1]. Even more, the future is for the Semantic web, a Web of actionable information [2], [3], delivering the next generation of the Web as an extension of the current one with data semantics and easier ways to find, share, reuse and combine available information.

The large amount of information in diverse forms available on the Web today has led to information overload and introduced problems of successful information retrieval. Regardless of the exponentially growing information resources on the Web, humans' ability to perceive it however is and has remained individually limited. In other words, users are faced with information overload. To tackle this problem, traditionally users have turned to search engines to narrow down the information load based on their current informational interests. To some extent, search engines provide a solution to the problem; however they retrieve documents based on the keywords specified by user as a search criterion. As these keywords may have various semantic background or meaning, the results returned by search engines may not be of relevance to the user. Even more, the results returned by search engines are ordered based on a small amount of data available in the user's query and by website popularity rather than individual user interests, backgrounds, and context of the task the user is currently committed to. Therefore, to find the necessary piece of knowledge, users need to go through the returned search results manually. Also, a lot of information has been moved into social web environments, where new data is produced in collaboration between users, making orientation and search even more troublesome. This makes the situation even fuzzier and raises the question of how to successfully manage the large amounts of data available in an efficient and "painless" way.

When it comes to the way how information in WIS's is made available and represented, it greatly affects how users interact with systems and provokes either positive on negative attitudes towards their further use. Simplicity of navigation, general site attractiveness, visual experience, placement of objects, applied colour schema, and page loading time are only a part of factors that make up the attitude [4], [5], [6], [7]. Moreover, while browsing for information, different users have various previous experiences, cognition of the domain, and follow certain patterns of behaviour that are heavily dependent on their own approach to the subject area of interest. Also, users' needs tend to change over time. In the real world it is easy for users to explicitly state what they are searching for and how they want to do it. However, in the Web world, users' opportunities to express their needs are limited and thus their wishes may remain veiled. This makes designing complex websites a difficult task, where designers need to foresee users' informational needs based on the information being published, and establish a structure for the site accordingly.

In order to comprehend users' needs, it is required to understand how they browse available data and actually make use of WIS's. User tests can bring light into it but as they are expensive and time-consuming, they are often skipped. Nevertheless, users' intentions and behaviour can be revealed by investigating server or application logs and applying techniques of data mining. These logs embody valuable information about users for system designers and developers. On one hand, data mining can be used to learn general statistics and access trends for websites served by WIS's. On the other hand, this data can be used to study sites' usability, navigation, efficacy, and use these results to introduce improvements to better meet users' needs. Moreover, the data available in the logs enables to reason about users' behaviour, their browsing preferences and intentions, and construct user profiles to tackle the problem of information overload by enabling personalized view to websites.

Adaptive web and web personalization is seen as a solution to information overload problem and meeting users expectations. Personalization as the process of presenting the right information to the right user at the right moment [8] is becoming a vital component of every web information system aiming to offer effective and efficient services. Mulvenna et al. [9] have formulated the goal of web personalization as follows: "*provide users with the information they want or need, without expecting from them to ask for it explicitly*". As stated, the aim of any personalization system is to make users life easier by providing more relevant information without disturbing them, and thus enhancing users' web experience.

1.2 Problem Formulation

Regardless of advanced methodologies and technologies used for WIS development, users are searching and browsing the Web in unpredictable ways, heavily dependent on their own comprehension and conceptual model of the subject area, which may or may not partially overlap the general conceptual model applied by WIS developers. Each of these users has distinct background and specific needs for information. Thereby, there exists a well-known problem of possible mismatch between web users' domain model and website domain model. The mentioned gap between the designers' point of view and actual users' intentions and needs has created a demand for adaptive web, systems recognizing visitors' interests, and actions for web personalization.

Despite the fact that it is impossible to foresee how websites will actually be used and what will be the users' intentions, it is possible to learn about user's interests by investigating their actions afterwards and reasoning about those. The navigation traces users leave behind while browsing websites are valuable in optimizing WIS's and constructing conceptual models of subject areas known as users' domain models applicable for web personalization.

Although available technology would allow collecting of such data about users in various ways, there still exist problems of successful users' action capturing. Further, the massive amount of data needs to be somehow processed to obtain the necessary knowledge from it. This calls for web usage mining to get a better understanding of investigated web information system and its users. All this together is costly and time-consuming in comparison to general site statistics made easily available today by several free and commercial software tools running online, or enabling offline data processing.

A true challenge herein is the modelling of an anonymous ad-hoc web user, about whom there is no previous knowledge available in the system. Usually, personalization is applied in systems where users are somehow identifiable, for instance by means of logging in or user tracking. However, this dissertation concentrates on modelling users with applied privacy (no prior knowledge about a particular user is stored) and on the basis of collective intelligence obtained from WIS usage.

Still, once launched, little attention is paid to websites delivered by WIS's and their improvement based on their usage and users' interests. Thereby, the aforementioned conceptual gap remains discarded, which in the long run may lead to a loss of users that is of vitality, especially for systems involved with e-commerce. Evaluation and optimization of web information systems based on users' behaviour modelling is one of the research topics observed in this thesis.

Developing user models to provide recommendations for adaptive web, is not a trivial task. It requires careful modelling of web users, models of predicting their actions and ways to classify them. Several different approaches have been studied by researchers of the field, yet the best methodology and model for the task are still uncovered.

In this thesis the problems of successful user behaviour data capturing, web information system improvement based on users' behaviour, and learning users' domain models for visitors action prediction and recommendations generation are addressed. In particular, the dissertation concentrates on the ways how to better organize and deliver information on medium-sized websites to its visitors. The work aims to deliver simple yet effective methods for the listed tasks.

1.3 Thesis Contribution

The main contribution of the current dissertation is a novel approach for understanding users and learning their needs by examining and modelling their behaviour. In detail, the contributions are summarised as follows:

- Firstly, development of an original users' activity log capturing system and an analyzer system. The first system is used to record users' operations and applicable characteristics of accesses on websites. The latter on the other hand is applied for processing these records into a knowledge base and storing data in a coherent form. Processed and stored data is already able to answer the questions of general statistics.
- Secondly, establishment of a methodology to evaluate web information systems against usability issues and identify areas in necessity of improvement. The methods introduced are based on post-processing users' behaviour data and drawing conclusions based on that. The methodology proposes several metrics for WIS evaluation and strategic adaptation.
- Thirdly, a framework for learning users' domain models, delivering a method for attaining web user profiles by combining users' browsing behaviour with ontology-based modelling.
- Fourthly, a methodology for modelling web users' behaviour for predicting anonymous ad-hoc web users' activity. Two distinct models established for users' action prediction will be described.

- Fifthly, formation of a framework for a lightweight and yet effective and reliable recommender system for providing web adaptation and personalization for anonymous ad-hoc web users. The framework composes the delivered methods of web users' domain models construction and online visitor activity prediction into recommendation generation for tactical web adaptation for anonymous web users.
- Sixthly, establishment of an original learning management system e-EDU and several frameworks for digital learning objects to improve learning-teaching process and enable contemporary ways of learning.

1.4 Thesis Organization

This thesis consists of 9 chapters, covering the research topics and questions regarding web users' behaviour data collection and modelling, WIS evaluation and optimization, users' domain model establishment, web users' action prediction, and recommendation generation for anonymous ad-hoc web users (Chapters 3–7, presented in the order the research was conducted); and the author's research in the e-learning field (Chapter 8). The rest of this thesis is organized as follows.

Chapter 2 provides state-of-the-art background information on the research topics, covering issues of web mining, adaptive web, web event logging, essentials of recommender systems, and finally outlining related works of the field.

Chapter 3 concentrates on logging web events. The chapter starts off with describing the main research environment, followed by a discussion on developed special log and analyzer systems. This section also includes general statistics for the main research environment, introduced through several analyses. The chapter ends with a discussion and summary.

Chapter 4 is dedicated to evaluating and optimizing web information systems on the basis of users' behaviour. Several metrics for evaluation are introduced as a part of the framework. The chapter ends with a discussion on WIS evaluation.

Chapter 5 presents the method of learning users' domain models from browsing behaviour. It provides an overview of web ontology and user profiles ontology, used in the process of automatic classification of user profiles. Finally, a method for detecting a profile for online anonymous ad-hoc web user is presented.

Chapter 6 deals with ways of predicting online visitor actions based on collective intelligence. Two different prediction models are proposed: a sequential prediction model and a conceptual prediction model. For both of the models experimental results are provided and also a comparison of the models is presented.

Chapter 7 ties together the results from Chapter 5 and Chapter 6, and presents a framework of providing recommendations for anonymous ad-hoc web users for tactical website adaptation as a part of a personalization service. The method is proved by an empirical study on prediction refinement. The chapter ends with a discussion on method applicability.

Chapter 8 provides a short overview of other web information systems developed by thesis author, with the emphasis on e-learning, digital learning objects, and improving the learning-teaching process for students of information and communication technology.

Chapter 9 draws conclusions for the thesis and presents ideas for future research work.

There are 7 appendix sections (A–G) included at the end of the thesis.

Chapter 2
PRELIMINARIES

Studies of adaptive, personalized and intelligent web involve many disciplines, technologies and methodologies in reaching the aim of meeting web users' needs. This chapter is dedicated to provide sufficient background information on the state-of-the-art topics related to current research. The topics covered in this chapter are presented in the order similar to the process flow they are applied within the web studies described in forthcoming chapters.

Firstly, methods of attaining web usage data to base decisions on are presented. This is followed by a discussion on processing such data. In further, adaptive web, web personalization and recommender systems are introduced together with user modelling. As ontologies are exploited to model users' domain within the current framework, they are also covered in brief. A short discussion is provided on user privacy issues concerned with data collection. Finally, a summary on related state-of-the-art research works is outlined accompanied with significant results in the field.

2.1 Data Collection

In order to adapt web to its users' needs it is essential to learn visitors' intentions, interests and behaviour. The capabilities of the World Wide Web (WWW) provide an opportunity to collect such data at highly detailed level, with various approaches being applied for the task. Regardless of data collection form, completeness and accuracy of data is of main concern.

From the users' point of view data collection methods that can be applied vary from implicit to explicit, where in the latter active participation of users is crucial. For web users it means they have to spend some of their valuable browsing time besides information search to additional activities such as ticking checkboxes, filling in forms and providing feedback to web information system. For instance, users might be asked to evaluate on the scale of 1 to 5 how useful they found the page they just landed on. However, users are usually not willing to actively participate in such evaluations, moreover to fill in forms on every single page they visit. And yet, providing users a way to easily and unambiguously express their opinion has proven to be successful, especially on social networking sites. A prime example of such an explicit rating is the Facebook's 'Like' button, where users can with one simple click express their support to certain content. Nevertheless, Sarwar et al. [10] concluded in their study that even though explicit rating provided by users is precise, users are not willing to rate each page they visit, thus it cannot be as efficient as expected. Moreover, many authors [11], [12] oppose to explicit rating, claiming it to disturb and greatly affect users' browsing activities and behaviour. Despite their somewhat intrusive nature, explicit data collection methods are basically the only option to collect certain type of data about users such as their opinions, ideas, demographics, indicated interests, and so forth.

Implicit data gathering methods on the other hand provide an alternative to explicit methods and are based on automated data gathering. These methods are transparent for users, thus they do not disturb their browsing sessions and thereby help to overcome the disadvantages of explicit methods. Implicit methods generally involve data collection from server, proxy or client level using either web server logs, modified web browsers, or specially designed log systems [13], enabling, for example, easy monitoring of pages users' access, navigational paths they follow, and discovering of usage patterns. They also provide a rather good accuracy at the same time: Al halabi et al. [14] report an accuracy of more than 90% obtained by using implicit rating for inferring user's interest. Gauch et al. [15] argue that even though implicit methods are preferred for collecting information about users, they have a drawback of potentially displaying only positive feedback, as it remains unclear how to precisely indicate disinterest in comparison to interest. The research discussed in this thesis is based on implicitly collected user activity data.

Physically data collection can happen either on client side or on server side. Collecting data on client side usually requires cooperation from users. The techniques herein usually involve remote agents implemented in JavaScript or additional plug-ins that need to be installed at the time of using the site or beforehand. Probably the best example of a JavaScript based client-side data capturing is the Google Analytics[1] tool, which is widely exploited to obtain website usage statistics. For research purposes, special web browser agents [16], [17] and customized web browsers [18], enabling very precise data capturing, for example even to track mouse movements, have been used.

However, users may not always be willing to consent that their actions are followed and may block such scripts or plug-ins from running by employing special software to disable JavaScript, Java, Flash and other executables on web pages they visit, enabling them only according to their personal preferences. To exemplify, one of such tools is an open-source plug-in NoScript[2] for Mozilla-based browsers. Similar tools exist for other browsers as well. This may lead to incomplete data or inability to capture any user data, depending on the implemented method and technology used for data collection.

Server-side data collection mainly relies on logs, either web server logs or special application logs. Although a lot of research in the community has been

[1] http://www.google.com/analytics/

[2] http://noscript.net/

based on web server logs, it should be noted that due to the stateless nature of the HTTP protocol its traffic logs are flawed [19] and issues of data incompleteness exist [13], [20] with inability to identify visitor sessions [21]. As a consequence, pages requested by user during a session end up to be logged as individual accesses. Thus, they need to be bound together into sessions using available data (e.g. agent and IP-address) in the log before they can be meaningfully used. The latter is the main problem while dealing with web server logs. The discussion on this topic is continued in Chapter 3.

2.2 Web Usage Mining

With each page users request on the web they leave behind a set of information (clickstream data) characterizing them and their informational needs. This large set of data collected in a log describes various aspects of website usage and its users. However, before this vital knowledge can be attained, collected data in the logs needs to be processed. This is achieved by web mining – a form of data mining techniques to automatically process and discover useful information in large data repositories [22].

Researchers have provided various definitions for web mining. One of the earliest definitions provided by Etzioni [23] says that web mining is the use of data mining techniques to automatically discover and extract information from web documents and services. Srivastava et. al. [24] have refined the definition as follows: web mining is the application of data-mining techniques to extract knowledge from web data, in which at least one of structure or usage (web log) data is used in the mining process (with or without other types of data). The other types of data are not specified. Depending on the mining task these sources can be website content, structure, operational databases, or semantic knowledge in the form of ontology, allowing to enrich the various aspects of user activity data. The purpose of web mining is to develop methods and systems for discovering models of objects and processes on the World Wide Web, for web-based systems that show adaptive performance [25]. Researchers of the field divide web mining into three main categories based on the primary kinds of data used in the mining process [26], [27], [28]:

- Web content mining, as a process of extracting useful information from the content of web documents, concentrating on the information presented to users as texts, images, structured data, videos and so forth.
- Web structure mining, as a process of discovering structure information from the Web, dealing mainly with hyperlinks and web documents.
- Web usage mining, which mainly deals with discovery of access patterns from web usage logs.

Some authors also highlight a fourth category called user profile mining [29], while others believe it to belong to web usage mining. The major application areas of web mining are as follows: web personalization, system improvement, website modification (attractiveness in terms of content and structure), business intelligence (discovery of marketing trends), and usage characterization (users' interaction with the browser interface and navigational strategies taken) [13].

As this thesis concentrates on studying web users' behaviour, web usage mining will be examined in more detail. Web usage mining (WUM) is defined as the application of data mining techniques to discover usage patterns from web data, in order to understand and better serve the needs of web-based applications [13]. WUM is a paradigm, where data mining techniques are applied to web usage data with the aim not only to capture, model and analyze users' behaviour, construct user profiles, or develop an adaptive system, but also to support decision making processes with a better understanding of website visitors and their needs [25], [30]. Usually three main phases of WUM are applied for the task: data pre-processing, pattern discovery, and pattern analysis. WUM gives meaning to valuable data in web usage logs.

Typical data sources for WUM are web server logs, application server logs or web usage logs collected separately on an application level, and holding clickstream data representing users' navigational behaviour. This data typically needs to be cleaned from unwanted records, e.g., access references to embedded objects, or page requests made by web crawlers. Data cleaning is typically followed by user identification and dividing requests into sessions. These are the typical steps of data pre-processing in WUM. These steps are automatically covered by the special log system introduced in this thesis in Chapter 3.

The simplest form of WUM is simple statistical analysis. However, more elaborated data mining techniques are also applied for web usage mining, involving classification, clustering, association rule mining, and sequential pattern discovery [22], [27], [31]:

- Classification (also called inductive learning) is a process of assigning data items to one or several predefined classes. Classification is usually based on decision trees, naïve Bayesian classifiers, or neural networks. A typical application field is detection of spam emails.
- Clustering is used to organize items with similar characteristics into groups, whose members are alike. Web page clustering identifies pages that seem to be conceptually related based on their access by users. Clustering web users will disclose user groups with similar interests and needs based on their navigational behaviour. Typical representative of clustering is the K-means algorithm.
- Association rule mining is a fundamental data mining task for discovering regularities in data, such as frequent patterns, associations and correlations among sets of items. The objective is to find all co-occurrence relationships, called associations, regardless of the sequence they occurred in. For that sequential pattern mining is applied.

Association rule mining not only helps to discover relationships between pages, even if those pages are not directly connected, but also users' access trends for predicting their future actions based on visit patterns. Typical algorithms that are usually applied are Apriori and FP-Growth.

The main applications of WUM are general web usage statistics, overall users' characterization, discovery of user navigation patterns, user behaviour prediction, improvement of website information, structure and kernel system, recommender systems, and web personalization [25], [30], [32], [33]. It is heavily used for e-commerce, business intelligence and marketing, but also in other areas for providing personalized information presentation. The results of web usage mining provide input for user modelling, which makes it an essential component of any intelligent web information system targeting adaptive web and features of personalization for enhanced user experience.

2.3 Adaptive Web

One of the application areas of web usage mining is providing users with enhanced web experience through adaptive websites. Perkowitz and Etzioni define adaptive websites as sites that automatically improve their organization and presentation by learning from visitor access patterns [32]. For example, when a user navigates from one page to another, the adaptive system can detect interests of that user and manipulate the set of links or content presented to user accordingly. Adaptive websites employ visitors' past behaviour to approximate to users' needs. Web adaptation can be either tactical, involving a refined site topology, or strategic as general site improvement.

Strategic adaptations, or also known as long-term adaptations, are those that introduce improvements to current WIS. Typically these are made as a result of an in-depth analysis of website usage and affect all site users. For example, web usage mining has identified sets of pages that are frequently used together. Comparing existing website structure to these newly discovered relationships, improvements are introduced to site's topology as new links between pages to improve content availability for visitors. In Chapter 4 metrics for identifying possible areas for strategic adaptation and web improvement are discussed.

Tactical adaptations (i.e. short-term) on the other hand are triggered in real-time as a result of web recommendation or personalization. They do not modify original website structure, and are applied for a particular user without affecting other users' web experience. For instance, these adaptations can be in the form of providing references to other pages users might be interested in, highlighting or raising certain hyperlinks to draw more attention to them and thus make them easily located by users. Business-oriented adaptations (e.g. advertisements) fall also into this category. The objective of tactical adaptations is to deliver recommendations and web personalization. Tactical adaptations are based on monitoring user's activity pattern and automatically adjusting the interface or content provided by the system to meet users' predicted needs.

Perkowitz and Etzioni [34] have formulated a set of testimonials for developing adaptive websites. These are as follows:

1. Adaptive features should not create extra work for visitors;
2. Make the website easy to use for everyone;
3. Hold the webmasters workload at minimum while administering and authoring the site's adaptive features;
4. Leave the existing site structure intact; objects such as links and pages can be added, but not removed;
5. Keep the human webmaster in control of any automatic system.

Through strategic and tactical adaptations websites can be made more accessible and brought closer to its users' expectations, considering visitors' real needs and interests, and providing information on this basis. The latter leads the discussion to recommender systems and web personalization.

2.4 Web Personalization

The results of web usage mining are a direct input for systems providing recommendations and web personalization through the features of adaptive web. Research in the area of personalization covers disciplines such as information retrieval, machine learning, artificial intelligence, data mining and of course web usage mining [15], [35].

The main goal of web personalization is to provide users with information they want or need, without expecting them to ask for it explicitly, as defined by Mulvenna, Anand and Buchener [9]. Mobasher, Cooley, and Srivastava [20] define web personalization as any action that tailors the web experience to a particular user, or set of users. They outline it to be applicable to any web browsing activity. Berendt [25] however defines web personalisation as adaptation to individual users that are identifiable by the system, admitting that this terminology is not always used consistently in the field. As this definition suggests, users need to identify themselves in some way for personalization, like for example by providing login information. However, in [35] authors do not restrict their definition of personalization to explicitly identified users and consider personalization as customization on web environment for each user through observing users and providing preferred relevant information. On the other hand, Mobasher [30] views personalization task as a prediction problem, where the system needs to predict user's level of interest in items, specific content categories, pages or items, and rank these according to their predicted values, which are then delivered through recommendations. As can be seen, the definition of web personalization is not that explicit.

Nevertheless, personalization is not always just about recommending information or adapting web automatically; it is also seen as an action of enabling users to manually customize the web interface they are seeing on a particular website. Examples of that are the 'My Yahoo!'[1] page, Microsoft Network MSN[2], and iGoogle[3], where users have the ability to select colour schema for their page, arrange tabs of information, add, remove and rearrange apps (e.g., local weather, news, etc.) by dragging and dropping page sections according to their personal taste and interests. This is a clear example of explicit manual personalization. Still, Eirinaki and Vazirgiannis [31] recommend to distinguish these two actions, calling the latter as customization. Evidently, customization is only applicable after users have identified themselves by means of logging into a WIS and their stored preferences have been retrieved from databases.

In the context of this thesis, the definitions of web personalization provided by Mobasher, Cooley, Srivastava [20], and Shahabi, Banaei-Kashani [35] match the research interest the best. The thesis concentrates on operating around anonymous ad-hoc web users, defined as website visitors about whom no information is explicitly stored in a user profile, or in any other form in the system, neither such users need to identify themselves or to log into the system. Anonymous ad-hoc web users can be either new or returning visitors. Thereby, personalization herein and in further will be used in the context of such users and is defined as web adaptation based on generated recommendations to meet the probable needs and expectations of active online visitors based on some typical user definitions.

The online component of a web personalization system is called recommendation engine or recommender system [20], [36]. Recommender systems (RS) are used to compute a recommendation set for active user session, consisting of the objects (link, advertisements, text, images, products, and so forth) that most closely match current user profile based on learned user preferences, and thus might be of interest for that user. Mobasher, Cooley and Srivastava [20] suggest to include structural characteristics and domain knowledge about the site as an additional measure of significance in providing recommendations. They also advise to consider pages that are farther away from the current user location based on link distances for better recommendation.

Recommender systems are usually based on matching active user's activity to aggregate user profiles, doing it within a rapid-response online process. They have been already successfully used in personal web-based agents such as Letizia [17], Syskill&Webert [37], Personal Webwatcher [38], OntoSeek [39], and even implemented for personalised e-learning [40], [41], [42]. The most popular recommender system is believed to be the online shopping environment

[1] http://my.yahoo.com
[2] http://my.msn.com
[3] http://www.google.com/ig

Amazon.com[1] [31]. Lately recommender systems advantaging of information available about persons in social networks have emerged [43], [44], [45], [46], [47]. Social web services, such as Facebook[2], LinkedIn[3], Twitter[4], Orkut[5], and so on are seen as an information source to obtain user profile or providing initial data to bootstrap recommender systems. These social web services and networks usually provide access to user profiles also for third parties through different APIs provided that a user has previously consented to this.

Providing users with dynamically discovered recommendations and personalizing their web experience through it has become a popular practice especially in e-commerce and marketing, where the achievable profit is measurable. For instance, on an e-commerce site a client initializes a product purchase and then a list of similar products or products other consumers have bought is presented. Features of personalization have been adopted into web search engines for a while. For example, users in Europe or Estonia will get different results than users located in the USA while searching for certain items. Lately, personalization is making its breakthrough in large web-search engines such as Google [48]. The main financial benefits of recommenders are seen in increased engagement of visitors, customization of customer-company interaction, better return on investment (ROI) on displayed content in e-commerce applications, attracting more visits to advertising financed systems and so forth [49]. The aim of any recommender system is to assist users to find needed information in the easiest way via helping them to discover pages or page-sets they otherwise might not find during their site visit.

Different approaches for personalization exist. Researchers of the field outline three main categories [20], [30], [35]:

1. Rule-based systems, where either manually or automatically generated rules are used to tailor web content for users. In these systems web administrator may set the rules based on user demographics or other characteristics. However, rule-based systems suffer from vague and subjective user descriptions and these profiles tend to be very static in their nature, which may lead to system degradation over time. Manually set rules are very commonly used in e-commerce but not only. For instance, it is a common practice for large corporate sites to ask visitors to identify their location and based on that they are allowed access to customized content.

2. Content-based filtering is based on comparing content-features and representing information in terms of similarity. User profile is expressed through content descriptions for items a user has previously visited. In

[1] http://www.amazon.com

[2] http://www.facebook.com

[3] http://www.linkedin.com

[4] http://twitter.com

[5] http://www.orkut.com

providing personalization, items similar to those the user liked in the past are considered for recommendation. Usually items and user profiles are represented as weighted vectors, and predictions are based on vector similarities. Although content-based filtering models linked information rather well, it lacks obvious ways of exploiting information about users, and recommendations are solely based on individual user's preferences.

3. Collaborative filtering is based on modelling users' behavioural activities in the past and comparing their similarity. It is based on the assumption that users with similar behaviour have alike interests. Collaborative filtering techniques search for correlations between users in terms of their ratings assigned to items in user profile. The nearest-neighbour classification is applied to find closest match to active user. In comparison to content-based filtering, items are recommended based on user similarity. The quality of recommendations improves with the increase of available user profiles. However, collaborative filtering systems suffer from so called 'cold-start' problem – new items cannot be successfully recommended until a sufficient number of users have viewed the items. Collaborative filtering is also used in social networks to provide recommendations of friends, groups, products and so forth.

As each of these approaches has some drawbacks, therefore in practice a combination of the above methods are applied. These systems are called hybrid recommender systems. Recommender systems and thus web personalization take advantage of web mining results and use these to build some sort of user models or profiles on which they operate on.

2.5 User Profile

User profiling is an important task in developing and maintaining recommender systems. In order to successfully provide website personalization, systems need to distinguish between their users or sets of users. It is essential to learn users' browsing behaviour, heavily dependent on their own conceptual model of the subject area rather that the given structure provided by developers. For this, user profiling, also known as user modelling, is applied. The process of accumulating knowledge about users the personalization system is about to operate on, is called user profiling [31]. Without accurate knowledge about the user, any system, regardless of the used algorithms, is unable to provide correct information resources to the user. User profiling as a process is an essential component of web personalization.

A user model (profile) is an explicit representation of the properties of an individual user; it can be used to reason about the needs, preferences or future behaviour of the user [50]. User profiles provide information about users of a website, containing data either acquired from users explicitly or implicitly by tracking their activity. Examples of explicit user information are demographic data such as name, age, sex, location, interests, marital status and so on, which users usually provide by registering to a site or answering some questionnaires to fill in their explicit profile. Typically explicit information in user profiles is static (it rarely changes) whereas implicitly collected data is dynamic and changes through user's actions on website. Generally user profiles are rather application-specific and thus the information and assumptions about user preferences they withhold are miscellaneous. User profiles may contain person's history, consumer patterns, past search queries, set of interests, user context (i.e., location, device, search goals) up to name, sex and age. It is suggested to store as much as relevant data on an individual as possible, in some standardized and compact representation to facilitate comparison of profiles and their updating upon arrival of new information [51].

User profiles can be either individual or aggregated to describe a set of users. They can also be limited to a particular system or shared between several systems. Information users have made available in social networks may also be used in their profiles. For instance, partners of Facebook may be given access to public and shared information about users (name, friends lists, groups, interests, etc.) to personalize user's web experience on that partner's website.

The key issues of user profiling are: how to collect data, model a user, and manage instances of models for personalization. The utmost aim is to estimate what is the most important to a user at a given moment in time and space. The methodologies applied for user modelling vary from using weighed collections of URIs [20], vectors or histograms [51], [52], concept hierarchies [53], to clustering [54], [55], and recently applying ontologies [55], [56], [57], [58], [59], [60].

2.6 Ontologies Represent Domain Knowledge

Ontologies are an explicit specification of conceptualization, representing domain knowledge and making it possible to reuse the latter [61], [62]. They allow to capture domain knowledge in a human-understandable, yet in a machine-processable way, consisting of entities, attributes, relationships and axioms. This makes ontologies very suitable for the Semantic Web. The exploitation of ontologies allows to turn software more adaptive and intelligent, share common knowledge about structure and semantics not only between human beings but also on the machine-to-machine basis, re-use domain knowledge, and reason about it. Ontologies are an important part of any intelligent system.

There are a number of languages to describe ontologies, such as traditional ontology languages Loom and Ontolingua to name a few, and Semantic Web languages, such as OIL, DAML+OIL, RDF Schema and OWL. Out of these, OWL is of interest in this thesis.

The standard language to describe ontologies is the OWL Web Ontology language (OWL) [63] recommended by the W3C in 2004. In 2009, W3C introduced OWL 2 Web Ontology Language (OWL 2), which is the extension and revision of the OWL. OWL can capture knowledge by representing the concepts and relationships among concepts of a given domain. OWL ontologies may be categorized into three sub-languages in terms of their expressiveness: OWL-Lite supporting only classification hierarchy and simple constraints, OWL-DL providing maximum expressiveness while retaining computational completeness, and OWL-Full with maximum expressiveness but without computational guarantees. Out of the three variants, OWL-Lite is the least expressive and OWL-Full the most expressive sub-language. OWL-DL variant of the language is supported by Description Logic (DL) [64] and as such is suitable for knowledge representation as well as for automated reasoning. Consequently, OWL-DL is an advanced language that could be used in creating intelligent recommender systems. Formal semantics of OWL-DL can be used for inference of classification taxonomies and to help identify inconsistencies [64].

Ontologies consist of concepts, their hierarchical organization, relations among them and axioms that formalize the definitions and relations. The components of OWL ontologies are classes (sets containing individuals), properties (binary relations linking two individuals together) and individuals (instances). The most important relations between concepts are the *is-a* relation, which defines the class-sub-class hierarchy, and the *part-of* relation, relating an entity and its components. While developing ontologies, it is advised to include only the domain information that is needed for the application and not to try to include all the possible information about the domain. Nevertheless, ontology should contain all the possible properties of and distinctions among classes in the hierarchy [65].

Ontologies are usually composed using special software – ontology editors. There are many editor tools available, either freeware or commercial software. For instance, the Protégé ontology editor and knowledge-base framework[1], NeOn Toolkit[2] or Altova SemanticWorks[3] (commercial software). The Protégé ontology editor has been used for the research described in this thesis.

In addition to ontology editors, reasoners, also known as classifiers, are used in ontology-based data management. Reasoners allow to infer logical consequences from a set of asserted facts or axioms. They are used to check OWL ontology consistency (whether a class can have any instances or not based on its description), and to compute inferred ontology class hierarchy. Thereby, reasoners can be used already during ontology design to ensure a consistent and coherent hierarchy. The advantage of ontologies described in OWL-DL is that

[1] http://protege.stanford.edu

[2] http://neon-toolkit.org

[3] http://www.altova.com/semanticworks.html

they can be processed by a reasoner tool. As with ontology editors, there are many reasoner tools, also freeware and commercial software available. Each of these has their set of supported features. The Protégé Ontology editor allows easy integration with many reasoner tools, for example the Pellet[1], and HermIT[2] reasoners, which have also been used by the author for research herein. Other reasoners to be noted are Fact++[3], and RacerPro[4].

Presently, on a state-level, domain ontologies in OWL-DL are created for Estonian state information systems, and semantic annotations added to governmental web services, which have been made available on the X-Road platform where data exchange is handled in the form of SOAP messages, web services described in WSDL and service descriptions published in a UDDI repository. These ontologies are established according to practical methodology for development of e-government domain ontologies [66]. This process is a legislative requirement introduced in 2008. These domain ontologies are maintained in a repository belonging to the Administrative System of the State Information System (RIHA[5]), a secure web-based database and software application supporting various processes of public sector information systems. The e-government ontologies, being created, could be in the future used together with the recommendation methods described in this thesis to provide users of the Estonian State Portal[6] with highly detailed personalized information.

In recommender systems, the use of ontologies allows to alleviate the so called cold-start problem, where there is no information available or the available information is insufficient to provide recommendations in the early learning phase. To overcome the problem, an initial ontology or external ontology can be used, as suggested by Middleton and colleagues [11]. Over time, recommender systems learn users' preferences and are able to automatically detect things of similar interest.

In this thesis ontologies are used to describe concepts of WIS, web users' interests, build users' domain model, and maintain a profile for online anonymous ad-hoc web user. As a language, OWL-DL is used to enable use of automated reasoning capabilities.

[1] http://clarkparsia.com/pellet/
[2] http://hermit-reasoner.com
[3] http://owl.man.ac.uk/factplusplus/
[4] http://www.racer-systems.com/products/racerpro/
[5] https://riha.eesti.ee
[6] https://www.eesti.ee

2.7 Privacy Issues Concerned with Personalization

A system cannot provide personalization without knowing the user. The more is known about the user, the better recommendations can be established. However, this comes on the cost of user privacy.

In real life, people expect that they are treated personally, and this applies to the online world as well. Nevertheless, most users want to maintain anonymity on the WWW and are opposed to give away personal information either explicitly or implicitly. Regardless of the fear and negative attitudes, users still accept some level of privacy invasion just to be able to browse and search the Web. This can be compared to having a valuable customer card for some retail chain to get discounts. The price of the discount for client is consenting to reveal some of his/her personal data, consumption patterns, and all the basket data of purchases made using the customer card.

Conscious web users however try to disclose as little as possible of their identity-relevant information, whereas with the rapid expansion of social networking some people do not realize the threats of virtual world and reveal detailed data about themselves, their thoughts, made available via their profiles.

The cookie technology is the most used implicit way to collect user-related data. Cookies can be stored on computers and used to track users' online activities either site-wide or being shared between different websites introducing more concerns over privacy, as the data is then made available to many parties. Cookies are not the only way to track users' activity. An alternative option, for instance, is to add a special session identifier to every URL user visits on a site. Blocking cookies however may harm recommender systems and personalization in a way they are not able to solve their tasks, as necessary information is either missing or incomplete. Also, it makes the task of logging users' activities somewhat difficult and urges to find other ways to identify user session in a log or just to ignore such users. Most of today's web browsers let users to choose whether cookies are allowed and restrict third-party sites to place cookies on their computers.

Despite privacy issues, monitoring users' activities and collecting logs is the only way to analyze website operation and provide its visitors improved web experience through recommendations. General usage statistics reveal access trends, devices, top accessed pages, and so on, allowing site administrators to reason what should be improved, what is of visitors' interest and what is not. In terms of personalization and providing recommendations, the granularity of data is the key in identifying users uniquely every time, to meet their expectations and deliver personalized content, allowing users to discover new information, products, and services they could otherwise miss.

The price of personalization is allowing invasion of privacy at a reasonable level, whilst improved web experience and benefits delivered to users must prevail over lost privacy.

2.8 Related Works

The research in the fields of web improvement, adaptive web and web personalization is very diverse, falling into three major categories: web usage mining, user profiling and recommender systems. The following provides a discussion covering the latter topics in the context of this thesis.

Web server HTTP traffic logs as a source for user interaction data have been explored by many authors, where they outline the flaws, problems of data incompleteness and inabilities to identify user sessions from such data [13], [19], [20], [21]. As a solution to the problems of standard web server logs and AJAX-based applications Atterer and Schmidt [67] introduced a HTTP Proxy logging solution 'UsaProxy', which made small functional modifications to the pages and enabled to catch very detailed information up to precise mouse coordinates, clicks, key presses and scrolling, together with the exact HTML DOM tree objects involved. In addition to web server logs, specially designed browser agents [16], [17] and customized web browsers [18], [68] have been used for collecting web usage data.

Different methods of collecting data about web users have been explored in [10], [11], [12], [13], [14], [15], arguing whether explicit or implicit methods suit for the task the best or not. Implicit user modelling for personalized search was explored by Shen and colleagues [69], where they found their implicit search agent to improve search accuracy over Google. Thus involving data about the user can improve results of web search. Overall, researchers have positioned themselves in favour of implicit methods.

Miscellaneous web mining techniques have been investigated by many authors [13], [20], [25], personalization based on WUM and conceptual relationships between web documents in [25], issues of web mining for personalization with different approaches and tools in [31]. In [35] authors explored the ways of anonymous web-usage mining for personalization. Pre-processing and mining of web log data for web personalization was of interest for Baglioni and colleagues [70] in the ClickWorld project, where they built classification models for inferring user sex and interests based on web visitor navigational behaviour; unfortunately with only a small improvement over random choice.

Several methods for improving web mining techniques by using ontologies have been studied. For example Lim and Sun [71] proposed to define web pages as concept instances and apply ontology-based structure mining to derive linkage patterns among concepts from web pages. In their work they analysed user historical data, clustering previously visited pages based on content similarities in order to recommend users similar pages.

For user modelling diverse methodologies such as weighed collections of URIs [20], vectors or histograms [51], [52], concept hierarchies [53], and clustering [54], [55] have been used. Recently ontological or ontology-based user profiling approach has emerged and been applied in several studies [55], [56], [57], [58], [59], [60], [72], [73], [74], [75], [76].

Personal ontologies for navigating the web have been explored by Chaffee and Gauch [72], [73], where website characterization reference ontology was matched to personal ontology to allow users to browse the web according to their personal ontology. In this system the user had the responsibility to provide the ontology for the system, which was a major drawback. In [77] authors applied ontology-based user modelling for the e-commerce domain. Their user profile consisted of user ontology (explicit data about user), domain ontology for product catalogue and interaction ontology describing user's browsing history. They concluded that ontology technology is the best choice to model users in the context of e-commerce. In [78] a semantic user modelling based on description logic for web personalization was explored, while in [79] a method for constructing semantically enhanced user model representing user's interests from click-stream data or web logs was studied.

In [80] authors observed users' news reading behaviour and modelled their interests in a personalized news system 'YourNews' to recommend relevant articles. While exploring users' news reading behaviour they came up with an interesting finding that allowing users to change established profiles typically harms system and user performance.

User modelling has also been applied for personalizing search results. Speretta and Gauch [8] looked into ways of constructing user profiles based on user behaviour on a search site to personalize the results, gaining an improvement of 34% in the rank order of user-selected results. They also noted that users rarely, if ever, look for search results beyond the first page and users' judgments are affected by the presentation order of these results. In [58] and [81] Sieg et al. explored ways to re-rank search results by applying user context represented by ontology-based user profiles, where each profile was initially an instance of reference ontology and concepts in the profile were annotated with interest scores, and user profile modified according to user's actions. Their experiments affirmed that semantic knowledge embedded in ontology can be used to effectively tailor search results based on users' interests and preferences. In [59] re-ranking of search-engine results using ontology-based user profiling and personalized information service agent PISA was investigated. With the PISA agent an improvement of 27% was gained over search results. Issues of search results personalization and mechanisms for recommendation have also been explored in [82].

In [56] Joung, Zarki, and Jain investigated user models for context-aware systems. Despite their effort, they failed to find a rich enough model to capture various aspects of user information. However, they remained certain that any information about user will help to accurate personalized services. In [83] Olsen and Malizia called for standards and formalized infrastructures to manage the data already available and make use of it in automated personal assistants.

Godoy and Amandi [75] studied a document-clustering algorithm named WebDCC (Web Document Conceptual Clustering) that carried out unsupervised concept learning over web documents in order to acquire user profiles. By extracting semantics from web pages, their algorithm also produced intermediate results for ontology generation.

In [76] a navigation assistant that provides personalized web navigation by exploiting domain-specific ontologies was presented. In this approach, web pages were converted into concepts by referring to domain-specific ontologies, which employ a hierarchical conceptual structure. The proposed navigation assistant recommended web documents that were associated with the concept nodes in the upper-levels of the hierarchy by analyzing the current webpage and its outwardly linked pages.

An approach to reduce the effort of building user models by integrating collaborative and rule-based methods of user profiling together with sophisticated user and identity profile was introduced by Korth and Plumbaum [84]. The authors achieved a classification result of 87.1% on evaluating their model against the 'MovieLens' data sets.

Recommender systems (RS) have been exploited for miscellaneous purposes, starting with suggesting books, CDs, news, financial services, wines, and other products and services mainly in the area of e-commerce, but also web pages that a user is likely to seek [20], [85]. Personal web-based agents such as 'Letizia' [17], 'Personal Webwatcher' [38], 'OntoSeek' [39], and even personalized e-learning systems [40] make use of RS's.

Diverse recommender systems for web personalization have been proposed, based either on sequential web access patterns [86] or on an ontological approach [87], [88]. Middleton et al. [88], [89] investigated two experimental ontology-based research paper RS's 'Quickstep' and 'Foxtrot' in terms of the cold-start problem of recommender systems. Their work suggests that ontological approach helps to solve recommender systems bootstrap problem; however, a question on the sophistication level of the initial user profile remained open. In [90] a framework for a recommender system that predicts user's next page request based on their behaviour discovered from web log data was described. In this work researchers used rule-based approach to predict user's next page request. In particular, they applied association rules, frequent sequences and frequent generalized sequences for the task. In [91] researchers claimed that semantic representation of recommender knowledge, in particular by means of ontology, will become an essential part of the design of future context-aware recommenders. Representing concepts through ontologies expressed in some Semantic Web ontology language (i.e., RDFS, OWL) can provide an unambiguous definition of various concepts represented in the knowledge base of a system, allowing to enrich information when it is imprecise or incomplete, and to support interoperability and exchange of information between systems, forming a shared, formal and semantic description of the "recommendation object". Web page recommendations modelled as a Q-Learning problem using common web usage logs to train the system were explored in [92].

Tag-based collaborative filtering approach for providing recommendations was explored by Zhao and colleagues [93]. The underlying idea of their work was to find top-N nearest neighbours of an item by using the semantic similarity among social bookmarking tags. They concluded that including semantic information of tags into collaborative filtering can improve predictions of a recommender system. Wand and Kong [94] explored methods of semantically enhancing personalized RS based on collaborative filtering. In their experiments they gained a 10% improvement in prediction precision by exploiting their method. Bonino and colleagues [95] proposed to use evolutionary algorithm to predict users' next request on the web, claiming that statistical approach is not effective in all cases where real-time adaptation is required.

Content-based recommendation systems have been explored in [96], [97], rule-based systems in [98], whereas collaborative filtering based approaches requiring active participation from the user community in [99], [100], where suggestions were based on the opinions of other users of the same service. In [97] website content documents were analyzed and clustered into a hierarchical taxonomy tree, which was used to match active user's interests based on viewed content. Liu et al. [101] developed a personalized news recommendation system based on user activity in Google News[1], using content-based news recommendation combined with collaborative filtering, showing that hybrid method could improve the existing collaborative one.

Bollen and colleagues [102] investigated the choice overload problem in recommender systems. Although recommender systems are used to tackle information overload problem and filter out information relevant to a particular user, their (over)use can evoke choice overload – a condition where users are faced with difficulty of choosing from a large set of good alternatives. Even though people are attracted to a large set of choices, they usually find it easier to choose from a smaller set, and are more satisfied with choices based on a fewer options. The authors conducted experiments in a web-based movie RS in three categories: top 5 and top 20 most relevant items, and a selection of 20 items from top 100 with lower significance. The experiments showed that extending recommended item list does not increase user satisfaction with recommendations, however it increases attractiveness. Small sets of recommended items are limited in variety but easy to choose from. Bollen et al. concluded that even though a larger set may be more attractive in terms of variety, there is no necessity to overwhelm users with large recommendation sets. Based on their experiments, they also came up with a suggestion that user satisfaction with a chosen item can be increased if user can contrast it against inferior items, thus making a user to believe he/she has made a good choice.

[1] http://news.google.com

Web personalization based on navigational patterns usage mining and conceptual relationships between web documents has been explored in [103]. The authors note that combining these two methods together helps to overcome the problems of solely content-based recommendations and improve the personalization. Their experiments with blind testers confirmed users to benefit from semantic enhancement of recommendations.

In [33] Baraglia and Silvestri introduced a novel solution to implement web personalization as a single online module performing user profiling, model updating and recommendation building. In [104] an update to the recommender system ‘SUGGEST’ 3.0, which acts as an Apache web server module aiming to provide users with information about pages they may find of interest, was presented. The authors adopted LRU-based (Least Recently Used) algorithm to handle knowledge base. Their system was distinct from other systems as the typically offline component of pattern discovery was provided online, evaluating interest into pages by the order a page was visited in session, regardless of its content. User sessions were identified by cookie mechanism; however the authors noted that their recommendation mechanism could be nullified if cookies got disabled. The experiments have shown their RS to be able to generate valid suggestions and reduce the average session length.

Mobasher, Cooley, Srivastava [20] introduced a system to recommend hypertext links. Their ‘WebPersonalizer’ system applied association rules to capture relationships among URI references in users’ navigational patterns. On the discovered relationships support and confidence values were applied.

Vassiliou and colleagues [105] argued that seamless personalization is an emerging trend. Future websites will not let users know that they are being tracked and offered personalized content, providing users an impression that a particular website specializes in and offers content that they are interested in.

In [106] legal aspects and privacy issues of web personalization and recommendations were discussed. Even though in general users expect systems to understand their needs, there are several issues to address concerning privacy. While traditionally privacy issues were concerned with data processing, with the success of social networking users have become active providers of recommendation data. The community-based conception of social networking has created an illusion of “friendly” and “trustworthy” environments, where users freely share personal data about themselves and also about others. In [107] Chen and Williams discussed the need for privacy-aware recommender systems in the context of social networking. They proposed requirements architecture to address the problem, where the system interacts with users providing them choices and informing them about privacy concerns.

Users’ interest towards web pages is another issue to be considered when talking about user behaviour on the web. Hofgesang [108] outlined that the vast majority of WUM researchers usually apply a list of web pages visited and the order of these pages to express user interest, not paying any attention on time spent on pages (TSP). He argued that TSP is a well-recognized relevance and interest indicator in information retrieval (IR), human-computer interaction (HCI) and even e-learning, and should be applied as a distinct and natural indicator of importance of a webpage in WUM – the more time users spend on it, the more important it is assumed to be. Farzan and Brusilovsky [109] also investigated the effect of using the time spent on a page as an interest measure on navigational footprints. They concluded that applying TSP on footprints helps to identify important usage patterns, highlighting that users are unable to accurately assess interest towards a page in less than 5 seconds. Issues of measuring TSP either on client-side or server-side have been explored by Srivastava and colleagues [13], preferring server-side measuring because of the overhead on client side. In [110] the authors conducted an eye-tracking study on web users exploring their web viewing behaviour and concluded that users evaluate the importance of the information found on the page during the first few seconds.

Implicit interest indicators in users' behaviour have been investigated by Claypool et al. [18]. They developed a special web browser called 'The Curious Browser' through which they were able to track users' actions (mouse activities, scrolling, etc.) while users were provided an interface to explicitly evaluate web pages. They concluded that time spent on pages is a good implicit indicator of interest, while mouse movements and clicks are insufficient.

Another study on web users' behaviour by Weinreich and colleagues [111] concluded that browsing is a rapidly interactive activity, and even pages with plentiful information and many links are regularly viewed for a brief period of time only. They also noted that in their experiments nearly in 50% of cases users were spending less than 12 seconds on visited pages, leading to a conclusion that visitors browsed for a next page before reading a substantial part of the page contents.

Web usage mining is also being applied for WIS evaluation and metrics. Spiliopoulou and Pohle [112], [113] applied WUM to determine needed modifications to website design, content and link structure for improving its success. Needed modifications were detected by modelling user navigation patterns and comparing them to the existing site structure. A method for web quality evaluation and a 'WebQEM' tool to evaluate websites was proposed by Olsina and Rossi [114]. Their tool assessed websites against preset requirements.

Modelling web users' domain and website topology through domain ontologies for semantic portals has been explored in [115] and [116]. Mikroyannidis and Theodoulidis [117] introduced a framework enabling adaptation of web topology and ontology to match the needs and interests of web users, where users' access data was used together with the semantic aspects of the web. Coenen and Swinnen [118] introduced a framework for implementing self-adaptive websites. They emphasised that strategic adaptations need to be implemented strictly offline and human-managed to avoid visitors getting lost in hyperspace; while for tactical adaptations a recommender system should be used to enhance the existing site structure. Different algorithms for adaptive websites have been explored in [34].

Understanding web users and their behaviour on websites has recently become very important with the objective to provide users easy access to necessary information and help them to tackle the problem of information overload provoked by the exponential growth of the Internet.

In terms of research, the majority of the work in the area of web personalization and adaptive web deals with the problem how to process and model web usage data into user profiles to generate recommendations and through it personalize users' web experience. The efforts on user modelling carry valuable information usable for learning from users for an improved web already today. Continued success of a website is driven by new visitors who need to be attracted to the site and returning users who need to be retained to visit the website. The key to achieve the latter relies in user satisfaction.

2.9 Chapter Summary

In this chapter an overview of essential background information required to understand the main parts of this thesis has been presented. The chapter has two main parts, where in the first part ways of collecting web usage data and its further processing are described, covering the disciplines of web usage mining, adaptive web and web personalization, recommender systems, user profiling, ontologies, and related privacy issues. The second part of the chapter has been dedicated to state-of-the-art related works of the field.

The research of understanding users, their needs and expectations, as well as behaviour has gained interest of many scientists. Different approaches to address the problems of this domain have been proposed, some of them less successful than others. Regardless of it, the research work in the field has a crucial role to play in developing and maintaining today's web information systems. Still, there is no 'silver-bullet' solution to eliminate the gap between what systems are able to deduce from users' actions and what are their real needs at a certain point in time and space. This urges researchers to establish new methods for web personalization.

Chapter 3
LOGGING WEB EVENTS

Event logs carry valuable information about the status and usage of a system. Logs about web events – more specifically about users' actions – can be used to improve web information systems and minimise the risk of failure by improving the quality and user-friendliness of web based systems and provided services. The benefits gained from web event logs are however not limited to the latter. The data in the log is applicable also for modelling users' behaviour, deriving user profiles and predicting users' actions, which altogether form a basis for recommendation generation and web personalization. For these purposes various log systems, commonly web server logs or customized log systems, are used.

This chapter focuses on ways of collecting web usage data for general statistics, and also for web improvement and personalization, discussed in Chapters 4–7. A special log system developed to capture users' activity data is discussed and general statistics based on the collected data are presented to characterize the website used for the research on topics covered by the thesis.

3.1 Introduction

Millions of users interact daily with web information systems around the world, producing massive amounts of data about this communication. With every click users make on the Web they leave behind a trace of actions, in particular navigational paths of visited pages. Using fully automatic log systems, this valuable information can be captured and used to getting to know the users, understand their needs and behaviour. This data can be collected either site-wide where its applicability is restricted to one WIS or cross-sites, where the data gains more power, especially if users can be further identified (to some extent). Herein, the discussion is limited to on accumulating knowledge about users site-wide.

As previously discussed in Chapter 2, there are several ways to collect data about users. In this chapter, logging of web events in WIS as an implicit way to collect users' activity data is explored. In particular, a special log system will be introduced, which has been exploited to collect data for studies discussed in the thesis. The log system was initially developed only for attaining general website statistics and consequently this will be the first interest of discussion. Through the latter a characterisation of the research environment is also given.

So, why is logging of web events so important? Firstly, of course everybody is curious about who visits their websites, i.e., who the audience is, where they come from, and what pages they visit. Thus, an interest into general statistics either derived by curiosity, marketing and advertising needs, or the necessity to assess usability and effectiveness of design choices, is the main driver.

Secondly, for obvious reasons, it is impossible to develop a web system that would satisfy everybody's needs, as users have different interests, habits and cognition of subject domains. Generally, sites are developed according to the domain model established by web development team. However, this domain model may differ from general conceptual model applied by users while employing the system. Despite the latter is undetectable beforehand, it can be post-modelled based on logged users' activities and thus the gap between these two conceptual understandings could be reduced. Figure 1 illustrates the differences between the domain model employed by developers and actual website users based on collective intelligence. In an ideal situation, sets *D* and *U* are equal. The common understanding is the intersection of sets $D \cap U$ and the knowledge to be learned from users is the relative complement $U \setminus D$ of *D* with respect to *U*.

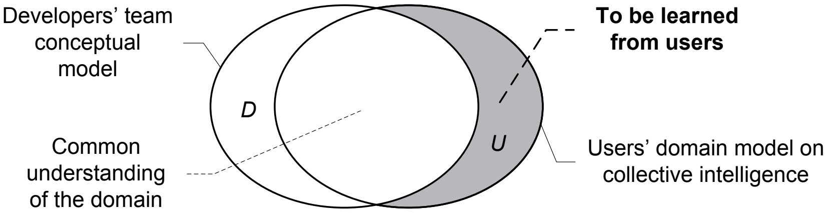

Developers' team conceptual model
To be learned from users
D
U
Common understanding of the domain
Users' domain model on collective intelligence

Figure 3-1. *Differences between the domain model employed by developers' team and actual website users based on collective intelligence.*

In order to adapt websites to its users' needs, it is essential to collect data about users' actions within identifiable sessions and analyze it in further to learn their behaviour. This knowledge is to be exploited in a smart way to eliminate the aforementioned gap and optimize websites according to its users' probable interests. Though, it should be noted that through implicit ways it is only possible to capture users' actions made at a certain moment in time and space and not the real intentions in their minds provoking those actions.

Studies based upon web users' activity deliver valuable information for system developers, marketing personnel and other stakeholders through the data collected and processed. On one hand, web mining can be used for sites' usability, navigation and efficiency analyses as general statistics, usually provided by means of page access summary, visitor overview, page views, IP hosts, unique visitors, unique sessions, new and returning visitors, referrers, search keywords, search engines, entry and exit pages, time spent on pages, operating systems and browsers used, screen resolution, JavaScript capabilities, cookies, demographics such as visitor city, country, language, and visitor paths in human-readable form. On the other hand, the collected data helps to comprehend web visitors' behaviour on a site and thereby uncover what is actually going on within a system. In addition to general statistics, deeper analyses can reveal usability problems such as fuzzy navigational structure, long page-load times, and so on, for system improvement. Thereby, the data in the log contains valuable knowledge not only for computing general statistics but also for web improvement and personalization through users' behaviour.

Today, it is common to have some sort of an analytical tool capturing web events in the background of a website to provide site owners some statistics about their site, how it is used and who the users are. In terms of site usage analyses, the most commonly exploited approach is to analyse either web server logs or to rely on third-party hosted log collectors and analytical tools. There exists a number of tools for processing web server logs, both commercial (e.g., WebLog Expert, AlterWind Log Analyzer, Wusage), and open-source and freeware (e.g., The Webalizer, AWStats). The main drawback of web server logs is that they suffer from insufficiencies and are not usually kept for a long period of time because of their large size [13], [19], [20], [21].

With the rapid development of web technologies and the boom of social web over the past years, hosted web analytics tools have gained their popularity. These solutions collect their data through embedded scripts and mostly rely on cookies and JavaScript or Flash technology. Probably the most popular in this category is the Google Analytics tool, accompanied with a variety of tools alike – to name a few more GoStats, WebTrends, Crazy Egg Web Analytics, Site Meter, W3Counter, OneStat, 3DStats and so forth. The market on these tools is very diverse from free tools to web analytics solutions with different rate plans. It has become an industry of itself.

The remainder of this chapter is organized as follows. Section 3.2 discusses capturing of web events; Section 3.3 provides an overview of the research environment. In Section 3.4 a discussion on the developed log system used to capture web usage data in the research environment is provided together with a description of the log analyzer tool. Section 3.5 presents an overview of different types of possible analyses based on the collected data. Section 3.6 provides a short discussion on general web analyses, and Section 3.7 summarizes the chapter.

3.2 Collecting Web Events

The key issues in accumulating web usage data are how to collect data precise enough, and how to identify user sessions to reason about system usage, users' needs and behaviour. It has been already mentioned that this data can be collected with various software agents, customized web browsers, from proxy servers, web server logs, or using customized log systems, which utilize session based IDs and cookies.

3.2.1 Web Server Logs

Originally, web server access logs were designed to trace server breakdowns, and thereby to provide a limited source of client information. They are set to chronicle all the operations on server, and not to produce log data for further analyses and reasoning about the data. Amongst other standards, the World Wide Web Consortium (W3C) maintains a standard format[1] for web server log files; however, other formats also exist. Typically the log is presented in the common log format[2] (CLF) supported by the majority of analysis tools. With the common log file format the following data is stored:

1. Remote hostname,
2. Remote log name of the user,
3. Username as which the user has authenticated him/herself,
4. Date and time of the request,
5. The request line exactly as it came from the client,
6. The HTTP status code returned to the client,
7. The content-length of the document transferred.

Combined Log Format, another commonly used log format, adds to the list two more fields:

8. The "Referrer", that is the HTTP request header, and
9. The User-Agent HTTP request header.

Typically, web servers allow to modify what parameters are logged, for instance the popular Apache HTTP server[3] enables to modify the log record format, dismiss or add items to be logged. Figure 3-2 presents an excerpt from the Apache web server log represented in common log format (CLF).

```
193.40.246.52 - - [18/Nov/2011:16:45:27 +0200] "GET /img/banner2009.gif HTTP/1.1" 200 109254
193.40.246.52 - - [18/Nov/2011:16:45:27 +0200] "GET /img/ati.ico HTTP/1.1" 200 1150
193.40.246.52 - - [18/Nov/2011:16:45:30 +0200] "GET /index.php?page=100 HTTP/1.1" 200 11002
193.40.246.52 - - [18/Nov/2011:16:45:31 +0200] "GET /img/ipic_1.jpg HTTP/1.1" 200 15996
193.40.246.52 - - [18/Nov/2011:16:45:34 +0200] "GET /index.php?page=500 HTTP/1.1" 200 11541
67.195.115.43 - - [18/Nov/2011:16:45:55 +0200] "GET /print.php?page=6305&p=2395 HTTP/1.0" 200 2680
193.40.246.52 - - [18/Nov/2011:16:48:13 +0200] "GET /index.php?page=100 HTTP/1.1" 200 11002
193.40.246.52 - - [18/Nov/2011:16:48:14 +0200] "GET /img/ipic_3.jpg HTTP/1.1" 200 24642
67.195.115.43 - - [18/Nov/2011:16:48:17 +0200] "GET /autorideklaratsioon.pdf HTTP/1.0" 200 8499
```

Figure 3-2. *Common Log Format. An excerpt from Apache Web Server Log for the DCE website, using the log format "%h %l %u %t \"%r\" %>s %b", where %h stands for remote host, %l for remote logname, %u for remote username, %t for time when the request was received by server, %r for first line of request, %s for request status, and %b for size of response in bytes, excluding HTTP headers.*

There are many issues, why web server logs were not suitable for the research discussed in this thesis. Firstly, even with a moderately busy web server the quantity of information stored in the log files is very large (access log

[1] http://www.w3.org/TR/WD-logfile.html

[2] http://www.w3.org/Daemon/User/Config/Logging.html#common_logfile_format

[3] http://httpd.apache.org

file typically grows 1 MB or more per 10 000 requests), introducing a need to periodically rotate the log files by moving or deleting the existing logs. For this, web server needs to be restarted. For example, until version 2.1 Apache HTTP servers were unable to handle log files larger than 2GB. In addition, web server log files are usually not accessible to general users – thus administrative rights and special access is needed to attain these logs.

Secondly, the aim of web server logs is to chronicle all the operations on that server and not to produce log data for particular analyses. As users' behaviour is presented through their actions (clicks) on a site, access to every single object presented on a web page (e.g. dots and lines as elements of graphic design) is not of interest of study usually. For instance, note the rows 2, 4, and 8 on Figure 3-2, which describe access to graphical elements as a part of website design.

Thirdly, it has been proven in [19] that HTTP traffic logs appear to be flawed. Many authors outline that when dealing with server-side data collection from web server or some other general system alike, which is not specially designed for capturing log from web visits, there are some major difficulties due to data incompleteness, and issues of data caching either through proxy servers or on client level [19], [20], [119]. Requests from proxy server may have the same identifier in web server logs, even if the requests were made by different users. Also, a single request from the server could actually be served to multiple users. This leads to a need of excessive pre-processing consisting of data cleaning, user, session and page-view identification, and path completion.

Fourthly and mainly, to be able to develop user models, one needs to be able to identify user sessions. Web server logs however lack of means to properly identify user sessions [21], [28], [119]. Although they contain client IP-addresses, these cannot be reliably used as Internet Service Providers (ISP) tend to have a pool of proxy servers, and a user might be assigned a different IP for every request, which leads a user session to have more than one address. This could be overcome with getting the users to identify themselves but it is a common truth that users prefer to maintain strict anonymity on the web and are against of any tracking of their web visits and other characteristics. Embedded session identification means and cookies could reveal the problem.

Yet, there are a few advantages of web server log files as well. Firstly, despite the weaknesses, web server logs carry valuable information for error analyses. Secondly, access data is available without the need to introduce any modifications to website. Thirdly, the data is on the hosting server and available for site owners anytime. Fourthly, there are no scripts needed for action logging, which could slow down page loading, and finally, web server logs do not depend on the capabilities of a visitor computer.

Altogether, considering the pros and cons, standard web server logs still occur not to be suitable for the task because of their flaws and inability to capture user sessions and to distinguish between new and returning visitors.

3.2.2 Hosted and Third-Party Analytics Solutions

As mentioned earlier, there exists a variety of third-party hosted analytics solutions. Some of them provide statistics in real-time, others with a delay. They are popular mainly because the small effort needed to set up them and the simplicity in getting ready-made statistical reports. From the customers' point of view, they are a great way to get general statistics at ease – all that is needed for an installation is just to include a piece of script on your page. The statistics are made available for customers in an online environment.

Despite the easiness of their usage, these tools also suffer from drawbacks. Typically, hosted analytics solutions return only general statistics and it is impossible to obtain the raw access data, even if some sort of a solution for data export is provided. This means that the original access data is traded for the statistics and is "lost" for the customer of hosted analytics tool. The results provided by these tools in the form of general statistics are however insufficient for web improvement and personalization. For example, out of the many statistical metrics provided in the Google Analytics tool, time spent on page is the only value usable in studies discussed here in further. On the other hand hosted analytics tools gain the power of globally profiling users as their interests and habits over several websites can be captured.

From the technological point of view, hosted analytics tools mainly exploit cookies and scripting, e.g., JavaScript or embedded Flash objects. This approach has its own advantages and disadvantages. The main advantage is that requests are counted on the basis of page opening, thus it counts cached pages as well (in opposite to web server logs). Also, scripts are run on client side and through that an additional access to information not available to server-side applications can be gained, for instance visitors' screen size. Furthermore, actions such as mouse movement, location and clicks can be captured. Hosted analytics is the most suitable form of tools to get statistics if there is no access to web server logs or to system kernel.

On the downside, users can turn off scripts and disable cookies, which results in no data or inaccurate data to be collected. Also, a fact that the collected data is not in possession of the site owner and it is not possible to get the raw data, if it would turn up to be necessary, should be considered as a drawback. Site owners also have to rely on and trust the results provided by hosted analytics.

Thus, if one is interested in raw access data, this can be attained either from web server logs or applying a special log system for the purpose. Reading and processing web server log files for web visits is sophisticated and time-consuming, therefore many prefer hosted script-based web statistics to web server log based statistics on the cost of losing their data and depending upon the analytics service provider. Altogether, third-party tools are unsuitable to learn from users for improving and personalizing web, although they have proven to be useful in providing general statistics over a wide range of metrics.

3.2.3 Other Issues to Consider with Web Events Logging

If not relying on web server logs as a source of visitor activity data, then there are several other issues to be concerned about. First of all, distinguishing different visitor sessions, as one of the shortcomings of web server logs, is of interest.

The most applied method for identifying visitor sessions and tracking web users is exploiting browser cookies. Storing cookies on visitor computer is transparent to users and requires no effort from them. However, cookies are computer and browser-based, so if a visitor uses several computers or has different browsers on the same computer, the cookie mechanism fails to identify the user uniquely along several systems. Nevertheless, cookies help to distinguish different users behind the same IP address or users who come back with a new IP address every time. Also, the cookie-mechanism cannot distinguish between multiple users who use the same computer and browser under the same account. The most troublesome aspect with cookies is that users can turn them off. Typically web browsers give users a choice to accept all cookies, accept cookies only from the site they visit, thus disabling third-party cookies on which hosted web analytics tools rely on, block cookies at all, or define cookies to be allowed or blocked upon their personal preferences. Also, users have the choice to set the time cookies are stored on their computer, either till the end of cookie lifetime, or until the browser is closed. If multiple users are using the same computer and the same browser, there is a danger that user profiles become a mixture of various and possibly conflicting interests of these users. Cookies as a tracking mechanism can be used by server-side and client-side web event logging applications.

As an alternative to cookies, lately W3C has introduced a web storage[1] standard (initially a part of the HTML5 standard) defining an API to enable websites to store more data locally within the user's browser than it has been possible with cookies (a typical cookie size limit is 4 KB). The DOM storage is divided into session data with lifetime limited to open browser session, and local data with no set expiration, thus making the latter data available whenever in the future. All modern web browsers now support this feature.

A similar approach to cookies is to include the session ID in the URL or hide it in web form as hidden field submitted with every page request. However, with session IDs in the URL there lies the danger that users may link or send the address with this session ID to other users and thereby enable them to take over the session. This approach suffers from the same drawbacks of identifying the same user on different computers or using various browsers as does the cookie-based user identification.

Eventually, the identification of users is up to either cookies, IP-addresses and other access-related data or a combination of these. Thereby, the definition

[1] http://www.w3.org/TR/webstorage/

of unique visitors, heavily used for marketing and in advertising, is dependent on the identification approach. There is no solid way of user detection without enforcing users to log into a web information system.

Another issue is concerned with web crawlers, also known as robots or bots or spiders, which are automated programs that systematically fetch information from websites. Web crawlers are mainly used by search engines to index web content; however they are also used for harvesting specific information, such as e-mail addresses. Web server logs contain also all the accesses made by robots where a single robot session may result in hundreds of clicks that were not actually performed by real users. At the same time the frequency robots visit a page also shows online presence and search engine optimization (SEO) friendliness; the higher the rate is the better it is. It is also worth to know which search engine bots find a way to website being studied and what pages they crawl in it. Typically robots do not execute JavaScript on web pages, so there is less worries about bots with script-based logging systems. Blocking of search robots is not a solution to the problem as it incorporates the risk of loosing potential visitors as most of the robots gather and index website content for free. Thus, keeping robots away introduces the risk of not being referred to in search results. Nevertheless, before data in web server logs can be used for general statistics or user studies, web improvement and personalization, robot accesses must be identified and removed as they do not reflect any kind of user activity.

Luckily, most of the crawlers follow the good practice and identify themselves via HTTP agent string. For example Google search robot uses a name ‘GoogleBot’, whilst Microsoft Bing search crawler identifies itself as ‘bingbot’. Most of web crawlers follow the robots exclusion standard (a file called robots.txt on domain level), which webmasters can set up to give instructions for robots before they begin indexing the site. An alternative is to use the HTML <META> tag on a web page to give instructions for crawlers. Still, crawlers may follow these requests prior indexing or just ignore them.

3.3 Overview of the Main Research Environment

Before continuing the discussion with web events logging, it is necessary to consider the main research environment, where the data collection has been implemented and data collected for the studies discussed in the thesis in forthcoming chapters.

The research presented in this thesis is mainly based on the access log data collected on the website of the Department of Computer Engineering[1] (DCE) at Tallinn University of Technology (TUT). The access rate for the DCE website is in average 80 visitors per day depending on the academic season. To obtain better coverage of accesses, the DCE website sends out HTTP headers

[1] http://ati.ttu.ee

instructing proxy and client level caches that the documents should never be cached. Identical approach has been taken also with other websites, where the log system is in use. Presently, the system withholds descriptions for 114 pages in the menu structure actively in use or been recently in use, in three main categories (General information, Research, Studies) in two languages (Estonian and English). The main language for the DCE website is Estonian. The aim of the website is to provide information about the department in general (organization, history, staff members, graduates, contacts, etc.), deliver accurate information about research (projects, main results, publications, etc.), and provide students with necessary information related to their studies (subjects lectured at the department, related curricula, graduation information). The available information is organized on three menu levels from general to more specific. In addition to providing information to general public, the DCE website also serves as an experimental environment for research and development.

The DCE web information system was chosen for the research as it is the sole creation of the thesis author, and thereby the author has gained an exhaustive knowledge about the system and its behaviour. This knowledge and unrestricted access to the system has greatly contributed to the research.

The kernel of the DCE dynamic website is built on widely used general-purpose server-side HTML embedded scripting language PHP (Hypertext Preprocessor) and on a popular relational database management system MySQL. It handles every request made towards the DCE WIS. Currently the system runs on Apache 2.2.9 web server with PHP 5.2.6 and MySQL 5.1.

The web kernel (Figure 3-3) was developed by the author of this thesis at the beginning of year 2000 and has been developed in further over the years. It is in active use for several web-based systems in the department as well as faculty-wide. Also, the website of the Faculty of Information Technology[1] (until March 2010, when a new CMS system was enforced by TUT) was based on this kernel. Other systems that make use of it are the department's intranet application ITA[2], e-learning environment e-EDU[3] [120], Career services of TUT[4], competence centres CEBE[5] and CREDES[6], EU FP7 Project Diamond[7], and many others. In most of these environments the log system is also present and data is captured. For the e-EDU system, an extended version of the log system has been established to better suit the needs of the environment. Figure 3-3 provides the general overview of the web system kernel and its modules.

[1] http://www.ttu.ee/itt/ (formerly), http://deepthought.ttu.ee/itt/ (as of March 2010)

[2] https://ita.pld.ttu.ee

[3] https://edu.pld.ttu.ee

[4] http://deepzone0.ttu.ee/career/

[5] http://cebe.ttu.ee

[6] http://credes.ttu.ee

[7] http://fp7-diamond.eu

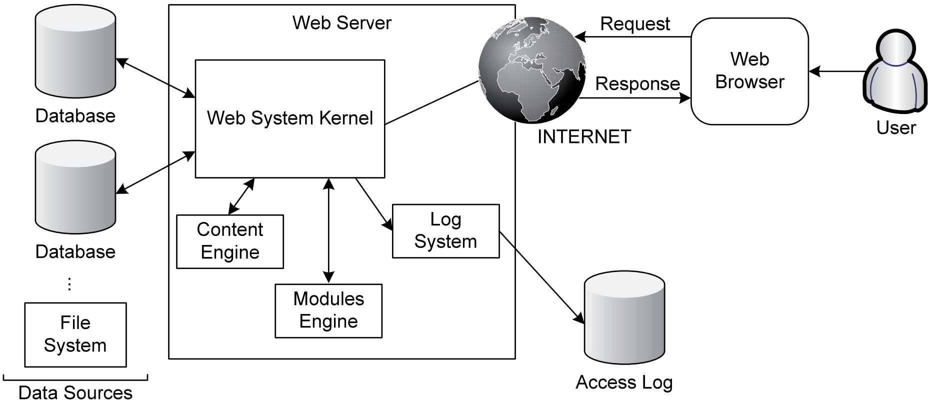


Figure 3-3. *General overview of the web system kernel and its modules.*

3.4 Developed Special Log System and Log Analyzer

After analysing different possibilities of collecting usage data and taking into account the aforementioned drawbacks, it was evident that a special log system was in demand to easily satisfy the informational needs for general statistics and for users' behavioural studies, and to gain more accurate records of data than the use of web server log files would enable. Moreover, at the time these studies were initiated, the situation on collecting and analyzing web events was totally different from the situation today, where a variety of hosted tools exist. Web server logs were considered as the main source for such data; however, as proven they appeared to be flawed. And neither have hosted solutions provided means necessary for studies discussed in the thesis. The decision was either to base the studies on web server logs and somehow tackle the problems concerned with those, or to develop a special log system for capturing users' activity on websites. The decision was made in favour of the latter and as a result a log system as an additional module to the web kernel was developed at the DCE. The log system allows to address the problems of web server logs and at the same time to benefit from the advantages of these logs. For processing the collected raw web usage data, a log analyzer system was established.

Usage log data for the DCE WIS has been collected since February 2002 and with an improved log system since November 2003. As research and various experiments discussed in thesis have been carried out at different times, characterization of log data will be presented with a discussion of a specific research topic.

3.4.1 Log System Architecture and Implementation

The log system was firstly introduced in 2002 and elaborated as an extension to the web systems kernel developed at the DCE. Initially, it allowed to store only some basic properties of actions users performed while employing the DCE website. At that point the system was only aimed on general usage statistics. Preliminary data analyses revealed that such a system could not be efficient enough; moreover it was unable to answer the many newly raised questions about the exploitation of web systems and users' behaviour. Therefore, many new features were added; for instance session capturing and logging of recurrent visits, which was a major improvement. The re-designed log system was launched in 2003, with some minor improvements in 2004 to adjust its flexibility even more. These modifications presently enable to capture data for many necessary studies and experiments discussed in this and the following chapters. Today, the log system allows accumulating of the following data:

1. Page requested by visitor;
2. Timestamp for page request;
3. IP and host of domain, where the request was made from;
4. Browser and operating system used to view the page (based on HTTP User Agent string);
5. Query method and full query string;
6. Site referrer, if present;
7. Visitor identifier (session based ID);
8. Number of operations performed during a session;
9. Time to load a page with a reference to server load during the page composition,
10. Previous visit identifier (session based ID) and time (if available);
11. Screen resolution used for viewing the requested page (client side detection using JavaScript).

As can be seen, the log system is designed to save data proposed by common log format and available in web server logs and even more. However, the system records only page requests, not every single request to individual objects (design elements) presented on accessed web pages. Also the system identifies user sessions, which is the main distinction from web server logs. Each website using the log system is advised to include mechanisms to output HTTP headers instructing proxy servers and client browser caches that the served content should not be cached. This is essential to collect precise access data.

The log system is initialised every time a page request is made (Figure 3-4). With the first page request, a new session is opened and a previous session cookie is searched for. In case of a match, a reference to a prior visit is made. This is followed by placing a new cookie to the user's browser for current session tracking and updating the recurrent visit cookie. The lifetime of current session cookie is set to 60 minutes since the last request. Thereby, user session expires in an hour of no activity or after closing the browser. The lifetime for the recurrent visit cookie is set to 1 year. With every request made to the web system kernel, the log system is initialized and data about the request captured and stored. The log system finishes after the full loading of the requested page, storing the data it captured into the log database. The actions carried out by the log system in order to store the actions, are described on Figure 3-4.

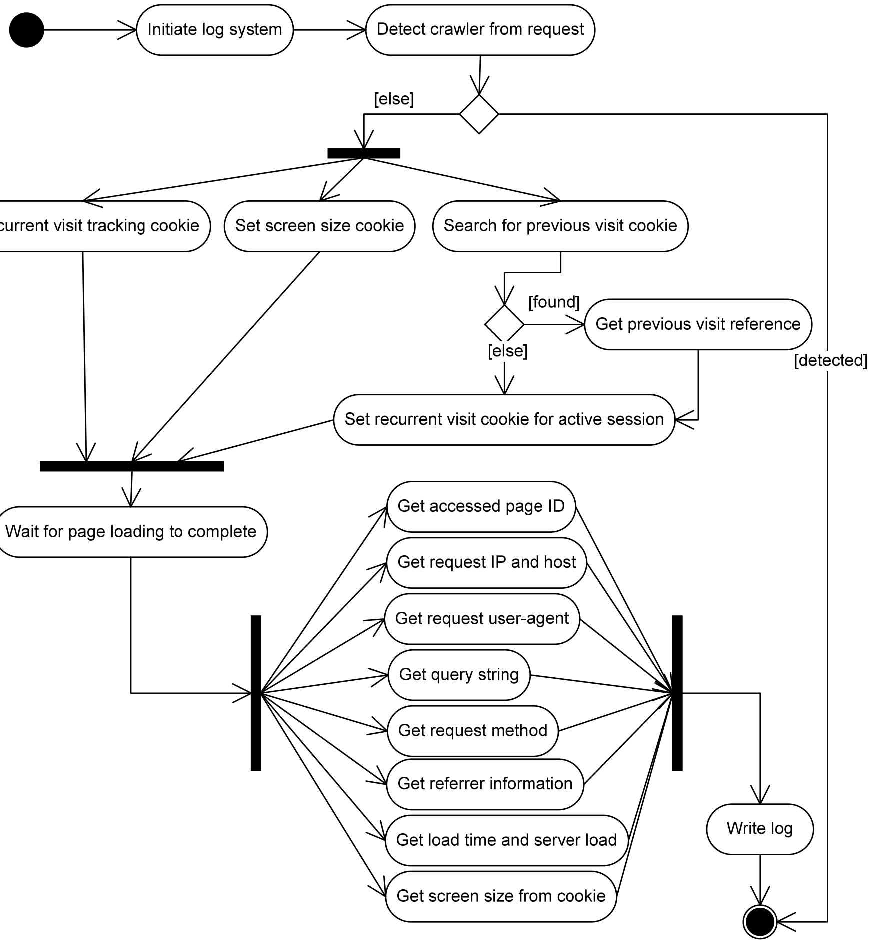

Initiate log system
Detect crawler from request
[else]
[detected]
Set current visit tracking cookie
Set screen size cookie
Search for previous visit cookie
[found]
Get previous visit reference
[else]
Set recurrent visit cookie for active session
Wait for page loading to complete
Get accessed page ID
Get request IP and host
Get request user-agent
Get query string
Get request method
Get referrer information
Get load time and server load
Get screen size from cookie
Write log

Figure 3-4. *Actions carried out for web usage data capturing by the log system.*

To deal with web robots, detection rules were added to the log system. These rules, based on [121], information available on the Internet and personal expertise, enable to skip logging of accesses made by known spiders and keep the log size smaller. In sense of general usage statistics and users' behavioural studies, web robot accesses produce just noise records and can be left out of the log. Despite the capabilities of the log system to identify known robots, presently the system has also logged the crawlers, as they provide feedback on website online presence and SEO friendliness.

Yet, there are a few drawbacks to be considered with the log system. In case a user has disabled or deleted cookies, the normal operation of the log system is compromised. If the cookie used to detect recurrent visits is missing, the request is classified as made by a new user. Even the use of IP address would not solve the problem as there might be several users behind the same IP address, for example in the case of using the same shared computer. If cookies are disabled, the system is still able to produce the log; however, immediate identification of user session is impossible. User sessions could then be identified by the log analyzer based on the requesting host IP, user's browser and operating system, and access time. Also, if a user accesses the website from different web browsers on the same computer, these are classified as separate sessions, as each browser stores cookies separately. For that, there exists no workaround. These are the limitations of the log system. The most reliable solution to detect and distinguish different visitor sessions would be to have users to log on to the system, which however is unrealistic for general purpose websites and portals, because anonymity and privacy is what the Internet is based on and what users of the World Wide Web value the most on.

The log system is detached from the analyzer IS (Figure 3-5), storing all the performed actions in a table (comparable to common web log file) with several attributes. By performing only one INSERT-command per request, the log system, which runs on MySQL using the MyISAM storage engine, performs its writing instructions speedily not causing delays in the web system itself and disturbing thereby users. Figure 3-6 shows an excerpt from the log table. Prior to data mining, logged data is transferred into a log analyzer IS, which is based on snowflake schema. The analyzer system uses InnoDB storage engine, which allows transactions and data processing to be executed on the database level.

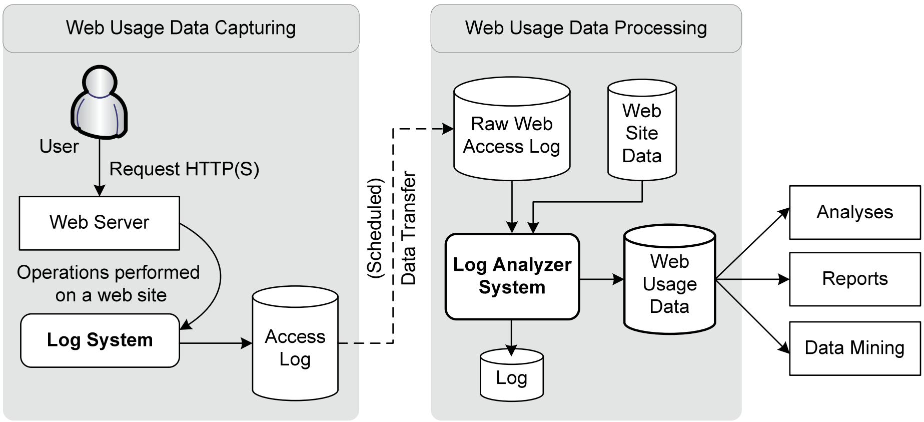


***Figure 3-5**. Data collection and pre-processing using the log system and analyzer.*

An alternative approach to using database for storing captured log data would have been to use a structured file or a text file. On the other hand, log records stored in database can be exported from the log system as a text file in a well-known format, i.e. in CLF, to be used with web server log analysis tools.

The current log system module works in the background and in cooperation with system kernel and besides cookies set in users' browsers the approach is totally transparent and requires no active participation from users. In the future, the log system will be further developed and features such as event logging over web service, hosted event logging and DOM storage will be introduced to capture users' actions cross domains from several websites without the need to integrate the module into a web system kernel.

The log system has proven to be a convenient way to get raw usage data for various studies, discussed here in the thesis and beyond.

Page ID	IP	User Agent	Access time	Referrer	Session ID	Load Time	Server load	Screen size.
100	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:13:21		ATI09.01.11.23.13.2180	0.14	1.05	1024x768
200	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:14:39	http://ati.ttu.ee/	ATI09.01.11.23.13.2180	0.05	1.10	1024x768
210	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:14:43	http://ati.ttu.ee/index.php?page=200	ATI09.01.11.23.13.2180	0.06	1.09	1024x768
210	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:14:47	http://ati.ttu.ee/index.php?page=200	ATI09.01.11.23.13.2180	0.04	1.09	1024x768
2103	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:14:49	http://ati.ttu.ee/index.php?page=210	ATI09.01.11.23.13.2180	0.13	0.98	1024x768
220	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:14:55	http://ati.ttu.ee/index.php?page=200	ATI09.01.11.23.13.2180	0.05	0.98	1024x768
2203	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:14:57	http://ati.ttu.ee/index.php?page=220	ATI09.01.11.23.13.2180	0.05	1.07	1024x768
230	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:16:48	http://ati.ttu.ee/index.php?page=200	ATI09.01.11.23.13.2180	0.24	0.82	1024x768
2303	85.117.111.26	Opera/8.51 (NT 5.1)	2011-01-09 23:16:51	http://ati.ttu.ee/index.php?page=230	ATI09.01.11.23.13.2180	0.11	0.80	1024x768

Figure 3-6. *Excerpt from the captured web log data. Shown are attributes for accessed page ID, user IP, browser and operating system, access time, referrer, unique session identificator, page load time, server load characteristics at the access moment and client screen size. Attributes not shown on the figure: log row number, host, query method and string, previous visit session ID and time.*

3.4.2 Log Analyzer System

While raw web usage data is collected by the log system, the task of the log analyzer is to process this data, and transfer it onto a more sophisticated data model for data archiving and mining (Figure 3-5). Web server log files could be used as an input for the log analyzer system as well; however, pre-processing needs to be applied beforehand. As a result of this data transformation, various complex queries can be run on the collected data to analyze website's usage and perform various data mining tasks for web improvement and personalization.

Currently the log analyzer is implemented as a prototype tool, consisting of stored MySQL routines (procedures and functions) using InnoDB database storage engine as the base engine. The analyzer system consists of 18 tables, 1 dynamically generated view, and 26 routines. The log analyzer data model and description of routines are presented in Appendix A. The log analyzer system deals with three types of tables:

1. Tables representing the log data (9 tables),
2. Tables holding classifiers for describing the log data (6 tables),
3. Tables necessary for the operation of the log analyzer system (3 tables).

The log analyzer is initialized with a call to a routine 'parse_log_entries'. Prior to that, the first and the last row ID to be processed from the web usage log are set. The log analyzer remembers the last processed log entry from the log system. Data processing is customizable via system settings enabling to:

- Select the table for input data,
- Set the number of entries to parse from the log,
- Set default values for entries with unknown classifiers,
- Set the session ID format to be processed from the log,
- Configure logging for parsing at four different levels: errors, warnings, info, and debug.

During the process of data loading from the log system into the log analyzer database, access data is processed row-by-row for each session and calculations are carried out to add value for the captured log data. For instance, session length and number of operations in a session are found. Session data is also compared to available classifiers of interest, such as browsers and operating systems, and necessary attributes added to describe each session. The process also includes detection of web robots based on the available detection rules and descriptions. Crawlers harvesting information on a website produce hundreds or thousands of unwanted "artificial" clicks, which would interfere web usage mining if not identified and removed. Still, as the system identifies crawlers by the user-agent string bots reveal about themselves, it might happen that a new robot or the one with an updated user-agent string is not automatically recognised. These sessions are tagged and need intervention of a webmaster to update the detection rules.

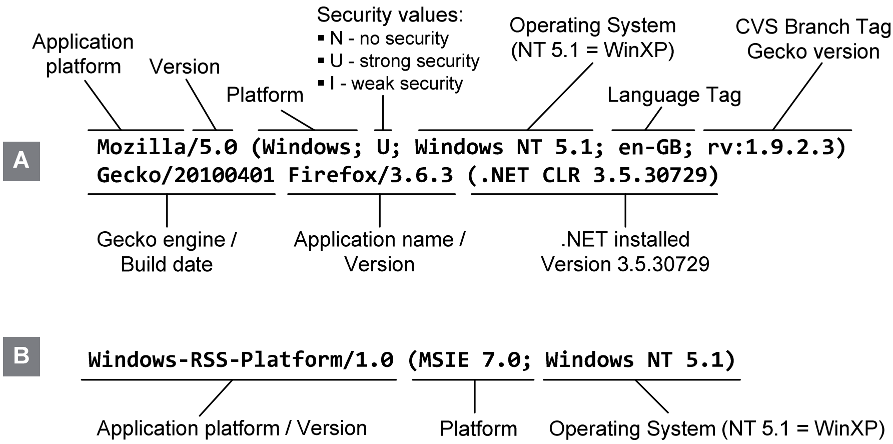
Typically the access log already contains identified user sessions. However, if a user has disabled cookies, user sessions remain unidentified. Presently, the log analyzer does not deal with the situation. Nevertheless, the upgrade plan involves a set of methods to address this issue. To overcome the problem the log analyzer will process those log entries and tries to reconstruct user sessions based on the host IP, HTTP user-agent string and access time. This approach cannot guarantee that requests coming from the same IP, with same user-agent string in a relatively close timeframe belong to the same user; yet, the probability of these kinds of requests is believed to be fairly low. Adding the presumption of a user having cookies disabled (by default they are enabled), the likelihood of this kind of a situation becomes relatively insignificant.

The most troublesome part of log entries processing is concerned with the HTTP user-agent strings, where the visitor's browser, operating system and any other information about the capabilities of users' system are presented.

The World Wide Web Consortium (W3C) Hypertext Transfer Protocol (HTTP) defines the syntax of standard HTTP/1.1 header fields, including the user agent field[1]. It states that the user-agent request-header field contains information about the user agent originating the request. The field can contain multiple product tokens and comments identifying the agent and any sub-

[1] http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

products, which form a significant part of the user-agent string. By convention, the product tokens are listed in order of their significance for identifying the application. Thus, the W3C gives general guidelines and leaves the actual format of user-agent strings open. Thereby, these strings contain various data and have become application-specific. Figure 3-7 shows two examples of user-agent strings and their tokens. Typically each application (browser, crawler, etc.) identifies itself by user-agent string, providing its application type, operating system, software vendor, or software revision and other variable data. Some browsers, like Opera, allow the user to set the user-agent value reported in the HTTP header. Thus, even though user-agent field is widely used for tracking user platform, its firmness can be compromised.



```
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB; rv:1.9.2.3)
Gecko/20100401 Firefox/3.6.3 (.NET CLR 3.5.30729)
```

```
Windows-RSS-Platform/1.0 (MSIE 7.0; Windows NT 5.1)
```

A
Application platform
Version
Platform
Security values:
▪ N - no security
▪ U - strong security
▪ I - weak security
Operating System (NT 5.1 = WinXP)
Language Tag
CVS Branch Tag Gecko version
Gecko engine / Build date
Application name / Version
.NET installed Version 3.5.30729
B
Application platform / Version
Platform
Operating System (NT 5.1 = WinXP)

Figure 3-7. *Examples of HTTP user-agent strings: (a) request sent by Mozilla Firefox browser, (b) request sent by a special platform.*

The processing of the user-agent field in the log analyzer is based on regular expressions, which provides a concise and flexible means for matching strings of text. MySQL has a built-in capability to process regular expressions and this is presently used to extract visitors' browser and operating system from the user-agent string, if available. There exist also web services to process the user-agent strings allowing instant detection of online visitor's web browser and computing capabilities information (e.g., the browser detection web service provided by FraudLabs[1]); however these services are usually provided for a fee.

While processing the web access log, the analyzer keeps a separate log on this process itself on 4 different levels: warning, error, information, and debug level. This enables to trace errors in the analyzer system and if needed define new classifiers to improve the analyzer tool. For example, a new browser is marketed and from the log we can also find a new kind of a reference for this browser, which however is not automatically known by the system. In the analyzer this raises a warning that a system was unable to detect the browser.

[1] http://www.fraudlabs.com

Through this mechanism the analyzer is also responsible for maintaining the database of web crawlers. Errors in the log however refer to a situation which has caused the parsing to stop. The information level describes system events such as the start of parsing, and so on. The debug level on the other hand allows tracing of the parsing process step-by-step.

The data processing with the log analyzer results in a web usage log database containing information ready to answer questions of general usage statistics as well as to be queried for solving specific tasks such as web improvement and personalization based on users' behaviour, as will be shown in the forthcoming chapters of the thesis.

3.5 General Analyses on Log Data

The web usage log produced by the log analyzer system can easily answer questions of general statistics about website and its visitors. These general log analyses can be divided as:

- Visitor statistics (e.g., visitor origin, browsers and OS's used);
- Site specific statistics (e.g., site referrers, errors, page hits, levels);
- Activity statistics and trends of usage over a period of time;
- Session-based statistics (e.g., user action paths and patterns, searches, session length, performed actions).

In the following, several analyses conducted in 2006 and published in [122] based on the collected and processed log data are discussed with the aim to demonstrate the kind of possible analyses based on the web usage log, and the value they carry. The presented analyses are limited to the study period of 2002 – 2006, and whenever valuable and found necessary the results have been supplemented with corresponding results from 2011. The initial analyses were based on more than 882 700 records (217 384 sessions) captured from the research environment DCE WIS (ati.ttu.ee) over the time period of 4 years. They provide a characterization for the research environment ati.ttu.ee (previously discussed in Subsection 3.3) – a background information needed for research described in forthcoming chapters. Logging of web usage data continues.

3.5.1 Viewing Experience: Browsers, OS's, and Screen Size

Good viewing experience is what impresses users and guarantees they wish to return to the site again. The analyses on viewing experience are bound to browsers, operating systems and screen sizes used to access a website, providing webmasters, system administrators, designers, marketing staff and other stakeholders with information about the hard- and software platforms used. Mostly the experience is influenced by the browser capabilities to render website with all the "bells-and-whistles", and available screen size. However, the same browser may act differently on various operating systems (OS), which possesses the need to include OS as a part of the analysis. The screen size is mostly concerned about scrolling, as websites should be viewable on most of the screens sizes, not veiling important parts of the site and introducing a need to scroll. In particular, horizontal scrolling should be avoided. The target of the viewing experience analyses is to detect:

- Browsers (and their versions) used to access the site,
- Operating systems (and their versions) used to access the site,
- Screen capabilities of hardware used to access the site.

Analysing visitors' hard- and software platforms is essential in the context of users' visual experience. The analyses enable detecting of available platforms and allow designers and developers adjust systems in a way that the visual experience users get from a website visit is the most pleasurable and in the long run the best possible. Browser analysis reveals different browsers and their versions used to access the site, enabling to test the users' browsing experience on different platforms, also on those not covered during the design and test phase, but still used to access the website. Despite the standards set for the HTML, CSS, and web content development, browsers still tend to render content differently, especially the Microsoft Internet Explorer, causing problems to site developers as well as to users, as it is rather costly to develop systems for different browsers, or to take into account all the quirks of one certain browser.

Table 3-1 presents the results of the browser analysis for the DCE website with a comparison to global statics (columns A and B). At the time of the analysis (2006), the site was mostly viewed using the Microsoft Internet Explorer (versions 4, 5 and 6), followed by the Mozilla Firefox and other Mozilla-based browser variations. By 2011 this has changed in favour of Mozilla Firefox, though Internet Explorer still holds a share of almost a third of accesses. The analysis also revealed Netscape 4.7x series browsers (released in 1999 and maintained until 2001) were still is use in 2006. It is well-known that Netscape 4.7x series browsers do not fully support CSS, therefore for example ignoring of this fact could have led to undesirable layout of pages at that time for some visitors. The other browsers identified during the analysis were AppleWebKit, now known as the Safari web browser, Konqueror, Lynx and Galeon.

As can be seen from Table 3-1, the statistics for a particular website may vary a lot from global statistical averages. The results for 2006 have been recalculated as [122] covered only the statistics for January 2006. Even though global statistics can show the trends, one cannot rely on them in maintaining a website as these wrong fundamentals may lead to complications. The results shown in Table 3-1 do not show browser version shares, however, the log enables to dig that information very precisely. It could play a crucial role in backward compatibility of applications for users not using the latest versions of browsing software. Also a combination of browser and OS might be of interest, which can be also queried from the web usage data.

Table 3-1. *Browsers used to view the DCE website through 2002–2006 [122] with corresponding results in 2011; and a comparison to global statistics: column A - W3Schools.com*[1]*, column B - StatCounter Global Stats*[2]*.*

Browser	**Browser usage for DCE website [%]**							**Global statistics 2011**	
	2002	**2003**	**2004**	**2005**	**2006**	**…**	**2011**	**A**	**B**
Internet Explorer	90.2	87.7	83.6	64.9	60.5	…	28.4	20.2	38.7
Netscape 4.x	5.7	5.3	3.4	0.1	0.3	…	0.0	n/a	n/a
Firefox	n/a	n/a	5.9	22.6	27.9	…	41.8	37.7	25.3
Other Mozilla-based	2.5	4.8	5.0	9.2	7.9	…	0.3	n/a	n/a
Opera	0.4	0.4	0.6	2.3	3.2	…	7.3	2.5	2.0
Chrome	n/a	n/a	n/a	n/a	n/a	…	19.2	34.6	27.3
Safari	n/a	n/a	n/a	n/a	0.1	…	2.6	4.2	6.1
Others	1.2	1.8	1.5	0.9	0.1	…	0.4	0.8	0.6

Browsers may act diversely on different operating systems, and platforms may have their own peculiarities. Table 3-2 outlines the results of the OS study for the DCE website. As can be seen, the majority of accesses are performed on Microsoft Windows-based computers; and lately operating systems on mobile devices such as iOS and Android have appeared in the log.

Table3-2. *Operating systems used by visitors of the DCE website.*

OS	**2002**	**2003**	**2004**	**2005**	**2006**	**…**	**2011**
Windows	97.7	99.7	99.4	96.8	93.1	…	88.5
Mac OS	0.0	0.1	0.1	0.2	0.5	…	2.6
SunOS	1.7	0.0	0.1	0.9	2.0	…	0.0
FreeBSD	0.3	0.0	0.3	0.2	0.7	…	0.0
Linux	0.0	0.0	0.1	0.2	2.5	…	7.7
Mobile OS's	0.0	0.0	0.0	0.8	0.0	…	0.8
Unknown	0.2	0.1	0.1	0.3	1.2	…	0.4

Yet another aspect to concern with displaying a webpage is the screen size available on visitors' devices. The screen sizes study for the DCE website

[1] http://www.w3schools.com/browsers/browsers_stats.asp

[2] http://gs.statcounter.com/

revealed 49 different resolutions to have been used. The results were further classified into the groups represented in Table 3-3. The majority of visitors used 1024x768 pixels resolution (2004–2006) and by 2011 had upgraded their hardware to better screen resolutions. This clearly illustrates the evolution of technology for display sizes and a need to adjust websites to better fit to screen sizes primarily used now. The DCE website was initially set to satisfy the requirements of the 800x600 and 1024x768 pixel screens, as it was the mostly used resolution in 2004. According to the analysis, over the years there has been a shift to screens 1280 pixels wide and even wider, on which the DCE website presently is surrounded with an area of unused space.

Table 3-3. *Screen resolutions used while accessing the DCE website.*

Screen size	2004	2005	2006	…	2011
Lower than 640x480 [%]	0.0	0.0	0.1	…	0.2
640x480 [%]	0.1	0.1	0.1	…	0.1
800x600 [%]	10.4	5.6	2.0	…	1.6
1024x768 [%]	65.5	65.3	53.1	…	9.4
1152x864 [%]	6.5	4.9	7.2	…	0.4
1280x1024 [%]	15.6	21.0	32.9	…	37.8
Higher than 1280x1024 [%]	1.7	3.0	4.6	…	49.6
Unidentified [%]	0.1	0.1	0.0	…	0.9

It is essential to know technical capabilities of your visitors' devices to successfully maintain a website and enhance users' viewing experience. Fortunately, today there is a variety of web analytics solutions available to choose from and get such statistics for websites in order to take prudential decisions.

3.5.2 Source of Visitors

Analysing the source of visitors answers the questions: where do the visitors come from and who are they in general? This analysis reveals how users have reached the site and who on the World Wide Web is referring to it. Knowing the source of the visitors allows planning and improving of the site content, and provides a good basis for marketing actions like site promotion through banners, subscribing to search engines, and so on. Typically the analysis addresses the following partitions: direct traffic, visitors arriving through search engines, and referring sites. Depending on the website purpose, different visitor segments can be identified.

For the DCE website, the analysis conducted in 2006 [122] reviewed visitors in two categories: new and returning users. Figure 3-8 highlights origin of visitors for the DCE website based on the location of their IP. The majority of new visitors as well as returning visitors have come to the department's (DCE) website from the university's web space, respectively 27% and 22%. There were 9% of new visitors outside of Estonia and those of returning 5%. The share of (detected) returning visitors is a fair 39%, which in reality can be even higher, as the statistics is based on anonymous web usage data. In 2011 the rate of returning users was 55%.

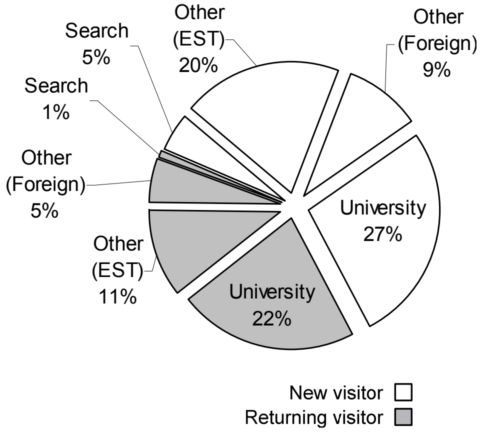


Figure 3-8. *New and returning users on the DCE website; selected by access domain.*

Figure 3-9 on the other hand presents the source of visitors by referring sites. As seen, in 61% of cases users have come through other pages in the university's web space, while 17% of visitors used a search engine reference and 22% of linked visitors came through other sites. In 2011 the corresponding values were 58%, 29%, and 13%. The analysis revealed that the DCE web was referred as an external source through more than 89 different websites at the time of the study.

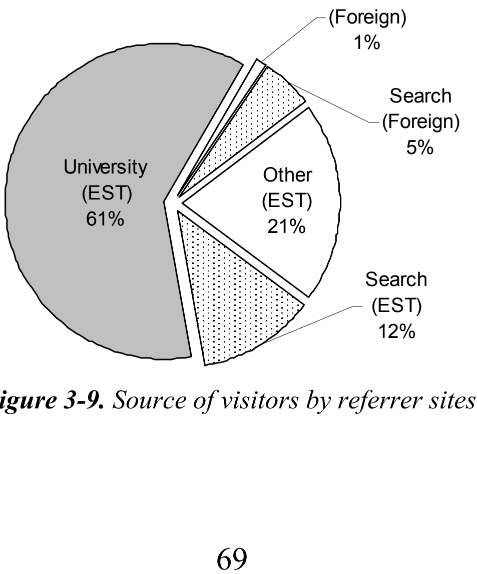


Figure 3-9. *Source of visitors by referrer sites.*

3.5.3 Access Trends on Timescale of 24-7-12

When planning site maintenance or changes, it is useful to know website access trends in time. The 24-7-12 time analysis (Figure 3-10) represents three major time characteristics on one scale of 24 divisions, where in each case the time has its own dimension (hours from 1 to 24, months 1–12, weekdays 1–7), enabling to introduce all the access trends on one graph. Access rate is found for each time characteristic individually over the time period being studied.

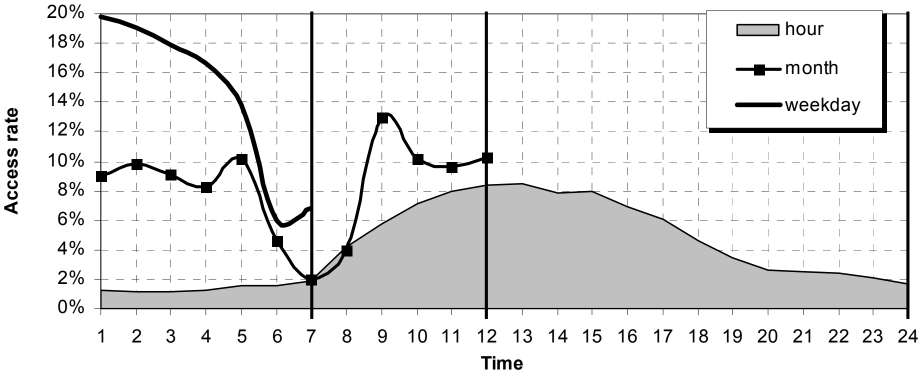


Figure 3-10. *Website usage graph for access trends in three time categories: hour, weekday and month. All times local (server) time.*

As can be seen from the graph (Figure 3-10), the monthly access trend for the DCE website has its peak values in February, May and September – these are the beginning and end months of academic semesters [122]. The access rate during summer time is very low, for instance in July only 2% of total hits. The reason is obviously in summer holidays – not much happening at universities during the high summer peak. The weekday and hourly access trends show the site to be accessed the most at the beginning of the week (1 stands for Monday on the graph) and during the midday. Hourly access, which is measured in local (server) time, becomes very useful for system administrators for maintenance planning, especially when serving clients from other time zones. In addition to maintenance, knowing the access trends also provides a good basis for e-commerce and marketing, e.g., raising the importance of certain banners or other directed information at peak access times to target more visitors.

3.5.4 Menu Structure Level and Page Load Analysis

Analysing website menu structure by levels together with pages by their loading times can reveal bottlenecks in the system otherwise undiscoverable. Besides, once a site is designed and its menu structure planned and implemented, changes may be necessary, as nobody can predict how users actually are going to use the system after its release.

The aim of the level analysis is to show whether the menu organization is efficient enough and to indicate possible bottlenecks. As users' access trends may change over time with the change of the site content, also the menu structure might be in need of some renewing. To expand the possibilities of level analysis, each item could be assigned a characteristic weight to represent its relative importance in the system, which would enable to perform automated assessments by levels.

In [122] the author proposed to use a graphical representation of hits per website item to easily discover whether there are resources that need to be relocated to minimize the number of clicks users have to make to reach an item. An alarming situation was defined as a case when an item on lower menu level is accessed more frequently than its parent item, indicating a probable structural design error. Figure 3-11 displays a top 20 of accessed items for the DCE web. As shown, a page with ID code "1505" has had twice as much hits as its parent item with code "150". The same applies to item pair with codes "410" and "4110" (a descendant of "410"). Consequently, these items need to be investigated in more detail, to detect whether they should be relocated at least one level upwards in the menu structure. This however cannot be done automatically and without knowing the actual page content. Thereby, the level analysis graph is considered as a useful tool for webmasters to assist decision-making over necessity of improvements. For example, in the case of item "4110", the "410" is a second level parent page representing all the subjects lectured at the department, whereas "4110" is a third level child page representing detailed view of a subject selected on page "410". Thus, the context justifies the detected anomaly and does not demand for a necessity to rearrange the structure concerned with item no. "4110".

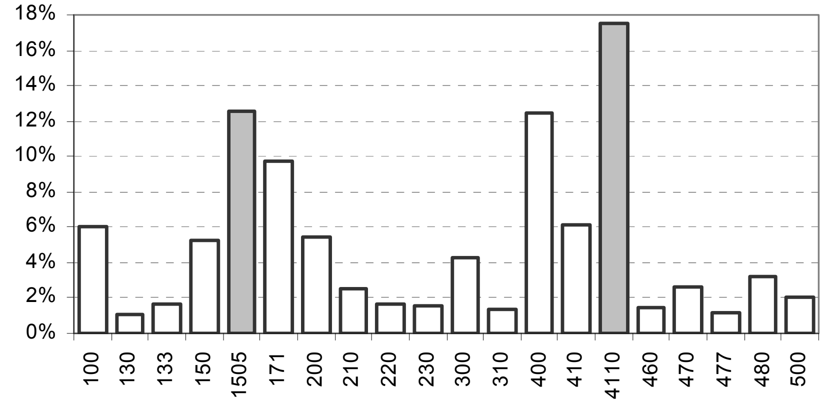


Figure 3-11. *Level analysis showing access rates per site item. Gray bars indicate possible problematic areas in the menu structure.*

Analysing page load times can reveal problematic dynamically generated pages. Studies on page load times have revealed that if users have to wait for more than a few seconds without any feedback from the system, they get impatient and unsatisfied [123]. For example, with the help of this analysis, an anomaly of unallowably high load time (12.3 seconds) of a page appearing only under certain circumstances was discovered in the e-EDU e-learning system

[120]. After repairing the defective database table and introducing some more indexes, the page loaded more than ten times faster. High page load times may be caused by high server load as well. For this reason, the log system stores also the recent values of server load at the time of each request – yet another advantage over web server log files.

The proposed analysis of menu structure and page load times help to detect problem areas in the system, which could be left unnoticed otherwise.

3.5.5 Search Combinations in Sessions

Session-based users' behaviour analyses add value to previously discussed general informative analyses, reflecting the efficiency and usability of a website. Analysis on search combinations should be carried out over two distinct user groups: new and returning visitors to enable comparison of site usability of those users who visit it for the first time and by those who have got accustomed to it already.

An initial study on search combinations in sessions to evaluate users' navigational behaviour was conducted in [122], proposing to measure the efficiency by the time between two operations performed. The average time between two operations was calculated as $\Delta t = t_i - t_{i-1}$, where t_i is a timestamp for a particular operation. The results (Table 3-4) indicate that users reached their requested target information approximately in 22 seconds. The average session length of approximately 4 minutes and the maximum search time of approximately 2 minutes (Δt_{max}=127s) indicate that in most cases the data has been successfully found in relatively short time. The results in Table 3-4 also revealed that returning visitors performed averagely more operations in session than new users, which could be explained by the engagement of returning visitors and by their explicit and predefined informational needs. The somewhat higher time between two operations performed on the first menu level is probably caused by the fact that the information presented on the level is mainly a generalization of specific data presented on the second and third level, thus users may need some time to think where to click next. The most specific data is presented on the third menu level, although whenever possible the information was not partitioned and presented on the second level to minimize the number of clicks. It is highly likely that the average search time (21.3 seconds) on the third level is greater than on the second level (19.4 seconds) due to the more specific information presented on the third level. For proper interpretation of the results of timing analyses, they have to be reviewed by a domain specialist.

Despite the subjectivity of this analysis, it provides an overview of the site's performance in sense of returning and new visitors. The statistical numbers give little information to a person not familiar with specifics of the website; however, for a specialist like the webmaster they can provide a valuable characterization and pose a need for deeper analysis.

Table 3-4. Search combination characteristics in session

Type of analysis	Avg. number of operations		Avg. time between two operations		
	Single visit	Recurrent visit	Δt [sec]	% of Δt	Δt_{max} [sec]
All levels	2.7	5.4	22.1	100	127
Menu level 1	2.4	6.2	26.2	119	n/a
Menu level 2	2.4	6.9	19.4	88	n/a
Menu level 3	2.5	6.2	21.3	96	n/a
Session length	n/a	n/a	220.3	n/a	n/a

3.5.6 Correlation Study on Session Parameters

Amongst other studies executed in 2006 and published in [122], a correlation study on session parameters was conducted. The study targeted to determine whether there is a clear connection between the length of the session and the number of clicks users perform during it. This study itself is not so tightly connected to any specific web application or its visitors' analyses. Instead, it helps to understand users' behaviour, in particular the relation between the quantity of operations and session length.

Figure 3-12 shows the correlation discovered during the study. As can be seen, there is a distinct and clear interrelationship by linear allotment between session time and number of performed operations until the quantity of clicks grows to around 10. After that, the correlation becomes ambiguous, which can be set off by the human ability not to remember all the actions performed on the site, in particular operations made several clicks ago – humans' short-term memory is believed only to last around 10–30 seconds, depending upon the number of items to be retained in memory, and to be limited to 7±2 items [124], [125]. Users remember having seen "something somewhere", however they might be unable to recall the exact location. It is comparable to the situation when visiting an unfamiliar city and getting lost. This explains the peaks on the graph, representing rapid surfing on the website. Another corresponding finding of this study was that the time between two performed operations shows a trend of steady linear decline as the number of operations performed in a session increases. This could be due to users' restlessness and unwillingness to concentrate on the content, feeling being lost or a decrease in interest.

These findings suggest that there is a strong dependence between session length and the number of operations performed during a session. Obviously, the clearer and logical the site's organization is, the more users can recall their previous actions; on the other hand, if there is no linear allotment at the beginning of the graph, it is alarming that the site might have a confusing structure. The findings of this analysis correspond to a similar study on session duration versus visit actions conducted by Markov and Larose [28], also confirming that there exists a strong positive association between the number of visit actions and session duration.

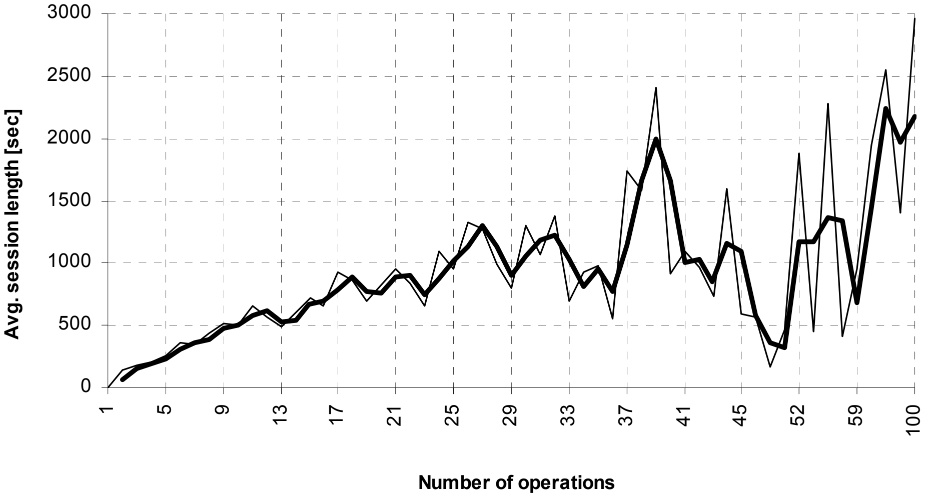


Figure 3-12. *Correlation between session length and operations performed. Average correlation trend line is shown in bold.*

3.6 Discussion

The list of studies presented in this chapter is not definitive as the collected log data enables producing of various analyses according to one's needs. These could include the following: top visited and referred pages, entry and exit pages, time spent on page, list of countries visits originating from, different languages of visitors (based on user-agent string), traffic segmentation (direct, referred, search engine redirects), referring sites and search engines, list of web robots crawling website for information, comparative studies on returning and new visitors on different basis, and so forth. Obviously, this list could be continued as long as all the different aspects of the web usage log data are described in all possible combinations. Hence, the log system and the log analyzer system enable to achieve more than described herein for demonstration and research environment characterization.

Today, many of the described analyses have become common and easily accessible via online analytics, such as Google Analytics, enabling a good set of predefined reports. At the time the log system was developed and the analyses conducted, the situation on the analytical tools market was however quite different. Nevertheless, the necessity to analyze web usage and reason about the results has remained and led to establishment of new tools and services. Over time, analytical tools previously implemented as desktop programs have evolved into web-based systems and services, rich in features unthinkable years ago. Yet, they have mostly remained commercial environments.

3.7 Chapter Summary

Logging web events can reveal a lot about a web information system, its usage and users. The importance of web-based systems has rapidly increased over the past years and the process continues. This has introduced a need to be able to improve web information systems and address the systems to match a user-oriented approach.

This chapter has discussed the importance of logging users' operations on the web, and ways to achieve this. More precisely, the chapter has addressed web server logs, hosted web event logging and the special log system developed by the author to capture users' actions on the research environment, the DCE website, as well as many other websites. Further the chapter introduced the log analyzer system, which is used to process collected web usage data. This was followed by a set of analyses and studies performed on the web usage data, to demonstrate the potential of the log system and its analyzer system, and to characterize the research environment. The analyses discussed herein have been mostly limited to the studies performed in 2006 and presented in [122]; many of the possible user and usage analysis have been omitted as they are easily derivable from the processed log data and would not have added any extra value. Nevertheless, these analyses have been discussed briefly.

The chapter constitutes the first phase of the author's web studies, continued by a research into evaluating web information systems on the basis of users' behaviour modelling, users' action prediction on different basis, and web personalization and recommendations generation, discussed in the forthcoming chapters.

Chapter 4
EVALUATING AND OPTIMIZING WIS'S ON THE BASIS OF USERS' BEHAVIOUR

Today, web presence has become essential to any successful company. Depending on company profile, websites are used to provide information, services or both, where the World Wide Web is chosen to serve as the platform to promote and support company interests. The power of online presence cannot be underestimated anymore. Well-developed and constantly maintained websites that acknowledge their users' interests and needs, positively affect visitors and prospective customers, and thereby help to establish a reflection of a successful and trustworthy company.

This chapter concentrates on the efforts made to investigate the benefit gained from studying users' browsing behaviour in favour of improving web information systems. In particular, several indicators of users' interest are described and applied to evaluate the current state of web information systems, and identify areas in need of improvement.

4.1 Introduction

Visitor satisfaction is a key success factor of any web information system, regardless whether it is a simple website or a more sophisticated system, driving the need to continuously improve such systems to better conform to changing demands and expectations of users. Additionally, over time websites evolve, as information is added, modified or deleted, causing pages and links to appear in unlikely places in comparison to the original design. Also, in practice sites may be used in ways they were not designed for primarily. Studying users' behaviour allows to evaluate and optimize such systems, targeting at the same time visitor satisfaction. Although the latter could be studied through explicitly collected data, such as questionnaires, or interviews, this approach cannot fully cover all WIS users and requires extra efforts to be enforced. Implicitly collected web usage data on the other hand covers all system users and does not require any extra actions to be carried out rather than web usage mining.

Application of WUM for adaptive and personalized web is becoming common for providing users with customized views to websites or dynamically discovered (personalized) recommendations during their web visits. Yet, little attention is paid to possibilities of improving WIS's usage by studying users' behaviour. The way visitors behave on a website can reveal a lot about its usability, as users are browsing the web according to their personal experience, informational needs and understanding of the topics presented, being at the same time influenced by site design and its comprehensibility. Information about users' behaviour, captured in web usage logs, allows to analyze and discover users' preferences for adapting WIS's to their needs. Reconstructing users' operations in sessions and reasoning about them helps to understand users' intentions and expectations towards a system, to bridge the conceptual gap between the developers' domain model and actual users' domain model (Figure 3-1), and deliver valuable improvements to a website and its structure.

This chapter presents a framework for evaluating web information systems based on user behaviour modelling, in particular on recognition and mining of navigational paths and patterns users typically follow while browsing. The log system and the log analyzer discussed in Chapter 3 provide data sufficient enough to evaluate WIS's based on users' behaviour, and are used for the task. A set of indicators to assess websites from usability point of view will be discussed. Web usage data is processed and users' navigation paths constructed in sessions based on the collected behaviour data. These navigation paths along with other data available in the log form the basis for WIS evaluation. Throughout the studies, new and returning visitors will be compared.

The rest of this chapter is organized as follows, Section 4.2 discusses indicators of user interest, while Section 4.3 explores several metrics for WIS evaluation, starting off with identifying pages of popularity and ending with studies on cyclic operations in users' navigation paths. This chapter also introduces the locality model, which will be applied herein and in the forthcoming chapters for processing users' behaviour data.

4.2 Indicators of User Interest in WUM

Different indicators of user interest have been proposed by researchers of the field, as has already been discussed in Chapter 2.8. Let us now take a closer look on indicators available from web users' behaviour stored in web usage logs. Users' interest towards web pages is mainly measured by page ranking, achieved in the WUM community usually by the order of visited pages and their popularity (hit count). Although this approach reflects the interest on the count of clicks, it has a serious drawback – it does not pay any attention to the actual time spent on page (TSP) as Hofgesang outlines in [108]. He claims that TSP is a clear and natural indicator of importance of a page in WUM – the more time users spend on a webpage, the more important it is assumed to be.

In general, the indicators of user interest (identifiable by actions from web usage logs) are as follows:

- List of pages users are visiting, and their order,
- Popularity of visited pages – frequency measure,
- Time spent on pages (TSP).

The most commonly used and the most trivial way to measure visitors' interest towards a site is to list accessed pages and their hit count per page. It gives a clear response on what pages are of more interest to users than others, and already this may indicate a need to make some changes in the system so that pages with high hit rate would be easily accessible, for instance via shortcuts on the index page. However, this frequency measure is dependent on website's navigational structure. In particular, the location of a page in the site's navigational hierarchy influences the statistical popularity, i.e., the frequency of page visits. Obviously, pages at upper level in the navigational hierarchy have higher access frequency than pages on the descendant levels due to the fact that these pages often act as intermediate linking pages and not as much as content pages. Still, navigational position in the structure has no direct effect on another interest indicator, namely time spent on pages (TSP).

TSP is an important indicator of user intention and interest towards viewed pages and their content on a website, and therefore plays a crucial role in the proposed models in this thesis as well. It is yet another implicit measure of user interest attainable from actions visitors perform while browsing. TSP as an interest measure has been studied by many researchers in [18], [108], [109].

Despite the apparent simplicity of this indicator, there are some major problems to consider. Time spent on a page can be measured either on client-side or on server-side; nonetheless Srivastava et al. [13] have found in their research client- and server-side measuring of TSP to be equal or in favour of server-side approach because of the overhead on client side. Thus, the use of page access timestamp captured by the log system is more than appropriate for the tasks discussed in this thesis.

So what does influence time spent on pages and how to properly measure it? Ideally, TSP is only the time users spend on particular page actively dealing with it, i.e., reading, scrolling, and so on. Nevertheless, factors such as the time spent for page generation on the server, time spent on data transfer over the network and the effect of user distraction have to be also considered. From the aforementioned factors, user distraction as a set of activities unrelated to browsing is the most troublesome as it cannot be measured from web usage logs. Based on the access data in the log, it is impossible to detect whether a user was chatting (e.g., telephone, instant messaging, colleagues, family, etc.), had a break (e.g., coffee break) or was engaged with other activities while browsing a site, resulting in higher TSP values. Yet, user distraction, which is one of the drawbacks of TSP as an interest indicator, has no direct effect on the frequency measure. To tackle the user distraction problem, in practice an upper limit for TSP is usually applied; Hofgesang [108] reports a limit of 90 seconds to 10 minutes. Researchers, who have been using customized web browsers, like the 'Curious Browser' implemented by Claypool et al. [18] however have been able to measure the distraction to some extent by detecting whether the browser window was in focus or not. It is impossible to get such data from web server logs; however, with customized web event logging this could be now achieved. Namely, the HTML5 standard browser window events ‘onpageshow’ and ‘onpagehide’, and the W3C Page Visibility API[1] now provide means to detect current visibility state of a page. However, not all browsers provide support for these features yet. The latter could also be used to solve the problem of measuring TSP for exit pages (page viewed before leaving website or closing browser window), as today it remains immeasurable not only by the log system introduced here but also by hosted web analytics tools. The evolution of HTML and web technologies will in the future enable to capture web usage events more precisely. As a future development of the log system active TSP measuring is planned to be added keeping in mind future research work.

The formula for the ideal TSP is given by Equation 4-1, where t_i is the timestamp of a given operation (page view). However, in practice, TSP is usually calculated just as the time between two page requests as given by Equation 4-2 (with or without $t_{page\_generation}$, depending on its availability), and an upper limit is applied to eliminate the impact of user distraction. This approach presumes that page generation and transfer times are relatively small and user distraction is immeasurable. For instance, for the DCE website the average page generation time according to the log is 0.07 seconds, with higher peak values at 0.1 seconds. Also, page sizes today are a few hundred kilobytes and network bandwidth provides transfer rates high enough to make this component in Equation 1 insignificant compared to other elements.

$$t_{TSP} = t_{i+1} - t_i - t_{page\_generation} - t_{transfer} - t_{user\_distraction} \tag{4-1}$$

$$t_{TSP} = t_{i+1} - t_i - t_{page\_generation} \tag{4-2}$$

The minimum threshold value for TSP should be connected to the time required to accurately asses a page, i.e., to at least a few seconds, as this is the time slot when visitors make their decisions whether to stay on the page or move forward. According to [109] users are unable to accurately assess interest towards a page in less than 5 seconds, evaluating the importance of the information found on the page during the first few seconds [110]. This imposes a minimum (recommended) value for TSP. Very small TSP values may be caused by accidental double-clicks, or indicate robot transactions (if not filtered out) or redirect pages, which certainly are not of interest. The averages for web page view times as a measure of interest rate however vary from 12 seconds [111] to 48 seconds [126]. Still, Weinreich et al. [111] have found that nearly in half of cases users were spending less than 12 seconds on pages they visited, leading to a conclusion that they browsed for a next page before reading a substantial part of the content presented on a page. Moreover, experiments in

[1] http://www.w3.org/TR/page-visibility/

[108] refer to a fact that users spend less time on pages they visit frequently. Al habibi et al. [14] concluded in their study of three different attributes of time spent on page – minimum, maximum, and average TSP – that there exists a strong tendency for users to spend more time on pages they find interesting compared to those they find not so interesting, with a non-linear reference to page rank. The analysis of the DCE website usage log shows that in average new visitors spend 19 seconds on a page while returning users 17 seconds for the segment of TSP=[5..60]s. With a reasonable value of TSP>60s, in particular TSP=600s, the corresponding values were as follows: average time spent on page for new visitors 119 seconds and for returning users 72 seconds. These findings also correspond to the findings of other authors. Figure 4-1 outlines average TSP values for interpreting user interest based on literature and empirical studies on the DCE website.

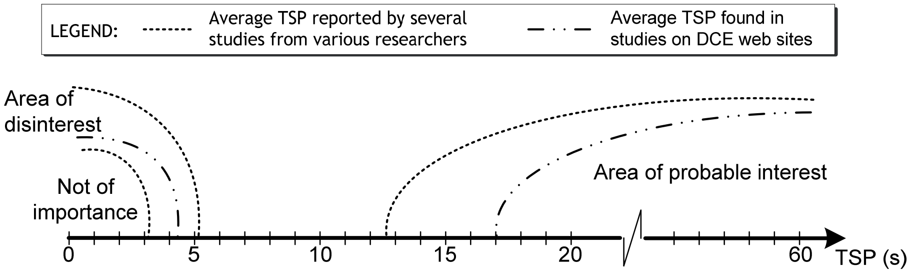

LEGEND:
Average TSP reported by several studies from various researchers
Average TSP found in studies on DCE web sites
Area of disinterest
Not of importance
Area of probable interest
0
5
10
15
20
60
TSP (s)

Figure 4-1. Recommended average values for interpreting TSP as an interest indicator.

Although, TSP is a good measure of user interest, it also suffers from some drawbacks to be considered. One of these – user distraction – has already been mentioned. The other major drawback to consider is the fact that TSP can also be largely influenced by page content and its type (linking page, media-rich page, and so on), but also by the quality of layout (length, logical placement of objects, etc.). More precisely, on linking pages there is little information to read and users proceed to select another page via link, whilst content pages may have several paragraphs of text, or media content (e.g., images and videos), which could lead to high values of TSP. Here, the HTML5 and page visibility API being introduced to modern web browsers might help to determine user engagement. In addition, visitors as individuals have different speeds of reading and navigating content, which also has its small effect on TSP.

For text-based articles, studies on the effects of human reading performance have shown that in general users are able to read unfamiliar text content on screen at speed of 180 words per minute (wpm) in contrast to 200 wpm from hardcopy [127]. The average reading speed value 180 wpm reported by Ziefle [127] enables to calculate estimated time needed to read full article on a particular page, and compare it to the actual time users spend on it. Also, it must be considered that since Ziefle conducted the study, with the development of electronic devices to read from (computer screens, smart-phones, tabs and pods, etc.), people probably have adapted to read faster also from electronic screens.

So the reading speed could vary in between 180 – 200 wpm. Automatic evaluation of video and especially graphical content viewing times however could probably need human intervention.

In [128] the thesis author investigated article reading speed on the DCE website, finding the average value to be 199 wpm. Regardless of this, it was interesting to find that for long articles users tended to spend less time than the average calculated hypothetical time would propose; therefore evidently reading only a portion of the article and then moving on. Shorter articles however got more attention from users in respect to time spend on an article page. The border seems to be somewhere near 220 words. There was no significant difference for TSP values for new and returning users, except for index page, where returning users tended to spend approximately 10 seconds less than new visitors. In this study website content was categorized as follows: short articles with content length up to 250 words, long articles with content of more than 250 words, comprehensive articles as specific in-depth content, and common articles as regular website content. The study showed that more sophisticated articles were explored for longer time irrespective of their content length, whereas too general articles (common) were briefly read. It was also noted that visitors tended to spend more time than the average hypothetical calculated time would suggest on the index page. This is probably due to the need to familiarize with the page and its structure, besides the article content. Figure 4-2 outlines the results of the TSP versus content length study for the DCE website.

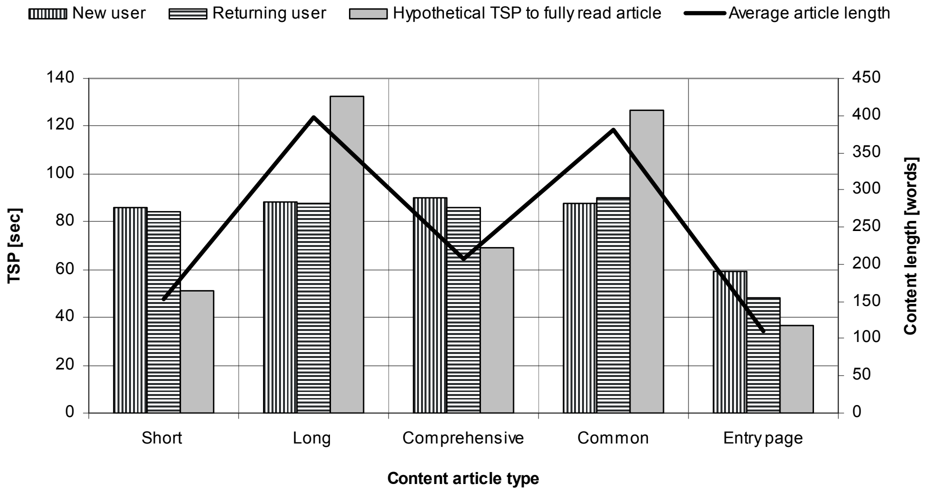


Figure 4-2. *Dependence between time spent on page and article content length.*

Obviously, in an efficient web system pages that get more attention from users according to this indicator should be reachable with a few clicks, whereas pages identified as of low interest need to be positioned on deeper navigation levels, unless someone intentionally wants to reinforce and bring them forward. Pages that are not at all of interest for visitors by this indicator should not be a part of the system. The evaluation of the results is the responsibility of an area expert, i.e., the webmaster.

Despite of its drawbacks, TSP is still a valuable indicator of user interest and used together with other metrics, can clearly indicate user interest.

4.3 Metrics of WIS Evaluation on Users' Behaviour

Let us now turn to what can be learned from users in favour of evaluating and improving WIS's. The collected and processed web usage log provides primary data to model users' behaviour and search for anomalies and bottlenecks in the web system usage flow. The operations in users' navigation paths with their attributes clearly describe the ease or difficulty of site navigability.

The framework discussed herein addresses web information systems evaluation and necessity for improvement through the following models:

- Popularity of pages with applied TSP,
- Rate of void operations in navigation paths,
- Rate of multi-sequential identical operations in navigation paths,
- Cyclic operations in navigation paths.

These evaluation models address users' sessions in two distinct groups: new and returning visitors based on the available data.

The discussion of these models is accompanied with results of empirical studies conducted on the DCE website. These studies were based on 269 782 sessions derived from the web usage log. Most of the results of these studies have been published in [129].

4.3.1 Steps of Data Processing for WIS Evaluation

The main processing of raw users' activity data has already been done with the log analyzer tool, resulting in web usage log data. This data is now exploited for WIS evaluation through users' behaviour modelling, providing answers to questions of how the system is used and what could be improved in it. The process starts with acquiring necessary data elements from the log analyzer data storage repository and reconstructing users' browsing sessions. A user session $s=<p_i, p_{i+1}, \dots , p_n>$ is defined as a sequence of accessed pages, where $p_i \in P$ and P is the set of all the pages constituting a website. With the composition of user's navigation path, each page access is equipped with corresponding TSP values. At this stage, the first set of metrics becomes available. The navigation paths are further processed to minimize the paths and detect void operations, defined by TSP values according to condition t_x. Equation 4-3 describes the threshold conditions for t_x.

$$t_x = \begin{cases} x = 0, t_{TSP} \text{ is not considered} \\ x = 1, t_{TSP} < T_{min} \,||\, t_{TSP} > T_{max} \\ x = 2, T_{min} \le t_{TSP} \le T_{max} \end{cases} \tag{4-3}$$

As a result, the second set of metrics, called void operations in sessions, and the third set called multi-sequential identical operations in sessions are produced. The final set of metrics, cyclic operations in navigation path, takes advantage of the locality model and are detected on the fully minimized path. Figure 4-3 depicts the steps of processing of this data with example session data. Each of these sets of metrics will be discussed in detail in further.

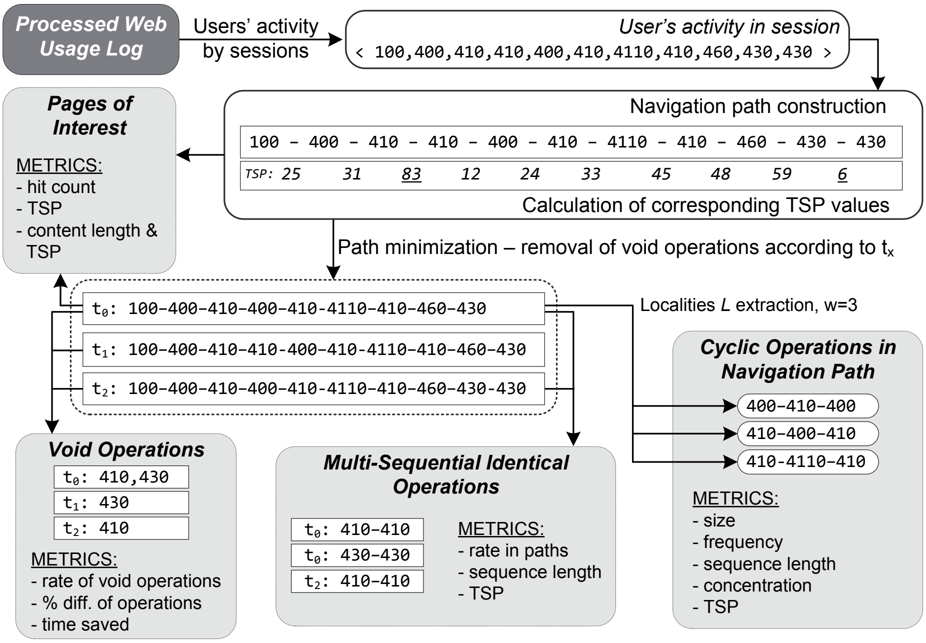

Processed Web Usage Log
Users' activity by sessions
User's activity in session
< 100,400,410,410,400,410,4110,410,460,430,430 >
Navigation path construction
100 - 400 - 410 - 410 - 400 - 410 - 4110 - 410 - 460 - 430 - 430
TSP: 25 31 83 12 24 33 45 48 59 6
Calculation of corresponding TSP values
Pages of Interest
METRICS:
- hit count
- TSP
- content length & TSP
Path minimization – removal of void operations according to t_x
t_0: 100-400-410-400-410-4110-410-460-430
t_1: 100-400-410-410-400-410-4110-410-460-430
t_2: 100-400-410-400-410-4110-410-460-430-430
Localities L extraction, w=3
Cyclic Operations in Navigation Path
400-410-400
410-400-410
410-4110-410
METRICS:
- size
- frequency
- sequence length
- concentration
- TSP
Void Operations
t_0: 410,430
t_1: 430
t_2: 410
METRICS:
- rate of void operations
- % diff. of operations
- time saved
Multi-Sequential Identical Operations
t_0: 410-410
t_0: 430-430
t_2: 410-410
METRICS:
- rate in paths
- sequence length
- TSP

Figure 4-3. *Processing of users' session data for WIS evaluation.*

4.3.2 The Locality Model

The locality model is based on the belief that if a large number of users frequently access a set of pages, then these pages must be related. The locality *L* is defined as the nearest sequential activity history of a user within a site visit (session) determined by a sliding window *w*. During their site visits users are moving from one locality *L* to another, captured by the sliding window of size $w \in W$. Figure 4-4 depicts user localities and their extraction from navigation path.

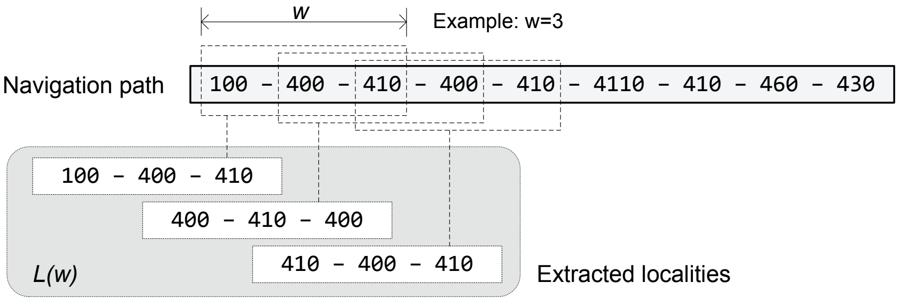


Figure 4-4. *Localities L and their extraction from user's navigation path.*

Localities *L* are extracted from users' sessions available in web usage log. A user session $s \in S$ is defined as a sequence of accessed pages $s=<p_i, p_{i+1}, ...p_n>$, where, $p_i \in P$ and *P* is the set of all pages, and *S* is a set of available sessions in the log. The order in which pages were accessed in a session is determined based on request timestamps. Thus, localities *L* of size *w* are defined as $L(w)=p_j,\ p_{j+1},\ ...p_w$, where $p_j \neq p_{j+1}$ and p_j is the ID for a visited page in the locality. The extraction process is carried out by function *ExtractLocalities(s,w)* applied for each $s \in S$ (Figure 4-5). Unless otherwise stated in forthcoming chapters where the locality model is applied, the localities are extracted from fully minimized navigation paths constructed based on user's actions in a session.

```
Function ExtractLocalities(s, w)
/*
 s: array                  = operations in a user session being processed
 w: int                    = locality window size
 sessionLocalities: array = set of extracted localities
 runningLocality: array   = locality being extracted
 quantity: int            = number of localities extractable from path
*/

    BEGIN
        quantity = count(s) - w + 1
        i = 0
        while i < quantity do
            j = i
            while j < w + i do
                add s[j] to runningLocality
                increment j
            end while
            add runningLocality to sessionLocalities
            reset runningLocality
            increment i
        end while
        return sessionLocalities
    END
```

Figure 4-5. *Algorithm used for localities extraction.*

The proper application of the locality model depends on the sliding window size *w*. As *w* is not a fixed value and depends mainly on the absolute menu depth of a particular website, a study to discover the best value for the sliding window *w* is recommended before the exploitation of the locality model.

An empirical study was performed on the DCE website to discover the best value for the sliding window *w* [130]. The absolute menu depth of this website is three levels. For the evaluation the following attributes were observed:

- Cover percentage for the number of combinations computed from the navigation paths,
- Average frequency of finding these combinations in navigation paths,
- Average number of possible localities in navigation path,
- Availability of next accessed item for each locality. The attribute “next item” is found as p_{j+1}, if available.

The experiments were performed with various window sizes W={2,3,4,5} on 87 953 navigation paths as series composed of unique numerical identifiers set for each page (page ID) in the DCE WIS. The results of the experiment (Table 4-1) suggest that for the DCE website the best window size to use is w=3. As can be seen, localities of size 3 performed the best in respect to observed parameters (higher values provide better efficiency) and considering the fact that w=2 would be too short for describing recent actions in sense of user profiling. The availability of next page request for a described locality *L* is very important for the prediction models of users’ actions discussed in Chapter 6. The results confirmed that the sliding window size *w* is in correlation with the absolute menu depth of a website. These findings also conform to the ones reported by other researchers in [92]. Thereby, to gain maximum benefit of the locality model, it is essential to study the sliding window size *w* for a particular WIS before it is applied to reveal users’ browsing behaviour.

Table 4-1. *Results of the empirical study for determining sliding window size w.*

Properties observed	**Studied window size *w***			
	2	**3**	**4**	**5**
Combination coverage [%]	31.2	35.5	20.7	12.6
Combination frequency	1.1	1.0	1.0	1.0
No of localities in path	6.3	6.6	6.5	5.9
Availability of next item [%]	76.6	77.4	74.1	76.3

The locality model has a central role in a number of studies discussed in the forthcoming chapters. Any particular details concerned with the model are presented within the corresponding chapters.

4.3.3 Pages of Interest

Based on the interest indicators such as hit count for pages and time spent on pages, there are several metrics which can be used to evaluate and improve web information systems. Let us refer to the set of these metrics as pages of interest. These metrics concentrate on single page views rather than on navigation through site structure. Nevertheless, void operations in sessions must be removed beforehand in order to reduce bogus page hits. This is achieved by path minimization phase during data processing (Figure 4-3).

The pages of interest metrics proposed by the framework to evaluate WIS usage are as follows:

- Top N pages by popularity,
- Top N pages with TSP applied to express users' interest into content,
- Pages with low TSP values in respect to content length,
- Pages existing in WIS without gaining any access or significantly low access rate from users,
- Pages existing in WIS without gaining any access or significantly low access rate from web crawlers, leading to no or low indexing.

The '*Top N pages by popularity*' is the most general and widely used metric for WIS evaluation to identify the most accessed pages. Accesses are counted and listed in a descending order for the *N* most requested pages. The value of *N* depends on the website scope, size and webmasters intentions. A top 10 accessed pages is the most common approach. Having identified these pages, their accessibility through the navigation structure is to be considered, and if necessary, restructuring introduced or shortcuts placed on index page for easy information access.

It should be also considered that new and returning visitors may have different top *N* accessed pages sets. While detecting the top *N* pages set, each of the pages should also be assigned a weight to show its significance and allow its comparison to other identified pages of interest in the set. It is advisable to exclude the index page, as it commonly gets the highest count of hits. The weight for each page is calculated as the relative share a page has in the top items set. Figure 4-6 shows an example of top 10 accessed pages for the DCE website in two categories: new and returning visitors. Each of the items in the set is also equipped with calculated weight value. Despite the assigned weights pages rank in different order for these categories due to different preferences of returning and new visitors, although the first two items in both lists match despite their different weights. Still, all the items are present in both of the lists.

Although '*Top N pages by popularity*' provides a list of mainly accessed pages, it suffers from not being able to detect the users' real interest towards the pages clicked. This is the reason why the author advises to use '*Top N pages with TSP*' instead. Obviously, applying time spent on a page as an interest indicator to top *N* pages enables to filter out page requests where users have not paid attention to the page itself and just passed through it. This provides yet another method to lessen the rate of bogus hits. The applied TSP values should take into account the minimum time necessary for a visitor to decide over the suitability of page content as well as the approximate time to go through the content presented on a particular page. The possible value ranges have already been discussed in Section 4.2 of this thesis.

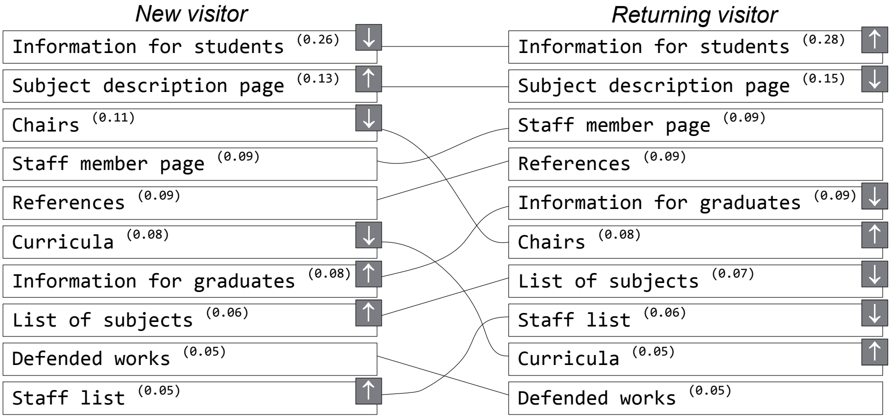


Figure 4-6. *Top 10 pages of popularity for the DCE website in two categories: new and returning visitors. Trend of order in comparison to the adjacent category is shown with arrows, whenever the calculated weight (shown in superscript) in the set differs.*

To investigate the impact of having TSP included as an interest indicator to eliminate bogus hits, an empirical study was carried out applying different threshold values of TSP, and the data from the DCE website usage log with the following values of TSP threshold t_x (Equation 4-3):

- t_0: no threshold set,
- t_1: T_{min}=5s, T_{max}=60s,
- t_2: T_{min}=5s, T_{max}=60s,
- t_{2a}: T_{min}=17s, T_{max}=120s,
- t_{2b}: $T_{min}=0.5 * t_{hyp}(p)$, $T_{max}= t_{hyp}(p)$,

where t_{2a} and t_{2b} conform to equation t_x, with x=2, and $t_{hyp}(p)$ is the hypothetical time needed to read through the content on page *p* on the assumption of average reading speed of 180 wpm. The experiments concentrating on top 20 pages showed that taking TSP into account greatly affects the significance (weight) of each item in the set and can even cause items to be replaced in this list. Table 4-2 outlines the amount of changes introduced to weights of pages of popularity when applying TSP threshold values t_0, t_1, t_2, t_{2a}, and t_{2b}. The study revealed that when using a variable TSP threshold like t_{2b}, pages where visitors stayed for a very short time got discarded, which is a major drawback. Namely, it rules out exit pages as usually their TSP remains unmeasured, unless advanced mechanisms that have lately become available are applied at data logging stage to determine users' commitment at those pages. A workaround could be to exceptionally accept exit pages in the results without valid TSP values. Also, returning users are not probably going to read the whole content article rather than search for a part they previously found interesting. Thereby, lower values of minimum threshold should be applied, leading to a conclusion that threshold conditions t_2 and t_{2a} provide better results. An outtake of the study with a thorough list of top 20 popular pages for DCE website in this study together with changes in the weight rank order is provided in Appendix B.

Table 4-2. *Amount of changes introduced to weights when applying TSP as an interest indicator to pages of popularity listing. Results of the TSP impact study performed on the DCE web usage log data.*

N	**TSP threshold condition t_x**				
	t_0	**t_1**	**t_2**	**t_{2a}**	**t_{2b}**
Changes in top 10	0.0%	20.9%	26.0%	35.8%	53.4%
Changes in top 20	0.0%	30.2%	33.3%	51.9%	64.8%

As shown, exploiting TSP as an interest indicator and using it over the top pages metrics relevantly affects the order of items in this set, and may even cause new pages of popularity to turn up. Thereby, the use of TSP together with page popularity is justified, and the metric '*Top N pages with applied TSP'* valid.

Figure 4-7 describes the results of the top *N* pages with applied interest for the DCE website. The range of TSP to indicate user interest towards pages was chosen from 5 to 120 seconds, covering the threshold conditions t_2 and t_{2a}. As can be seen, the items in the new visitors category remain the same as with the usual ranking (Figure 4-6). However, in the category of returning users many new items gain significance enough to be included in the top 10 pages of popularity, pushing out several items from the accompanying list. In this particular case, more general pages such as *Curricula*, *Chairs of the Department* and *References* to other resources outside the DCE web domain are replaced by more specific pages describing topics related to research and thesis defence procedures. This clearly refers to a fact that returning users have rather specific informational needs and expectations towards a site. While comparing those two top 10 lists to a list where new and returning visitors are not distinguished, it became clear that the particular distribution of new and returning users for a site has a strong impact on what pages are included in top *N* lists and that either of these two categories can mask some of the results with their prevalence. For example, the proportion of new visitors over returning visitors for the DCE website is correspondingly 61% and 39%. Obviously, the counted page hits are in favour of new visitors and thus some preferences of returning visitors can be masked and disappear from the results. A vice versa situation has an opposite effect. Thereby, exploring top *N* page hits with applied TSP should always be presented in two separate categories allowing to discover popular pages of interest for both: new visitors as potential returning users, and returning visitors as loyal users. The results could also be applied for a sort of indirect personalization, where different shortcuts are shown for new visitors and returning users. Still, in terms of general improvement, an intersection over the two latter categories of the top visited pages should be considered.

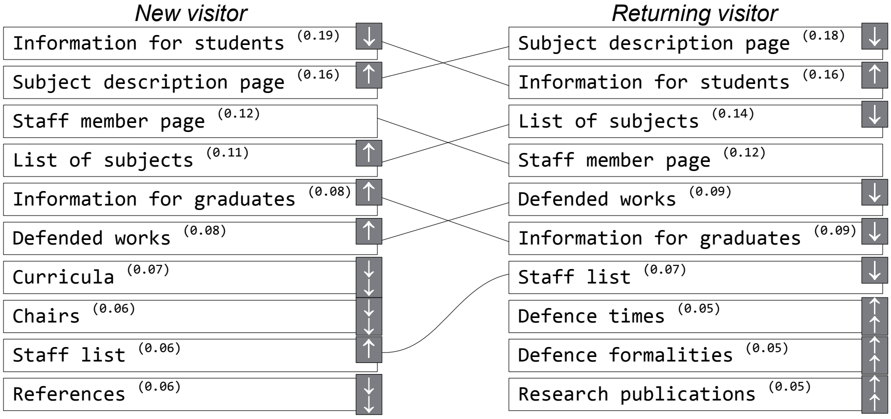

New visitor
Information for students (0.19) ↓
Subject description page (0.16) ↑
Staff member page (0.12)
List of subjects (0.11) ↑
Information for graduates (0.08) ↑
Defended works (0.08) ↑
Curricula (0.07) ↓↓
Chairs (0.06) ↓↓
Staff list (0.06) ↑
References (0.06) ↓↓
Returning visitor
Subject description page (0.18) ↓
Information for students (0.16) ↑
List of subjects (0.14) ↓
Staff member page (0.12)
Defended works (0.09) ↓
Information for graduates (0.09) ↓
Staff list (0.07) ↓
Defence times (0.05) ↑↑
Defence formalities (0.05) ↑↑
Research publications (0.05) ↑↑

Figure 4-7. *Top 10 pages of popularity for the DCE website with applied interest (TSP) for new and returning visitors. Trend of order in comparison to the adjacent category is shown with arrows, whenever the calculated weight in the set differs. Pages falling out or being introduced to the list are marked with a double-arrow trend icon.*

The next metric in the framework in this category to discuss is '*Pages with low TSP values in respect to content length*'. In Section 4.2 it was already shown that users tend to pay more attention to pages with shorter content, i.e., long pages are not fully read. This metric assumes that the data about content length is available. Now, of interest are pages where users hypothetically read only a small fraction of a page (ca 10% or less), rising a question why is it so and is there a good reason for such a page to exist in the WIS at all. Is the reason in lack of interest or is the page content too long to go through, or does it need restructuring? For example, on the DCE website a page describing the rules for using the computer classes was one of those, where only a portion of 9% of content was apparently read. Obviously, a long description of rules does not interest a lot of people. Empirical studies on the DCE website have shown that for short articles the hypothetically read content amount was 86% whereas for longer articles it was only 30%, with the overall content reading rate at 63%. In this context, new and returning visitors behave alike, as the differences found in the study in general were approximately 1% only. However, at detailed page level the differences may vary a lot. If a page of low TSP value per content length is identified, it is advisable to look into the access differences of new and returning users. If there are no significant differences, it reflects disinterest towards this page or a need for content restructuring. However, in the opposite case other metrics should also be considered to further investigate the reasons and proper actions to take.

Large web information systems incorporate hundreds of pages which all should gain some access from users, if made available. Nevertheless, there are almost always pages with extremely low access rate or no access at all. The metrics '*Pages existing in web information system without gaining any access or significantly low access rate from users or web crawlers*' target to identify such pages. For both of the metrics the following parameters are to be found: (a) hit count during a predefined period of time, (b) number of days in this period of time, (c) monthly hit rate, and (d) access ratio according to Equation 4-4, where h is the hit count during period T, t_T is the length of the period in days, and t is the time since the object was last accessed, measured in days.

$$r = \frac{30h}{t_T} \cdot \frac{1}{t} \tag{4-4}$$

Having identified pages users do not access, the next step is to investigate the reasons why they are not accessed. Questions like are these pages accessible at all, accessible through navigation menu or other pages, how the access is provided and so on need to be addressed? The actions concerned with these pages can be either to improve access to these pages, or if they turn out to be redundant, they may be removed from the system. The decision is to be made by an expert of area, i.e., the webmaster.

In terms of web crawlers, the similar metric helps to identify pages that are not reachable by web robots and thereby also potentially not indexed making them unavailable through search engines. Once again, a webmaster has to investigate every case and establish the circumstances why a particular page was outlined by the metric. If a page is purposely excluded from being indexed, this metric highlights that this exclusion is operable. However, in the opposite case actions to improve SEO friendliness such as refining HTML meta-tags, page title and keywords as well as including the page in sitemap for robots are to be considered.

The metrics in the category '*Pages of Interest*' concentrate on single page views for pages described in web information system and users' interests towards those, indicated by hit rate and time spent on those pages, providing one of the methods for WIS evaluation for improvement.

4.3.4 Void Operations in Navigation Paths

During their web sessions, users click on links and browse from page to page. The navigational traces they leave behind may contain repeated identical operations, i.e., for some reasons users have requested the same page several times without leaving it or still being on a page previously read – in other words, there is no progress moving forward through the navigational schema. Using operation timestamps stored in the web usage log, it is easy to sort out accidental double-clicks and rule out probable page refreshes. This produces a set of void operations – repeated page requests that should not be present in reconstructed user navigational traces in any of the planned surfing flows through the site. The processing of users' session data for detecting void operations is shown on Figure 4-3.

The level of void operations reflects the site's usage efficiency, and thus is considered one of the metrics in the WIS evaluation and improvement framework under discussion herein. Evidently, a low or zero level of void operations depicts a clear website structure easily understandable by users. The evaluation is provided through the following indicators:

- The difference between the actual navigational path and the minimized path (void operations removed),
- Average number of void operations,
- Attainable time-saving.

Thus, the indicators address the overhead visitors have been facing while browsing a website and the probable optimization achievable.

For void operations evaluation three different possibilities for minimizing the navigation paths and detecting redundant operations were explored. In other words the task was set on to find such p_i in $s=<p_i, p_{i+1}, ..., p_n>$ where $p_i=p_{i+1}$ and TSP(p_i) satisfied the threshold condition t_x according to Equation 4-3, where t_{TSP} is time spent on p_i. The values for $T_{\min}$ and $T_{\max}$ were declared correspondingly 5 and 60 seconds. Thereby, the threshold condition t_0 results in fully minimized navigation path, t_1 concentrates on intentional re-requests (as page reloads have been filtered out) and t_2 concentrates on the timeframe, where user should have been engaged with page content. An alarming situation is an unreasonably high level of void operations in session, which is a clear mark of fuzzy navigational logic in need of improving. The metric presumes that once a page is requested a user most probably will not to try to access it again in a certain timeframe. Yet, as experiments have shown, this holds for an ideal case.

The experiments with the web usage data of the DCE website (Table 4-3) revealed that returning users had less void operations in their sessions, in average 0.29 operations while the rate for new users was at 0.34, according to t_1. The discrepancy between the level of void operations in new and returning visitors' sessions was probably due to returning users being somewhat familiar with the site. A void operation per session is an averagely good result. The average amount of void operations in sessions was satisfactory low, although the higher rate in new visitor sessions could refer to small problems of easily understanding the navigational structure at once. However, this would require an in-depth study into the most frequent void operation patterns and close evaluation by a domain expert, taking into account the content of particular page itself, to identify whether there is a necessity for improvement. Out of the studied sessions (269 782), only 10.7% contained void operations in this case.

***Table** 4-3. Results for the void operations study on the DCE website over sessions where such operations were identified.*

Threshold condition t_x / Metric	New visitor			Returning visitor		
	t_0	t_1	t_2	t_0	t_1	t_2
Average session length [sec]	374.9	837.9	493.1	308.1	482.7	317.9
Avg. session length for minimized path [sec]	231.4	830.8	194.5	240.4	477.1	235.2
Avg. time saved per session through minimization [sec]	143.5	7.1	298.5	67.7	5.7	82.7
Avg. void operations in session	1.4	0.3	1.1	0.8	0.3	0.5
Average % of void operations	38.0	6.5	31.8	17.9	4.5	13.7

4.3.5 Multi-Sequential Identical Operations in Navigation Paths

The metric of multi-sequential identical operations, as the name implies, is a subtype of the void operations metric concentrating on actions which are identical and made in the timeframe defined by threshold condition t_2. Multi-sequential operations are defined as requests to pages where the following conditions hold: $s = <p_i, p_{i+1}, ..., p_n>$, and $p_i=p_{i+1}=...=p_{i+m}$, and $m<n$. Thus, operations made during normal page views when ideally there should not be any identical sequential operations, are of interest. In case the time between two operations is less than 5 seconds (T_{min}) the page was probably not accurately assessed by the user, and if exceeds T_{max}, it was probably reloaded or revisited. These operations belong to the domain of threshold t_1 and are not of interest herein. The processing of users' navigation paths for identifying multi-sequential identical operations is described on Figure 4-3.

If there is no obvious reason (e.g. automatic reload) for users to re-request the same page within a short period of time defined by threshold condition t_2, it can indicate website structure to be difficult to comprehend. However, the hypothesis of users forgetting where they have just clicked cannot be ruled out on log-based studies without having any explicit feedback from users. For the DCE website 33 158 sessions with at least one void operation were detected based on threshold conditions t_0 and t_2, and studied in further. The results of the analysis are listed in Table 4-4.

Once again, the results confirmed that returning visitors make less redundant operations also in the series of void operations. The percentage of sessions containing redundant operations is low – below 4% for new and 1.3% for returning visitors' sessions by condition t_2 – yet the difference is somewhat distressing on the side of new visitors. Nevertheless, the average time between two sequential operations refers to pages being viewed normally. A small quantity of sessions with sequential operations is inevitable and having no such operations over all the users' sessions is highly unlikely.

The efficiency of this metric is evaluated through the number of multi-sequential operations in users' sessions (the less the better), the time between those operations in sequence, and through comprehensive analysis of the most common navigation patterns where such sequences have occurred.

Table 4-4. *Characterization of multi-sequential identical operations for sessions containing such operations. A study on the DCE website.*

Metric	**New visitors**	**Returning visitors**
Sessions with multi-sequential identical operations (t_0)	12.8%	3.3%
Sessions with multi-sequential identical operations (t_2)	3.9%	1.3%
Avg. number of identical operations in sequence (t_0)	3.3	2.8
Avg. number of identical operations in sequence (t_2)	2.8	2.7
Time between identical sequential operations (t_0)	607.6 s	458.9 s
Time between identical sequential operations (t_2)	24.4 s	21.0 s

4.3.6 Cyclic Operations in Users' Navigation Paths

This metric aims to identify the rate of cycles in users' navigation paths that renders the efficiency of website usage. High level of short cycles refers to navigational fuzziness and rapid browsing. Unplanned round-connected page-sets in visitors' navigation paths are a clear mark of too confusing or difficult navigational schema for users to comprehend. Obviously, the shorter the cycle is, the more problematic it becomes as this is not an efficient way of seeking information and it is likely that users have lost their way in the website structure. Thus, there might be a necessity for restructuring. Nevertheless, cycles my also occur when users are returning to previously visited page to remind the content presented there. If the studies of user behaviour show visitors to take this path frequently, content presentation should be revised. Users are lazy and take shortcuts whenever possible and that is what can be learned from their navigation paths herein.

Each navigation path can contain zero or more cyclic profiles. The concentration of cyclic localities for websites with problematic navigation structure is higher than for sites with easy navigation. The efficiency of web browsing with this metric is measured through the following indicators:

- The size, frequency and rate of cyclic locality profiles in sessions,
- Time spent on pages involved in cyclic profiles,
- The number of operations needed for a cycle to occur.

The processing of users' session data for identifying cyclic operations starts with navigation path minimization, which is followed by the application of the locality model on the minimized navigation path (Figure 4-3). After the localities have been extracted, each of them is studied and matched against the following condition: $L(w) = p_j, p_{j+1} , \ldots , p_m$, where $p_j = p_m$, which holds true only for cyclic localities. In the experiments described in this section, various window *w* sizes were used. Still, of particular interest were the results for w=3, as it is the smallest set of operations to form a cycle, and also the locality size w=3 matches the best for the DCE website, as has been proven earlier.

In the experiments with the DCE web usage data, cyclic locality profiles were found in 33 813 cases out of available 269 782 web sessions. The study involved localities of various sizes, beginning with w=3 and ending with w=10. This range was selected in order to compare the results and observe trends, although in practice the emphasis should be on short cycles covered by locality profiles of size w=3.

The distribution of cycles in locality profiles of different size is shown on Figure 4-8. As can be seen, the number of cyclic localities in sessions occurred to be a bit higher in new visitor sessions; and as expected returning users had fewer cycles in their navigation paths, due to probably being familiar with the site. The rate of cyclic operations in sessions for the DCE website is low. The frequency of cyclic profiles in sessions is clearly in inverse proportional relation to profile size *w*. Obviously, short cycles are problematic as this is not an efficient way of seeking information, implying users to be lost in the website structure. Without explicit studies the possibility of users returning to previously visited page to remind the content presented there cannot be ruled out. Nevertheless, if such cycles are frequent, they need to be reviewed by a domain expert and actions of structure and content improvement carried out.

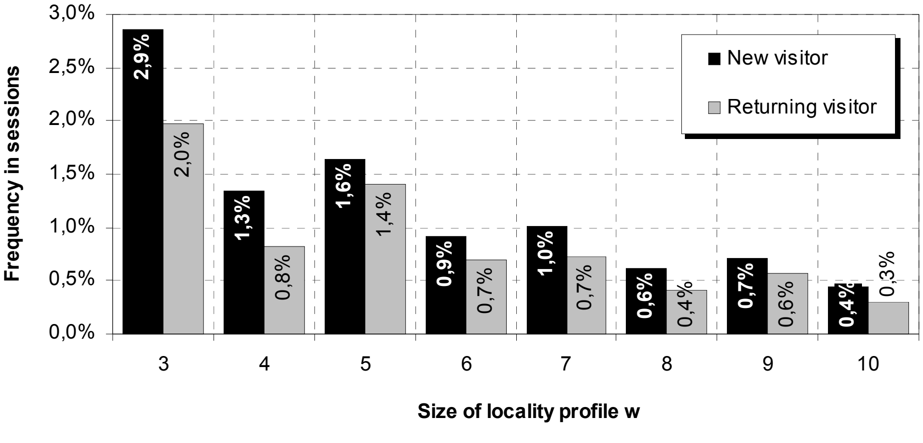


***Figure 4-8.** Rate of cyclic localities in users' sessions.*

The number of operations needed for a cycle to first occur in navigation path also reflects navigation efficiency. Obviously, the higher this number is, the better is the performance. Experiments with the web usage log collected on the DCE website (Figure 4-9) showed that this number is most certainly bigger than the size of the locality itself. In the worst case scenario the number of operations needed for cycle occurrence equals the cycle size. For sessions, which had at least one cyclic profile in it, the reference value was 9.2 operations needed for a cycle to occur. For the shortest and most important cycle (w=3), the average was 23.2 operations per cycle occurrence, which is almost 8 times more than the cycle length itself, and can be considered a good result. Timing analysis of cycle length showed that interestingly cycle duration is in favour of returning visitors due to longer page views, with average time spent on page belonging to a cycle evaluating to 20.4 seconds.

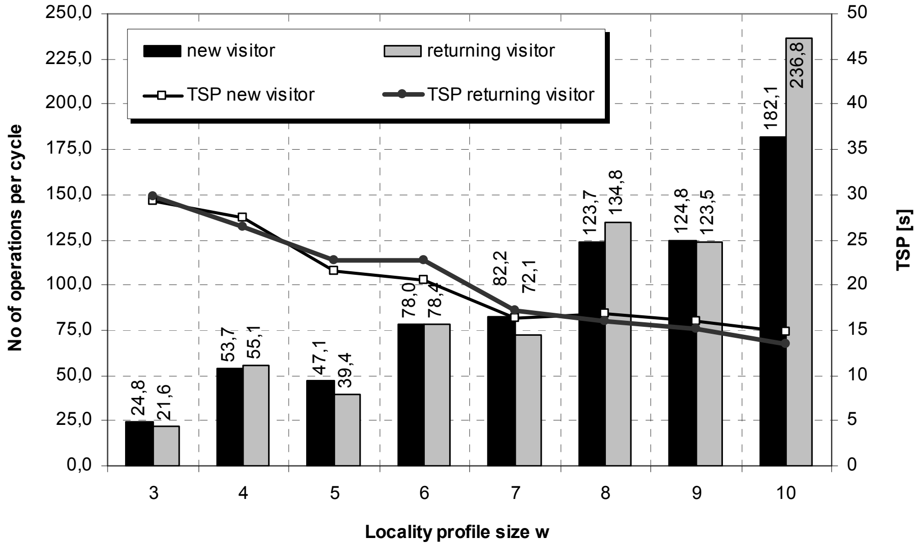


Figure 4-9. *The number of operations needed for a cycle to occur and the time spent on pages belonging to cyclic localities.*

When it comes to identifying troublesome cyclic operations, the author advises to use ranking based on cycle frequency and time spent on pages involved in cycles. Eliminating cyclic operations where at least one of the pages has high TSP value enables to concentrate on such cycles where the effect of user distraction is fairly low. The ranking of the remaining sets of cycles can be achieved by the use of formula given by Equation 4-5, where t_{age} represents time in days since the cycle occurred and was recorded in the log (this way recent users' behaviour has more important role in the formula and new trends are considered better), t_{TSP} the time spent on a page involved in a cycle, w is the cycle length and $f(i)$ is the frequency rate for such cycle to appear.

$$Rank_i = \sum_{i=1}^{n} \frac{f(i) * \sum_{j=0}^{w} t_{TSP}(i)}{t_{age}(i)} \tag{4-5}$$

Applying ranking of such localities enables to filter out the most significant cyclic operations in users' navigation paths for their further evaluation by domain expert, i.e., the webmaster and introduce actions of improvement, if needed.

4.4 Chapter Summary

While browsing websites, users follow certain patterns of behaviour according to their view of the subject domain being explored. Collecting web usage log and investigating users' behaviour on website through their navigation paths can reveal a lot about site's usability.

This chapter discussed a framework for evaluating web information systems on the basis of user behaviour modelling. A set of users' interest indicators was introduced and several metrics relying on these proposed. These metrics, based on collective intelligence and represented by users' behaviour on a website allow to identify a necessity for improving a web information system and provide a basis to minimize the gap between the conceptual model applied during development and the model employed by the actual site visitors – allowing to optimize WIS to its users. In practice, the results provided by the metrics must be interpreted by an expert of area (i.e., webmaster), identifying the nature and criticality of each of the problem based on the aim of that particular website.

The framework does not provide a 'silver-bullet' solution to address all the problems, and usually designers and developers are competent enough to avoid large errors. Nevertheless, they have no abilities to predict actual website usage, its changing trends and users' with their needs over time. Thus, evaluations based on users' behaviour modelling are needed to improve web information systems.

Chapter 5
LEARNING WEB USERS’ DOMAIN MODELS FROM BROWSING BEHAVIOUR

The information overload on the World Wide Web has evoked a necessity for adaptive, personalized and intelligent web to assist users while performing their browsing tasks. This intelligence for adaptive web information systems is based on user profiling and exploitation of user models. When browsing the web, users have an implicit conceptual model of the domain in their mind. This model is based on their knowledge of the domain and as a rule does not entirely match to the given website topology and domain model applied by developers, introducing a conceptual gap between these models.

This chapter addresses the latter mismatching problem and provides a solution by making web users’ domain models explicit by using ontologies created on the basis of user profile mining on the web. A method combining web usage mining and ontology engineering techniques to learn user profile ontologies based on user behaviour is established. This method is applicable for detecting online visitor type and for use in recommender systems to adapt web information systems to the needs of individual users or user groups.

5.1 Introduction

How users are browsing and searching the web is heavily dependent on their own conceptual model of the subject area rather than the domain model and website topology given by web developers. Due to this, there exists a well-known problem of possible mismatch between web users’ domain model and website topology. Various approaches have been provided to address the problem covering research in the areas of web users profile mining, personalization and adaptive web, and use of domain ontologies [13], [20], [72], [73], [87], [103], [115], [116], [117]. Still, the mismatch problem haunts researchers and urges them to develop new methods to understand and better match users’ personal views to the domain. A general solution to this problem is seen in providing a kind of personalisation service.

The majority of personalization and recommender systems take advantage of user profiles. User profiling is an essential component of any adaptive and intelligent web system targeting recommendations and web personalization and thereby also the conceptual gap introduced by the mismatch problem. A user profile explicitly represents the properties of an individual user or user group and allows systems to distinguish between its users. Commonly, researchers have applied predefined user ontologies in their systems instead of learning user profile ontologies.

In this chapter, a method for learning user profiles from web users browsing behaviour and underlying web ontology is proposed. The method concentrates on modelling an anonymous ad-hoc web user, which does not diminish its usability on identifiable users. The main contributions for the community delivered herein is a method of learning user profile ontologies from web usage patterns, and the modelling of an anonymous ad-hoc web user, rather than an identifiable web user. As the method is focused on anonymous users, it cannot assume any prior information to be available about users, neither to have them to identify themselves.

Following the method, user profiles are constructed based on users' browsing behaviour data accumulated from web usage log onto which a concept of the locality model is applied to extract user preference profiles. Ontology reasoning services are used to classify these preference profiles under predefined user profiles concepts. This allows specifying rather general user profile concepts according to mined user preferences giving as a result a definition of domain ontology of users. The novelty of the method lies in giving conceptual meaning to web usage mining results by using ontologies and automatic classification of concepts to ontologies via ontology reasoning as a part of the system. The learned users' domain models are applicable in providing personalization services and to improve the accessibility of information to certain user groups with predefined set of interests.

In order to create web users' domain models, two initial ontologies are required beforehand: a web ontology and a predefined user profiles ontology, which are described using the Web Ontology Language[1] (OWL), in particular OWL DL (Description Logic variance of OWL) in order to support automatic reasoning. These ontologies are constructed manually and in this case the author has used the Protégé[2] ontology editor for the task. Protégé is a free open-source platform providing a suite of tools to construct domain models and knowledge-based applications with ontologies. In principle, there are no limitations on editors to be used.

In addition to predefined concepts (classes), users' profile ontology contains a class for extracted user profiles which are automatically obtained from web usage log through original data mining and processing steps. Based on semantic annotations of web ontology, OWL definitions of corresponding concepts are generated automatically and added to the user profile ontology OWL

[1] http://www.w3.org/TR/owl-features/

[2] http://protege.stanford.edu

description file as definitions of subclasses of extracted profiles. In the following ontology reasoning is used and extracted user profile concepts are classified under predefined user profile concepts. This process gives as a result an ontology definition for users' domain model.

The following sections of this chapter describe these steps for attaining web users' domain model in more detail. In Section 5.2 the process of extracting user preference profiles from web usage logs by applying the locality model is described. Section 5.3 discusses web ontology construction, while Section 5.4 concentrates on user profiles ontology, covering predefined user profiles and the process of attaining extracted user profiles from user preference profiles via semantic annotation. Section 5.5 is dedicated to ontology reasoning and establishment of the users' domain model. Section 5.6 provides a brief discussion on the attained model, and Section 5.7 summarizes the chapter.

5.2 Extracting User Preference Profiles From User Behaviour Data

The construction of web users' domain model starts with processing web usage log data to extract users' preference profiles. These profiles render users activities into their probable interests and form the basis of user profiles in the corresponding ontology.

To extract user preference profiles, web users' browsing sessions are acquired from the web usage log and navigation paths reconstructed. The sequence of operations in the path is determined by operation timestamps available in the log. The constructed navigation paths are then fully minimized, thus any redundant operations in the path are removed. Onto these minimized paths an approach of the locality model is applied and user locality profiles of size *w* are extracted. These profiles are further processed to eliminate cyclic localities as these do not carry any value of progress in site browsing (users end up at the same page they started on). Figure 5-1 describes the process of extracting users' preference profiles in detail with example data from the DCE website. In the examples locality window size w=3 has been used, as it performs the best for the site as has been previously proven in Section 4.3.2. During the process 87 953 navigation paths were processed.

Altogether, assuming that user browsing session is defined as a sequence of accessed pages $s{=}{<}p_i,\ p_{i+1},\ ...,\ p_n{>}$, where $s{\in}S$, $p_i{\in}P$ and S is the set of all the processed sessions and P is the set of all the pages available in WIS, localities L of size w fulfilling the conditions $L{=}p_j,\ p_{j+1},\ ...,\ p_m$, and $p_j \neq p_m$, are extracted, where p_j is a visited page ID. For each $s{\in}S$ a function *L=ExtractLocalitiesUPP(s,w)* is applied for user locality profiles extraction (Figure 5-2).

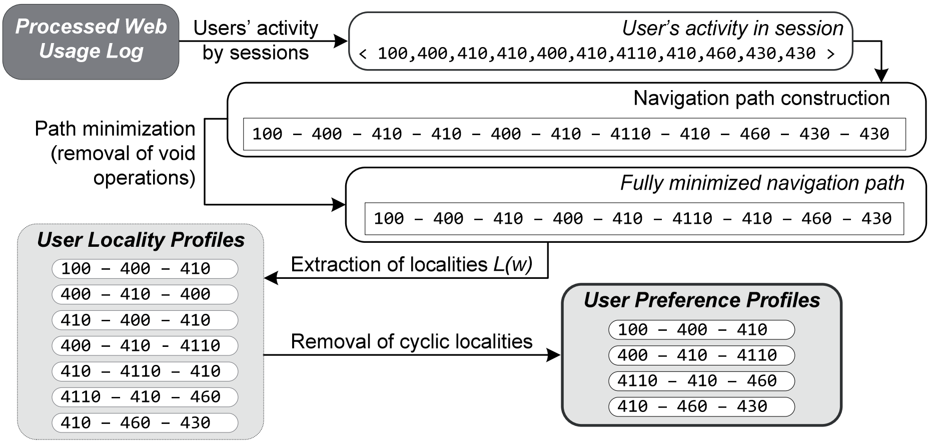


Figure 5-1. *Processing of users' session data to extract user preference profiles.*

```
Function ExtractLocalitiesUPP(s, w)
    /*
        s: array                  = operations in user session being processed
        w: int                    = locality window size
        LocalityProfiles: array   = set of extracted localities
        runningLocality: array    = locality being extracted
        quantity: int             = number of localities extractable from path
    */

    BEGIN
        quantity = count(s) - w + 1
        i = 0
        while i < quantity do
            j = i
            while j < w + i do
                add s[j] to runningLocality
                increment j
            end while
            if runningLocality[0] <> runningLocality[w-1]
                add runningLocality to LocalityProfiles
            end if
            reset runningLocality
            increment i
        end while
        return LocalityProfiles
    END
```

Figure 5-2. *Algorithm for user locality profiles extraction based on the locality model.*

Having obtained the set of user locality profiles, the next step is the elimination of infrequent localities that are probably the result of random browsing. Thus, of interest are only frequent localities, as they represent sets of related information from the point of view of web users.

In web information systems, where changes to structure and data are rarely introduced, ranking of extracted user locality profiles does not influence the frequent localities. Experiments with web usage log data for the DCE website showed that regardless of the period length, the set of frequent user locality profiles remains almost the same for the top 10% of frequent localities (differences up to 2% at most). The long term analysis with data collected over 5 years showed a difference up to 15% and for the short term (1 year) the difference was not over 10%, when comparing the sets of extracted localities with and without ranking. In both cases the maximum difference occurred at the end of the list, where the significance of locality compared to the top of the list was already in great decline. Thereby, ranking is important only in case the underlying WIS undergoes changes regularly or has gone it through lately, as then and only then the changes get more attention by ranking. As the DCE website is considered moderately changing, for each user locality profile a rank value was calculated according to Equation 5-1, where *f(i)* is the frequency rate of an extracted locality over a time period *t*, t_{age} represents the age of the locality since time *t,* which is measured in units of months as a reasonable step for trends to emerge. Other units of age are also possible, depending on the log size, WIS access rate and frequency in which changes are introduced. For the DCE website, the locality age was calculated in months.

$$Rank = \sum_{i=1}^{n} \frac{f(i)}{t_{age}(i)} \qquad (5\text{-}1)$$

The process results in a set of user preference profiles. In further these profiles will be semantically annotated. The discussion of this data processing is continued in Section 5.4.2, where user preference profiles are used to construct the ontology of extracted user profiles.

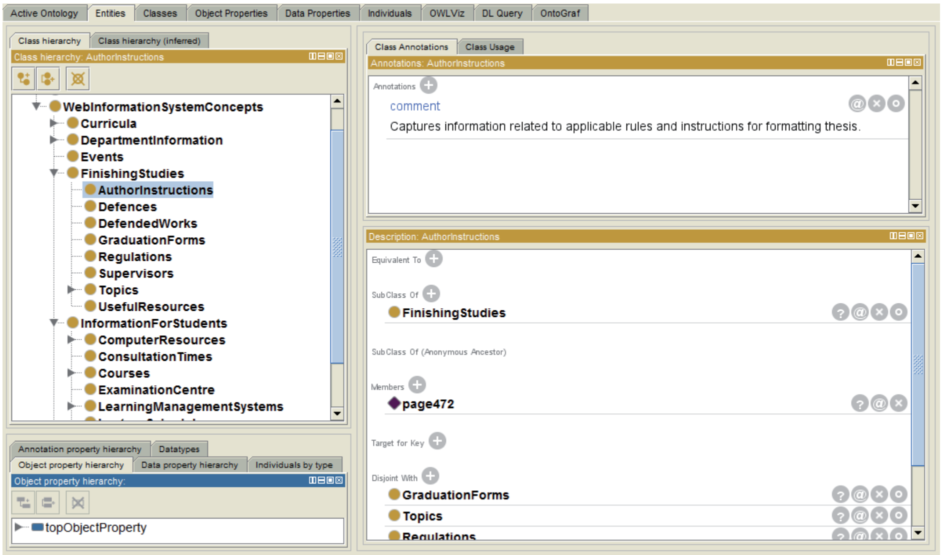
5.3 Web Ontology

The method described in this chapter assumes two initial ontologies to exist prior to its execution. These are web ontology and predefined user profiles ontology.

Web ontology is used to define the set of concepts captured by a web information system as well as relationships between those concepts. This ontology is manually created by a domain expert. While semantic portals rely on domain ontologies to structure knowledge, web ontology takes another approach and is created from the point of view of the actual web application and how its potential users experience it. Obviously, it is not possible to create one and final web ontology that entirely matches conceptual interests of WIS users in advance, mainly due to the following two reasons. Firstly, users and their behaviour are not known in full in advance and these are changing over time. This may cause a necessity to introduce modifications to WIS. Secondly, web information systems also change, information is added, updated and removed, which also influences users' behaviour. Thereby, web ontologies also need to evolve together with WIS's.

The aim of web ontology is to cover the information domain presented on and made available through the WIS. This is the basis for creating classes of the web ontology. The concepts described in the web ontology are used in defining user profiles ontology, as will be discussed in further sections of this chapter.

Figure 5-3 shows an example of the web ontology created for the DCE website. The concepts in the web ontology are defined as primitive classes and organized as taxonomy. In this ontology, for instance, classes to capture information related to the department, studies, and graduation procedures are defined among with other concepts. Each of these classes can have subclasses. The concepts captured with the web ontology are declared as subclasses of a superclass named as *WebInformationSystemConcepts*. The classes are annotated with comments and equipped with at least one representative of a corresponding webpage in the WIS added as an individual. Each of these individuals is annotated with comments and with an additional property *hasURL* referring to the actual page URL on the World Wide Web. This becomes useful later on when building a metadata database. Classes in the taxonomy are made disjoint whenever applicable, meaning these concepts are not overlapping and their individuals are assigned to be member of one class only. Appendix C provides an overview of the web ontology created for the DCE website.



Captures information related to applicable rules and instructions for formatting thesis.

Figure 5-3. *An example of the web ontology created for the DCE website (screenshot from the Protégé ontology editor).*

5.4 Establishing User Profiles Ontology

In addition to web ontology, user profiles ontology is created, where two types of profiles: predefined user profiles and extracted user profiles are presented. The latter is based on the results of web usage log mining, while the first set is added to the ontology manually. User profiles ontology is defined as superclass of extracted and predefined profiles classes.

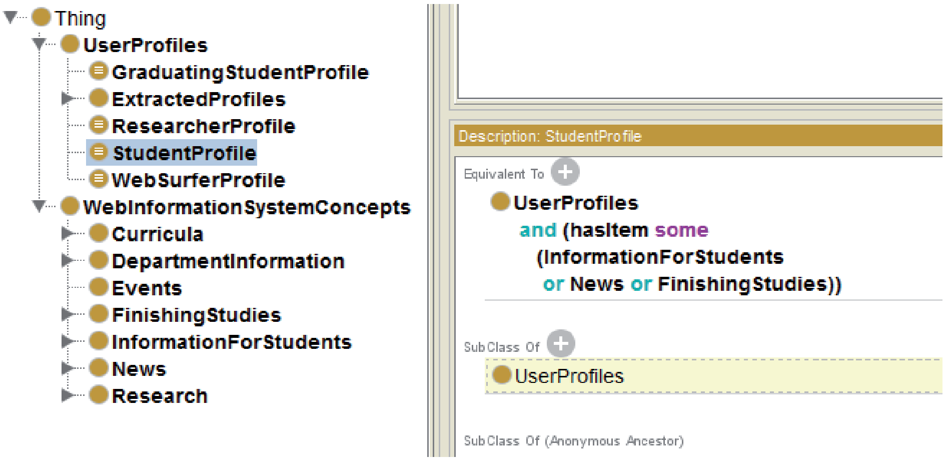
The classes in the user profile ontology link the user profile ontology concepts to the web ontology concepts via the *hasItem* object property. The *hasItem* property represents relationship between user profiles and web ontology concepts. Using this property, classes of user profile that have items from some classes from the web ontology are defined. The *hasItem* property is transitive and its domain is the class *UserProfiles* and its range is the class *WebInformationSystemConcepts*.

5.4.1 Predefined User Profiles

Predefined user profiles are declared as concepts that capture rather general user profiles and describe typical preferences of some well-known types of possible WIS users. These classes are specified as defined, or also known as complete classes, to be able to use reasoning services.

Predefined user profiles are defined using the *hasItem* property, via which certain property restrictions are created. These restrictions specify the conditions items must meet to belong to a particular class. In order to make the class definitions complete, these restrictions are defined as necessary and sufficient conditions. This means that the declared conditions are not only necessary for membership in a particular class but also sufficient to determine that something satisfying these conditions is a member of this class. Thus, if an item is a member of the class, it must satisfy all the declared conditions, and if any random item satisfies these conditions, it must be a member of this particular class. Setting necessary and sufficient conditions for a class and by that declaring it complete is essential for using reasoning facilities to automatically classify extracted user profiles under predefined user profiles.

To exemplify the previous, let us consider a predefined user profile called *StudentProfile* shown on Figure 5-4. The class *StudentProfile,* complete by its definitions, has an existential restriction saying that it has at least one item that is from the class *InformationForStudents* (a subclass of the *WebInformationSystemConcepts* superclass) or from the class *News,* or from the class *FinishingStudies*. The restriction in this example is simple but in general any restriction to predefined profiles can be defined according to the needs.



Thing
UserProfiles
GraduatingStudentProfile
ExtractedProfiles
ResearcherProfile
StudentProfile
WebSurferProfile
WebInformationSystemConcepts
Curricula
DepartmentInformation
Events
FinishingStudies
InformationForStudents
News
Research

Description: StudentProfile

Equivalent To

```
UserProfiles
 and (hasItem some
   (InformationForStudents
    or News or FinishingStudies))
```

SubClass Of

UserProfiles

SubClass Of (Anonymous Ancestor)

Figure 5-4. *Definitions of predefined user profile classes (screenshot from the Protégé ontology editor). A defined class StudentProfile has been selected (on the left) and its definition shown (on the right).*

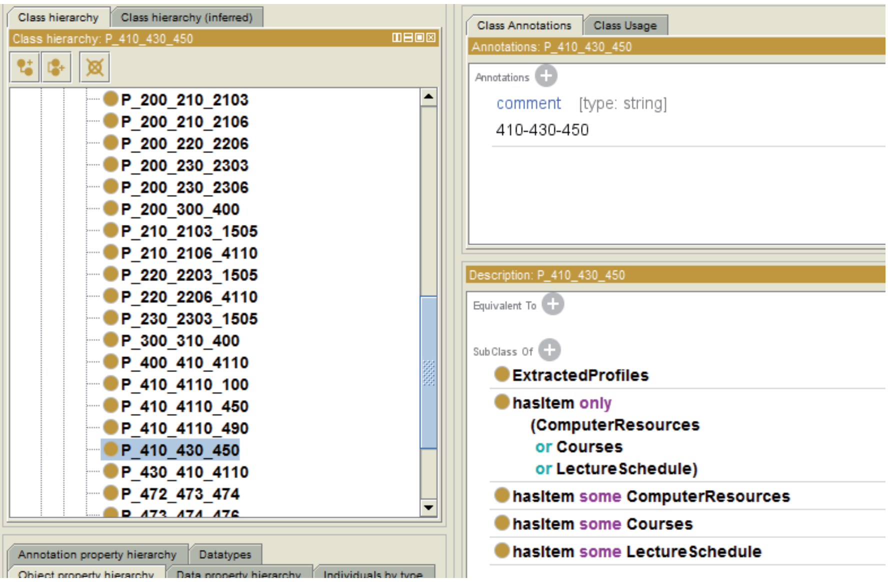
5.4.2 Extracted User Profiles

Extracted user profiles are based on the discovered user preference profiles (Section 5.2) obtained from web usage logs through original data mining procedures. The extracted user preference profiles do not contain any semantic information about the content represented in the WIS. Therefore, they are annotated by metadata, which in further allows to automatically add these profiles into ontology description. The metadata database used for annotation is a collection of data, where pages present in the WIS are referred to the concepts of web ontology by their URI references. This database can be composed manually, or if necessary information has been added to the web ontology during its design (through individuals and their annotation property *hasURL*) derived automatically. For the studies on the DCE website a manually constructed metadata database matching page IDs and web ontology concepts was used.

Using the metadata database, for each of the user preference profile a corresponding OWL class definition is automatically generated and added to the predefined user profile ontology OWL file. The generated classes are defined as subclasses of the extracted profile class from user profiles ontology (class *UserProfiles*). Extracted user profile classes are added to ontology as partial (primitive) classes, meaning their definitions have only necessary conditions.

A closure axiom on the property *hasItem* is specified for class definitions of extracted profiles. This closure axiom, consisting of universal restriction with filler that is union of all three components of a given extracted profile, limits the possible fillers to the defined set. This is necessary because reasoning in OWL is based on open world assumption, which means that it is not possible to assume something not to exist until this has explicitly being stated. Thus, adding closure axioms assures that these classes can only be related to classes defined over property *hasItem*, and not anything else. Closure axioms guarantee that the classification of extracted profiles under predefined profiles will be correct.

For instance, the extracted user profile ‘*P_410_430_450*’ has the following closure axiom specified: *∀hasItem (Courses OR LectureSchedule OR ComputerResources)*. Figure 5-5 depicts the set of predefined user profiles with a closure axiom and definitions shown for the latter profile in the Protégé ontology editor.


Class hierarchy
Class hierarchy (inferred)
Class hierarchy: P_410_430_450
Class Annotations
Class Usage
Annotations: P_410_430_450
Annotations
Description: P_410_430_450
Equivalent To
SubClass Of
Annotation property hierarchy
Datatypes
Object property hierarchy
Data property hierarchy
Individuals by type

P_200_210_2103
P_200_210_2106
P_200_220_2206
P_200_230_2303
P_200_230_2306
P_200_300_400
P_210_2103_1505
P_210_2106_4110
P_220_2203_1505
P_220_2206_4110
P_230_2303_1505
P_300_310_400
P_400_410_4110
P_410_4110_100
P_410_4110_450
P_410_4110_490
P_410_430_450
P_430_410_4110
P_472_473_474
P_473_474_476

comment [type: string]
410-430-450

ExtractedProfiles
hasItem only
(ComputerResources
or Courses
or LectureSchedule)
hasItem some ComputerResources
hasItem some Courses
hasItem some LectureSchedule

Figure 5-5. *Fragment of the predefined user profiles ontology (screenshot from the Protégé ontology editor). Class definition and closure axiom shown for predefined user profile ‘P_410_430_450’ (on the right).*

Automatic generation of the above descriptions is made possible as user preference profiles are semantically annotated on the basis of web ontology. Automatic generation is essential as the amount of extracted user preferences grows fast with each item made available in WIS. Appendix D shows a fragment of generated OWL 2 description of the extracted class ‘P_410_430_450’.

5.5 Automatic Classification of User Profiles

Now that the web ontology and predefined user profiles ontology are described, and extracted user profiles ontology generated, it is time to perform automatic classification of extracted user profile classes. The OWL-DL representation of these ontologies enables the use of semantic reasoners for the task.

With the use of reasoning service new user profile ontology can be derived from the logical definitions in the original asserted ontology. The classification process results in computing an inferred ontology which shows how the extracted user profiles are classified as subclasses of predefined user profile classes. For the reasoning service the Pellet[1] OWL 2 reasoner was used.

The result of the classification task is the specification of definitions of predefined user profile classes that can be considered as web users' explicit model of the domain. By exploring the subclasses of predefined profiles, one can see the concepts constituting users' model of domain. For instance, the predefined user profile class *StudentProfile* was defined in rather general way as seen on Figure 5-4. After classification it is specialized by several subclasses, indicating what concepts from the web ontology really were of interest for users. Figure 5-6 shows a comparison of asserted and inferred ontology for user profile *StudentProfile*. Due to the rather general definition of the class *StudentProfile*, there are quite many subclasses found for that user profile. Also, a predefined user profile *GraduatingStudentProfile* is classified by the reasoner as a sort of *StudentProfile*, which is true. The definition of the *GraduatingStudentProfile* in the ontology is given as follows: *∃hasItem (FinishingStudies OR InformationForStudents) AND ∀hasItem (FinishingStudies OR InformationForStudents)*.

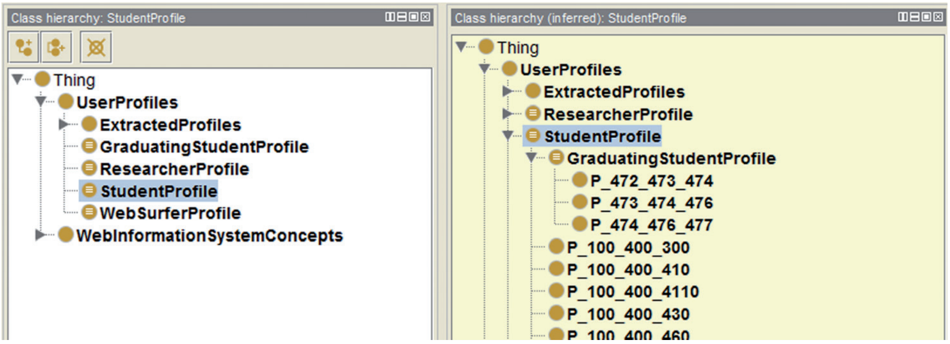


Figure 5-6. *Comparison of asserted (left side) and inferred (right side) user profiles ontology. Subclasses of the superclass 'ExtractedProfiles' in the asserted ontology have been classified under predefined user profiles in the inferred ontology. The reasoner has also affected the structure of predefined user profiles according to class definitions.*

It might happen that not all extracted profiles fall under some of predefined user profile classes. This might indicate a need for introducing new predefined user profiles but not necessarily. When comparing asserted predefined user profiles and inferred ones, differences can be learned and used to improve the web ontology and website structure in order to bring those closer to users' preferences.

[1] http://clarkparsia.com/pellet

The above described approach delivers users' profile in the form of ontology. Thus, the result of reasoning and classification of extracted user profiles is the users' domain model. Since users' preferences are changing over time, the process of learning users' domain model from users' behaviour must be iterative.

5.6 Applicability of the Method

The described method was developed for attaining users' domain model based on web users' behaviour recorded within WIS. The latter model can be used for various purposes. Still, the first and foremost aim while developing the model was to apply it for detecting the type of an anonymous ad-hoc online user, describe such users information needs towards a particular WIS, and explore the possibilities to exploit it as a basis for user profile within some kind of personalization service. Other areas of application may for instance involve web ontology refinement, website structural analysis and improvement (strategic adaptation), and detection of unidentified user groups.

Let us now consider the main aim of the developed method. By tracking and analysing the actions of the current user online and comparing it to the user profiles described in the domain model, it is possible to classify the online user as an individual into one of the conceptual user groups (predefined profiles) and thereby acquire the general model describing that online user and his/her interests. In other words, the predefined user profiles are used to determine the interests domain of an anonymous ad-hoc web user by his/her latest browsing activity. Thus, the method enables to derive a user profile for an ad-hoc anonymous online user. For this, the inferred user profiles ontology needs to be queried for superclasses of the current user preference profile. Having identified user type, it can be used as an input for a recommender system, or applied within a WIS management unit to adapt current web view to detected user type based on simple rules (tactical rule-based adaptation). Figure 5-7 describes the general system architecture of establishing and maintaining web users' domain model, and its exploitation for online user classification.

To detect online user profile, the following steps are taken: tracking of user actions, construction of the corresponding locality profile of size *w*, and mapping of those items to concepts of the web ontology. Having identified the concepts in the locality profile, users' domain model is queried and online user type determined based on the predefined user profiles.

Presently, the users' domain model has no quantity restrictions set on predefined user profiles. This means that the ontology could be queried against any set of concepts for obtaining a user profile. However, this can also result in multiple user profiles identified, especially during online user type transition from one profile to another. In case multiple user profiles are identified, the detection involves additional steps concentrating only on the *w*–1 last concepts available through the locality profile, until w=2. However, it may happen that for a given concept set explicit user classification fails. With more user actions coming available, profile detection is repeated. In the case of the DCE website and its users' ontology it was assumed that user has performed at least three operations (w=3) before user type can be determined.

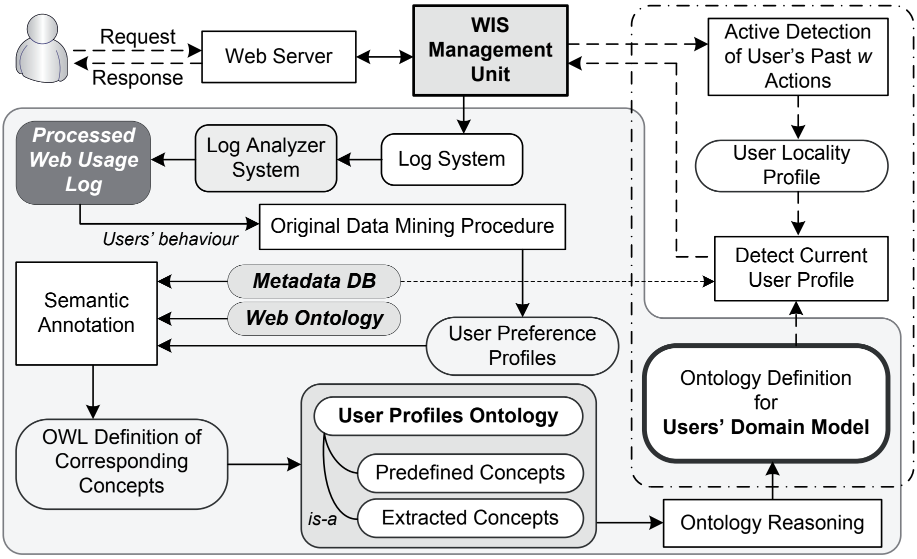


***Figure 5-7**. Overview of system architecture. The flow of learning users' domain model, covered by the method, is shown within the gray area, while the flow for detecting online visitor type is indicated inside the area bordered with a dash-dotted line.*

For example, if a user locality profile is identified as *L={100, 400, 410}*, it matches to concepts '*DepartmentInformation*', '*InformationForStudents*', and '*Courses*' described in the web ontology. The query to obtain the user profile is thereby defined as '*UserProfiles and (hasItem some DepartmentInformation and hasItem some InformationForStudents and hasItem some Courses)*', returning two predefined user profiles, namely the rather general '*WebSurferProfile*' and a specific profile '*StudentProfile*'. Now, as the explicit user classification failed, the eldest member in the locality profile is discarded and the classification procedure is repeated. The query '*UserProfiles and (hasItem some InformationForStudents and hasItem some Courses)*' returns only one user profile, namely '*StudentProfile*', which is also the correct user profile for a given concept set as described in the ontology.

Presently, the development of such an online user profiler tool remains a future work. The task can be solved by using the OWL-API[1], which is a Java API and reference implementation for creating, manipulating and serialising OWL Ontologies, including a parser to read and write ontologies described in RDF/XML and OWL/XML formats.

[1] http://owlapi.sourceforge.net

In Chapter 7 the described method and the users' domain model attained according to it will be used for establishing recommendations for online user.

5.7 Chapter Summary

While browsing websites for information, users have a sort of implicit conceptual model in their mind, driven by their expectations and previous experiences. These conceptual views are partially shared with other visitors and can be reproduced as user models based on collected access data representing web users' behaviour on that website.

In this chapter a method for constructing web users' domain models by using ontologies created on the basis of user profile mining on the web was proposed. The method is based on the original web ontology and user profiles ontology composed by a domain expert. It is assumed that these ontologies are available in advance. Fully automated log system is used to collect data about users' preferences into web usage log. Mining this data, OWL descriptions of extracted user profile concepts are generated and added into user profiles ontology. Using ontology reasoning services available for OWL-DL, extracted user profile concepts are classified under predefined user profile concepts, giving as a result a definition of domain ontology of users. The method described in this chapter has been published in [130].

The main contribution of this chapter is the method for learning user preferences for making users' domain model explicit in order to classify online visitor and provide personalization services. The method also provides a machine-readable ontology of user browsing behaviour, opening new possibilities to explore and exploit users' behaviour data, for instance in recommender systems.

Chapter 6
PREDICTING WEB USERS' ACTIONS

Logging users' behaviour while they search and access information on websites delivers an excellent source to learn users' probable actions in the future. Regardless whether the user is identifiable or not, this is a challenging task as users' intentions, interests, and behaviour differ on the individual basis, and are dependent on particular informational needs. Nevertheless, modelling users' behaviour as collective intelligence enables to predict their future actions, and thereby provides an opportunity to assist users while browsing, to adapt web information systems to users' needs, and to offer personalization services.

In this chapter two different models for predicting users' web activity are proposed, in particular the next page to be viewed. Both of the models take advantage of the implicitly collected web usage log data and the previously introduced locality model to uncover online visitor intentions.

6.1 Introduction

Daily, people use the Internet to search for information, read news, communicate with friends and colleagues. While browsing websites, users follow certain patterns of behaviour, dependent on their approach to the subject area and driven by their informational needs and expectations. While in real world users are able to explicitly state what they are searching for and how they would want to do it, they still lack this possibility in the virtual world. Nevertheless, with each page request they make on the Web they leave behind traces of data referring to their informational needs. Logging these users actions in WIS's can provide valuable knowledge for solving the task of identifying users' intentions and foreseeing their probable behaviour. This is particularly important in sense of adaptive web information systems and systems delivering personalized web experience.

While Chapter 5 focused on profiling and detecting anonymous ad-hoc web user profile type, this chapter concentrates on predicting browsing activity for such a user. Two models for predicting user behaviour are proposed. These models rely on web usage log, collected by the log system and processed with the log analyzer, discussed in Chapter 3. The applied automated and implicit data gathering methods provide data sufficient enough to aggregate the behaviour of the silent majority of web users and enable machines to reason about online users' current intent. Data available in the web usage log is mined and users' navigation paths constructed. Then the concept of the locality model, introduced in Chapter 4, is applied to extract locality profiles and items of further interest based on collective intelligence. This knowledge is then used in the models to predict web users' further actions in the nearest future. Details for each of the models are given in corresponding sections.

This research is based on the web usage data collected from the DCE website. From the web usage log the following data has been used: (1) requested page identifier, (2) user session identifier, (3) set of operations performed during a session, (4) timestamps for each of these operations. To evaluate the proposed models, data from the web usage log as well as from an active real web prediction system has been used. Experiments with the prediction models were carried out with randomly selected data from web usage log, and by exploiting the prediction models in a prediction engine working on the backgrounds of a real website, allowing to compare model performance in real life situations. The prediction models and results of the experiments have been published in the author's works [131] and [132].

The rest of this chapter is organized as follows. Firstly, a probabilistic sequential model for users' action prediction is discussed together with experimental results in Section 6.2, while Section 6.3 delivers the conceptual model for predicting web users' nearest future behaviour together with a comparative study of the two models. Section 6.4 provides chapter summary.

6.2 A Sequential Model for Users' Action Prediction

This section discusses a sequential model for predicting users' actions. The name sequential implies to the assumption that users follow certain patterns while reaching for desired information and these patterns are commonly shared. The sequence in which the items are accessed is of importance with this model. In brief, the task of this method is to extract frequently accessed sequences of page views in their original order and based on that construct relevant locality profiles to be used in the prediction process.

The process of establishing the sequential prediction model starts with acquiring web usage data from the log and reconstructing users' navigation paths in sessions (Figure 6-1). These paths are then fully minimized to remove any redundant operations. The first part of the processing is somewhat similar to the method used for extracting user preference profiles; however the methods differ as of the locality profiles extraction. The minimized navigation paths undergo a step for filtering out non-relevant paths. Paths, where the number of operations after minimization remains less than three do not carry any value for the prediction model. This is due to the fact that if the path contains only one or two operations, there are no items in it to indicate navigation progress, on which the prediction model is based on. Thus, such patterns are not of interest.

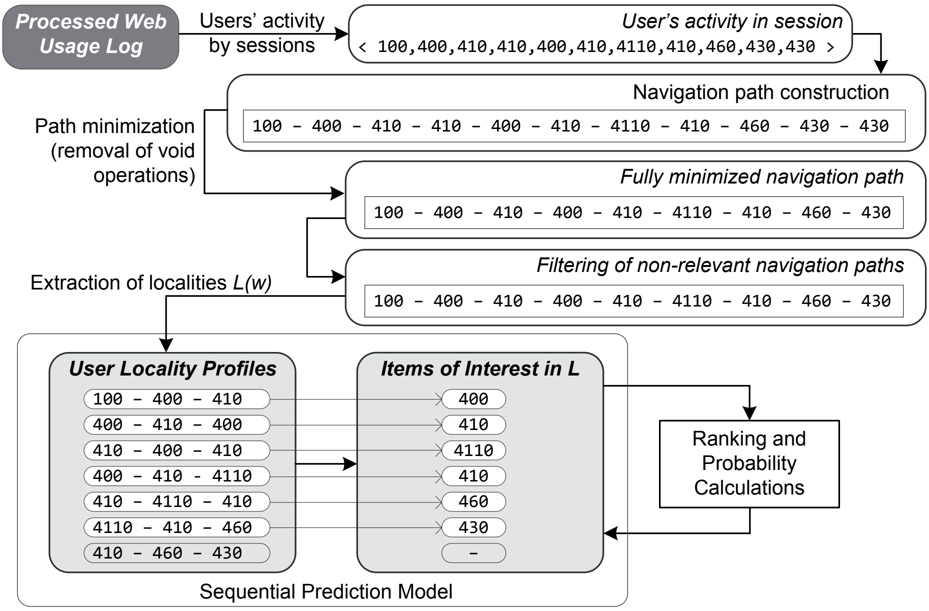


Figure 6-1. *Processing of users' behaviour in sessions to extract user locality profiles and set of items of interest. Example data shown for locality size w=3.*

On the minimized and filtered navigation paths the concept of the locality model is applied and localities of size *w* are extracted together with items of users' interest while occupying particular locality profile. These two sets together form user locality and interest profiles. In other words, sessions $s=<p_i, p_{i+1},\ldots, p_n>$, where $s\in S$, $p_i\in P$ and P is the set of all pages in WIS are processed and localities $L=p_j, p_{j+1}, \ldots, p_m$ are extracted, where $p_j \neq p_{j+1}$, and p_j is a requested page ID, and items of further interest I detected such that $I(L)=p_{m+1}$, if available in the processed data. Figure 6-2 outlines the algorithm for extracting user locality profiles and corresponding items of users' interest. The algorithm assumes that the session data has already been processed for path minimization and elimination of non-relevant navigation paths.

After the localities have been extracted, each of them is equipped with a set of items of further interest I for users while visiting a locality L. For each item in this set rank and probability values are computed. The probability value for each item of interest is calculated as given by Equation 6-1, where $f(I_i)$ is the request rate for item of interest and $f(L_i)$ is the occurrence rate of a corresponding locality profile. For ranking, an inverse time weighting algorithm is applied as given by Equation 6-2. In the formula, t_{age} represents the time into past, and $f(I_i)$ the hit rate for a given item during a period of time specified by t_{age}. In general t_{age} is measured in days.

$$p = \sum_{i=1}^{n} \frac{f(I_i)}{f(L_i)} \qquad (6\text{-}1)$$

$$Rank = \sum_{i=1}^{n} \frac{f(I_i)}{t_{age}(I_i)} \qquad (6\text{-}2)$$

```
Function extractLocalityInterestProfiles(s, w)
    /*
        s: array                  = operations in user session being processed
        w: int                    = locality window size
        localityProfiles: array   = set of extracted localities
        runningLocality: array    = locality being extracted
        localityInterest: array   = set of items of interest for localities
        quantity: int             = number of localities extractable from path
    */

    BEGIN
        quantity = count(s) - w + 1
        i = 0
        while i < quantity do
            j = i
            while j < w + i do
                add s[j] to runningLocality
                increment j
            end while
            if exists(s[j])
                add runningLocality to localityProfiles
                add s[j] to localityInterest
            end if
            reset runningLocality
            increment i
        end while
        return array(localityProfiles, localityInterest)  /*container for results*/
    END
```

Figure 6-2. *Algorithm for obtaining user locality profiles and corresponding items of interest from user's operations in session.*

Hence, the model is based on discovering sequences of pages that are frequently requested together and are common for many users. The order in which items are accessed plays a crucial role. The model groups users with similar browsing patterns and discloses further requests for pages outside the particular locality users currently occupy. These requests are then mapped to locality profiles with calculated rank and probability values. As the model is dynamic and heavily dependent on web usage log, it needs to be regularly retrained to cover users' new access trends and cope with modifications introduced to WIS.

6.2.1 Using the Sequential Model for Predicting Users' Actions

The established sequential prediction model is stored in a repository. The author has used the MySQL[1] DBMS for storing the processed model data. This repository contains all the extracted user locality profiles with additional characteristics (e.g., pattern length) and references to corresponding items of interest, each with a rank and probability value.

[1] http://www.mysql.com

To generate a prediction for online user, the repository is queried based on the active locality profile detected from the online user's browsing activity. The prediction engine extracts from the online user's nearest action history the latest locality profile L_a of size w, such that $L_a(w)=p_{m-w+1}, \ldots, p_{m-1}, p_m$ and $p_{m-i} \neq p_{m-i-1}$, and performs matching over the data available in the prediction model. Thus, only the latest actions covered by active locality profile are of interest while generating predictions. It is essential that the online user has performed at least $q=min(w)=2$ operations before the prediction engine (PE) can be evoked and a prognosis for next page request computed. The concept of the prediction establishment is outlined on Figure 6-3.

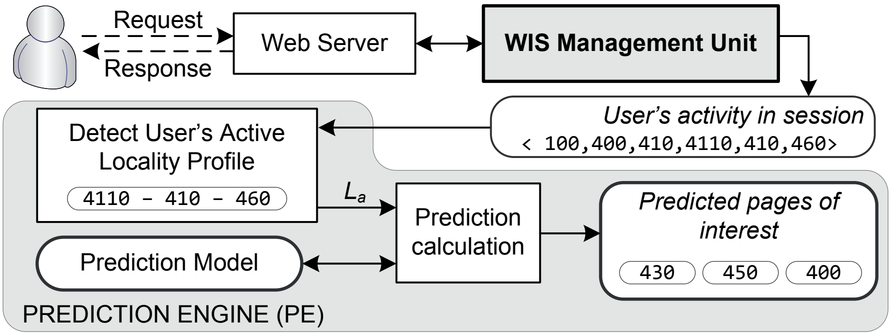


Figure 6-3. *The concept of prediction establishment with sequential prediction model.*

Typically there are multiple predictions possible with varying confidence, based on the interest item ranking and probability. The confidence is calculated as the product of rank and probability values. The engine may choose to form no prediction at all, if no matching locality profile is found. While exploiting the prediction engine in a WIS for strategic adaptations or web personalization, the PE must be with rapid response, otherwise the prognosis should be declared unavailable, to avoid user disturbance and delayed page load completion.

6.2.2 Experimental Results

Several experiments were conducted with the proposed sequential prediction model to evaluate it in practice. The experiments were performed in two categories:

- Active use of the prediction model on real website,
- Empirical study with randomly selected sample data from the DCE WIS web usage log.

While establishing the sequential prediction model for the DCE website, 269 782 navigational paths were processed. After minimization, removal of duplicate neighbouring items and filtering out paths with only one or two operations, 87 953 paths remained to be processed in further. Based on these navigation paths, 13 137 distinct localities of sizes w=2 and w=3 were extracted to establish the prediction model for the experiments.

For the active use experiment on a real website, a test prediction engine was set up, which worked in parallel with the WIS management unit of the DCE website. With every page request the latest action history of a user was taken from the session cookie and sent to the PE. The PE returned the prognosis for different algorithms implemented in the engine. These results together with the users' actual next page request were stored into a database for further analyses. At the time of the experiments 2 251 predictions had been calculated during a 15 day trial period. In 79% of the cases locality profiles of w={2,3} were found; thus 21% of the localities remained in size w=2, due to extracting localities from active session data as soon as online user had made sufficiently enough page requests, in this case q=2 operations.

For the empirical study the experiments were conducted with two sets of randomly selected sessions: one with 29 311 (V10) and the other with 147 883 (V50) sessions. The labels V10 and V50 mark correspondingly the approximate amount of sessions used from the available web usage log data, expressed in percentages. As the experiments were aimed on locality profile sizes w=2 and w=3, the randomly selected sessions were restricted to have at least 4 operations. The operations in sessions were treated as if they were performed by online users and fed to the prediction engine as an input.

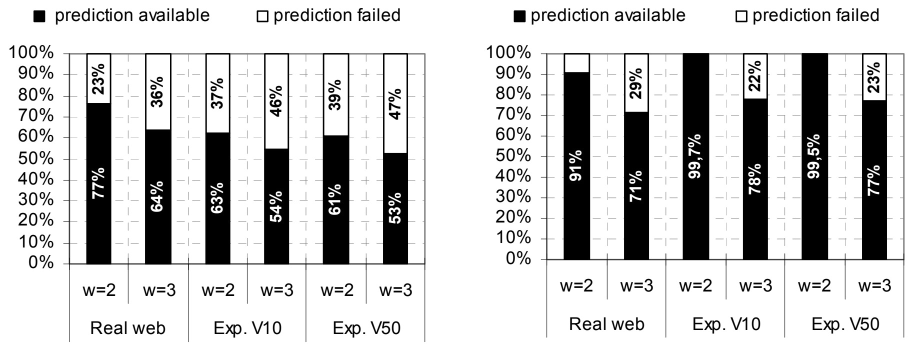
Six different algorithm implementations were established for the prediction engine as outlined in Table 6-1. All the algorithms are based on online users nearest activity history onto which the locality model with indicated window size *w* is applied on. The selection of predicted items in these algorithms is based on rank and probability values. More precisely, the confidence for predicting each item is found as a product of the corresponding item rank and probability values. The general implementation used to provide predictions based on algorithms A1–A6 is presented in Appendix E.

First, prediction availability was studied. The prediction availability as a measure indicates the systems ability to provide predictions and is expressed as a given by Equation 6-3. The conducted prediction availability analysis showed that the sequential PE needs to address the issues of redundant operations performed by online user. This is due to the fact that the data used for the model training was based on minimized navigation paths, thus there were no redundant operations. On the other hand, it is hard to predict when users are going to make void operation requests, as this does not indicate progress in browsing flow. In this case the majority of redundant operations consisted of series of page requests where a user was already on. The results of the analysis are outlined on Figure 6-4, showing a comparison of the three studied groups in two categories. When comparing the prediction availability over different locality profile sizes, localities of size w=2 performed better. If localities of size w=3 were used, the prediction coverage was still rather good. As expected, the prediction availability was better for the groups where the engine was run on sample data from the same log on which it was trained on. Still, the experiments with real website usage were promising.

Table 6-1. *Algorithms used for users' action prediction.*

Algo-rithm	Locality window size	Description
A1	w=2	Prediction of the most probable item of interest according to confidence value for online user's detected locality profile of size *w*
A2	w=3	
A3	w=2 & w=3	Prediction of the most probable item of interest according to confidence values over a union of predicted items with applied window sizes *w*. In case on equal confidence values, item predicted for locality profile w=3 is preferred.
A4	w=2	Prediction of the top three most probable items of interest according to confidence values for online user's detected locality profile of size *w*.
A5	w=3	
A6	w=2 & w=3	Prediction of the top four most probable items of interest available in the intersection of the predicted sets for different locality profile sizes of *w*.

$$Availability = \frac{\#of\ predictions\ provided\ for\ requests}{total\ \#of\ requests} \tag{6-3}$$



(a) (b)

Figure 6-4*. Prediction availability analysis: (a) over all operations performed, (b) with elimination of redundant user operations.*

The prediction accuracy analysis (Figure 6-5) showed that the sequential prediction model performs better than just using the probability for interest item request. Accuracy indicates the rate of system ability to provide correct prediction (Equation 6-4). The algorithms A1 and A2 had poorer performance in comparison to their derivates A4 and A5, due to the differences in their implementation. Algorithms A1 and A2 provided as a result only one most probable item of interest, while A4 and A5 a set of three. As seen, algorithm A3 did not provide expected results. Algorithm A6 on the other hand provided already rather good accuracy. Still, it should be noted that unless there is only one item of interest to predict, the accuracy can never be 100%.

$$Accuracy = \frac{\#\,of\ correct\ predictions}{total\ \#\,of\ predictions} \qquad (6\text{-}4)$$

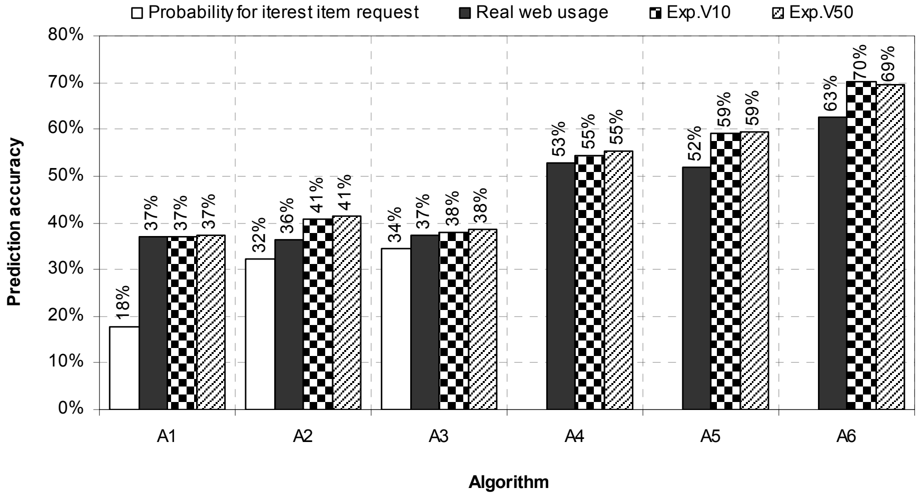


Figure 6-5. *Results of prediction accuracy analysis for the sequential prediction model on algorithms A1-A6.*

Obviously, prediction availability and accuracy are related to the amount of discovered locality profiles and corresponding items of interest available in the model. The more the set covers, the better prediction availability can be obtained. However, it may turn out to be on the cost of prediction accuracy, as with bigger set there are more items of interest to choose from for the PE, and in the worst case the confidence on ranking and probability values may turn out equal for the available items. Thus, with already two items to choose from the probability to be mistaken is already 50 percent. In the prediction accuracy analysis (Figure 6-5) the difference between the results for algorithms A2 and A5 is around 20% in favour of A5, which clearly illustrates the latter situation.

Figure 6-6 provides an overview of the proportional quantity of items of interest available in the model. As can be seen, with user locality profiles with size w=2 in almost half of the cases the prediction engine had to choose from more than three items, while locality profiles of size w=3 provided a more accurate approach and had only one item available in 50.6% of cases. In overall, the set of items of interest for locality profiles of size w=2 was 5.6 and in case of w=3 in average 2.9 per locality profile.

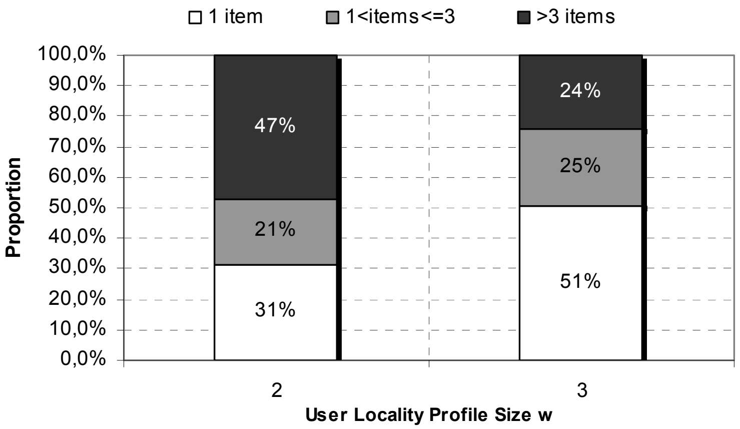


Figure 6-6. *Quantitative classification of items of interest to choose from while generating predictions.*

6.3 A Conceptual Model for Users' Actions Prediction

While the sequential prediction model took an approach to match users' locality profiles to items of interest discovered through original data mining procedures, the conceptual model provides an approach to predict users' actions on the modelled conceptual interests. The model assumes that pages users view are conceptually related and thereby aims to capture the conceptual model in users' mind while they are browsing for information, and apply it in the process of predicting their actions in the nearest future.

The process of establishing a conceptual prediction model starts off similar to the creation of the sequential model. With a fully automated log system, web usage log to describe users' preferences and behaviour has been gathered and processed. The data in the log is used to reconstruct users' navigation paths in sessions (Figure 6-7). These navigation paths are then minimized to remove operations not carrying any value of progress, thus being void. The minimized navigation paths undergo filtering for non-relevant paths which are eliminated during the process. Onto the minimized navigation path the concept of the locality model (Section 4.3.2) is applied to extract user locality profiles *L* of size *w* and their related items of interest for users. During the extraction, each item in the locality profile is mapped to a concept in the web ontology (Section 5.3) through concept detection. This is also done for each of the corresponding item of interest. Altogether, if a user session is defined as $s=<p_i, p_{i+1},...p_n>$, where $s \in S$, $p_i \in P$ and *P* is the set of all pages in WIS, localities *L* of size *w* consisting of concepts are extracted such as $L=\{c_1, c_2, ..., c_m\}$, $c_i \neq c_{i+1}$, where $c_i \in C$, and *C* is the set of concepts available in web ontology; and the following conditions hold $\forall(p_i \in P)\ \exists\ (c_j \in C)\ |\ <p_i,c_j> \in \varphi\ \&\ \varphi \sqsubset PxC$. The process results in two sets: user interest concept profiles (IC) and referred interest concepts (RIC) constituting to the conceptual prediction model. For each RIC its corresponding rank and probability value is found. Figure 6-8 outlines the algorithm used for obtaining user interest concept profiles with referred interest concepts.

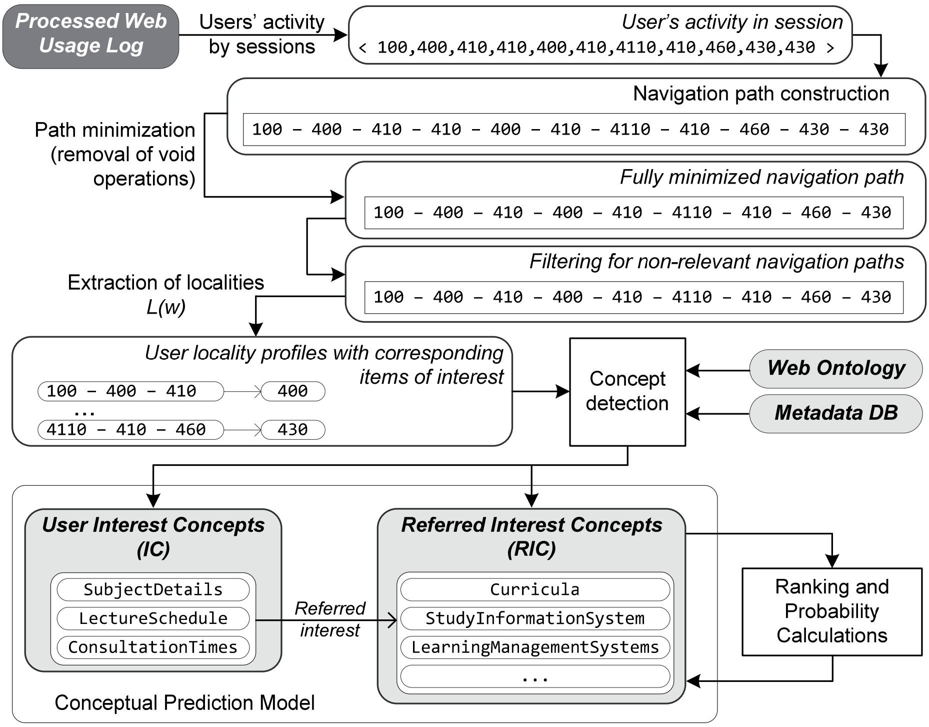

Processed Web Usage Log
Users' activity by sessions
User's activity in session
< 100,400,410,410,400,410,4110,410,460,430,430 >
Navigation path construction
100 – 400 – 410 – 410 – 400 – 410 – 4110 – 410 – 460 – 430 – 430
Path minimization (removal of void operations)
Fully minimized navigation path
100 – 400 – 410 – 400 – 410 – 4110 – 410 – 460 – 430
Filtering for non-relevant navigation paths
100 – 400 – 410 – 400 – 410 – 4110 – 410 – 460 – 430
Extraction of localities L(w)
User locality profiles with corresponding items of interest
100 – 400 – 410
400
...
4110 – 410 – 460
430
Concept detection
Web Ontology
Metadata DB
User Interest Concepts (IC)
SubjectDetails
LectureSchedule
ConsultationTimes
Referred interest
Referred Interest Concepts (RIC)
Curricula
StudyInformationSystem
LearningManagementSystems
...
Ranking and Probability Calculations
Conceptual Prediction Model

Figure 6-7. *Processing of users' behaviour in sessions to user interest concepts and a set of referred concepts of interest. Example data shown for locality size w=3.*

The probability is calculated based on Equation 6-5, where $f(RIC_i)$ is the request rate for a referred interest concept and $f(IC_i)$ is the occurrence rate of its related user interest concept. Ranking of referred interest concepts is obtained by Equation 6-6, where t_{age} represents the time into past and $f(RIC_i)$ the hit rate for a given item during a period of time specified by t_{age}. As with the sequential model, in general t_{age} is measured in days.

$$p = \sum_{i=1}^{n} \frac{f(RIC_i)}{f(IC_i)} \tag{6-5}$$

$$Rank = \sum_{i=1}^{n} \frac{f(RIC_i)}{t_{age}(RIC_i)} \tag{6-6}$$

In what the two models mainly differ is that with the sequential model the order in which the pages are requested while occupying a locality has a crucial role to play in providing predictions. The conceptual prediction model however concentrates on the concepts of web ontology behind requested pages while occupying a locality regardless of their access sequence.

```
Function extractLocalityInterestProfiles(s, w)
 /*
  s: array                       = operations in user session being processed
  w: int                         = locality window size
  interestConceptsProfiles: array = set of user interest concepts (profile)
  runningInterestConcept: array  = set of concepts in locality being handled
  setOfReferredIC: array         = set of items of interest for IC
  quantity: int                  = number of localities extractable from path
  getWebOntologyConcept(int): int = method to get corresponding concept ID in
                                   web ontology over metadata DB
 */

    BEGIN
        quantity = count(s) - w + 1
        i = 0
        while i < quantity do
            j = i
            while j < w + i do
                c = getWebOntologyConcept(s[j])
                add c to runningInterestConcept
                increment j
            end while
            if exists(s[j])
                add runningInterestConcept to interestConceptsProfiles
                cc = getWebOntologyConcept(s[j])
                add cc to setOfReferredIC
            end if
            reset runningInterestConcept
            increment i
        end while
        return array(interestConceptsProfiles, setOfReferredIC)  /*res. container*/
    END
```

Figure 6-8. *Algorithm for obtaining user interest concepts profile and corresponding referred interest concepts set.*

6.3.1 Using the Conceptual Model for Predicting Users' Actions

The predictions for online user are generated based on the user's latest browsing history defined in the locality window of size *w* and kept in the user's operations track updated with every operation. The prediction engine extracts the latest operation history for locality profile L_a of size *w* such that $L_a(w)=p_{m-w+1},\ldots, p_{m-1}, p_m$ and $p_{m-i} \neq p_{m-i-1}$. The pages in the locality profile L_a are then mapped to concepts of web ontology. The PE is ready to calculate predictions as soon as user has made at least $q=min(w)=2$ page requests.

As with the sequential model, there can be multiple concepts predicted with different or equal rank and probability values constituting the confidence and a selection has to be made. This is up to the algorithm used for the prediction. The prediction has to be provided in web time, thus the engine needs to have a short response time. In principal, the prediction engine is similar to the one used with the sequential model, with additional steps added to deal with concepts of web ontology. Figure 6-9 outlines the establishment of next page request by using the conceptual prediction model. During the process items in user locality profile are translated into concepts available in web ontology using the metadata

DB, and corresponding concepts of interest are detected. The prediction result is then translated back to pages available in the WIS.

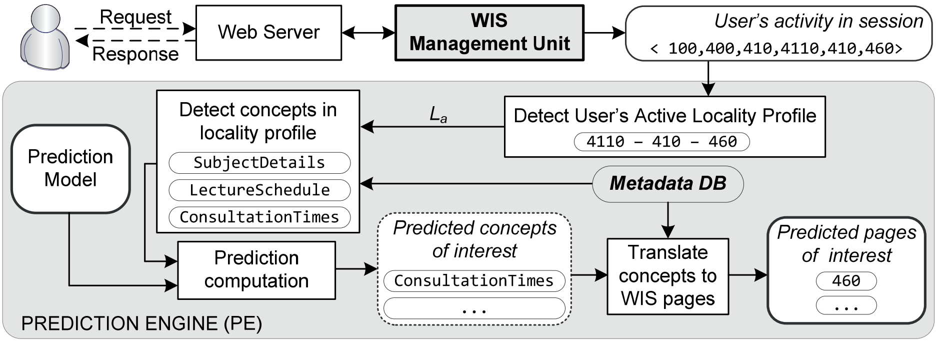


Figure 6-9. *The concept of the prediction engine and prediction establishment based on the conceptual prediction model.*

6.3.2 Conceptual Distance of Requested Pages

When users are browsing the Web, they have a certain conceptual model in their mind which they apply for the task. Also, the content presented on a website is organized according to some kind of a conceptual model. The conceptual prediction model proposed here is based on the belief that users are interested in conceptually close items. A study into users' actions has confirmed this.

To investigate the conceptual distance between requested pages during web sessions a conceptual distance *D* as the number of relationships between two concepts c_i and c_j in a *is-a* hierarchy in the web ontology was defined. For example, in the web ontology (Figure 5-3) the conceptual distance between the classes *AuthorInstructions* and *UsefulResources* is *D*=2, calculated on the basis of the number of *is-a* relationships between the two concepts. The study into users' behaviour showed that pages are requested within an average conceptual distance *D*=1.9 (Figure 6-10). The slight upwards curve at the end of the graph is apparently due to cross-links between pages. The study also unveiled that in case of prediction mismatch users had accessed pages within an average conceptual distance *D*=2.1, compared to match cases where the corresponding value was *D*=1.8. It indicates that in most cases users had requested conceptually neighbouring items, thus they searched for conceptually related information. The study was based on 87 953 user sessions.

The findings led to a development of an additional prediction model as a sub-model of the conceptual model taking account the distance between the predicted concept and the last accessed concept. With this model, in the algorithms A1–A6 a small modification to re-evaluate the active confidence of a predicted item on the basis of inverse conceptual distance to boost prediction of conceptually closer items was introduced. The conceptual distance was calculated for each referred interest concept available for prediction in reference to the last concept user had accessed by page request and thereby available in user locality profile. Section 6.3.3 provides experimental results for the sub-model, labelled as '*Conceptual model with D*'.

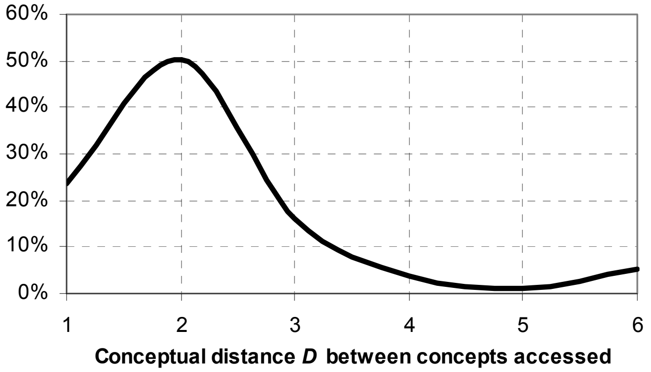


Figure 6-10. *Conceptual distance between requested pages.*

6.3.3 Experimental Results and Comparison of the Two Models

For the experiments with the conceptual prediction models the active use PE (introduced in Section 6.2.2) working in the backgrounds of the DCE website was modified and supplemented with the corresponding models. The prediction algorithms remained basically the same. The descriptions for algorithms A1–A6 in the context of the conceptual prediction model are provided in Table 6-2. In parallel, the prediction engine continued to compute predictions based on the sequential model, enabling comparison of the two rather different approaches.

During the conceptual prediction model establishment for the DCE website 87 953 suitable navigation paths were identified and used for model training. Out of these sessions a fair 37.5% were performed by returning users. The predictions provided on the model are based on 177 621 user interest concepts related to 313 910 referred interest concepts.

During the time period of over 11 months (335 days) predictions proposed by the PE for both of the approaches – sequential and conceptual – were logged to be analysed and compared. This log mirrors the actions of real website users and the prognosis calculated by the PE in correspondence to users' actions and available prediction models. Each of the log record was equipped with a timestamp and a reference to corresponding record in the web usage log to enable in depth analysis, model evaluation and tracing of sessions as necessary.

In addition, to evaluate the sequential model in the context of the necessity for re-training, the collected prediction log data was run in a simulation on the re-trained sequential model. Hence, the data used for re-training the sequential model contained also the data for which the prognosis had been already computed. By re-running the prediction engine simulation, approximated results for the predictions were obtained for the sequential model in the context of continuous model update.

Table 6-2. *Algorithms used for users' action prediction in the PE based on the conceptual model.*

Algo-rithm	**Locality window size**	**Description**
A1	w=2	Prediction of the most probable concept of interest according to confidence value for the concepts covering online user's locality profile of size w
A2	w=3	
A3	w=2 & w=3	Prediction of the most probable concept of interest over a union of predicted concepts covering online user's locality profiles of sizes w. In case on equal confidence values concept predicted for locality profile w=3 is preferred.
A4	w=2	Prediction of the most probable top three concepts of interest according to confidence values for the concepts covering online user's locality profile of size w.
A5	w=3	
A6	w=2 & w=3	Prediction of the top four most probable concepts of interest available in the intersection of the predicted sets for different locality profile sizes of w.

The analysis of prediction availability showed the conceptual model to perform better than the sequential one, which in 31% of cases suffered from inability to provide any prediction at all. As the comparison with the re-trained sequential model suggests, this is probably the cause of the model decay over time. The conceptual model was able to provide a prediction in 91% of cases, which is significantly better than the result for the sequential model. Also, it indicates that the conceptual model copes better with the so called 'cold-start' problem, where the sequential model is unable to compute prediction as the locality on which it could be proposed is missing from the model, i.e., a navigation path has never been taken by any of the users before or has not yet described in the prediction model. Figure 6-11 outlines the results of the prediction availability analysis over algorithms A1–A6 for the models.

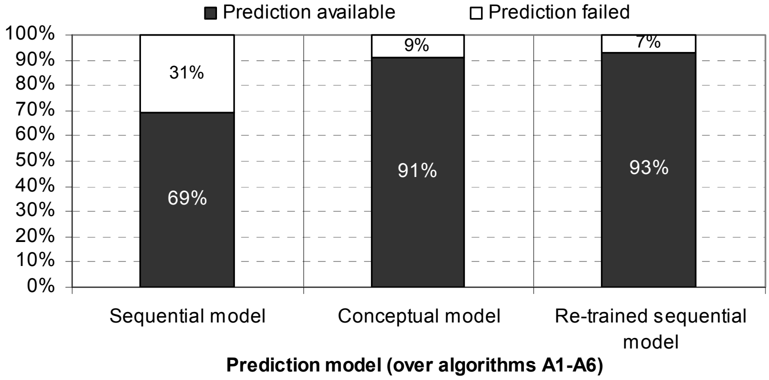


Figure 6-11. *Results of the prediction availability analysis over algorithms A1–A6.*

Depending upon the algorithm, the maximum prediction availability rate for the conceptual model was at 99.9%, while the algorithms A2 and A5 had the lowest performance rate at 79.9%. The latter two algorithms had poor prediction coverage also for the sequential model. The reason for poor performance for those two algorithms lies in the fact that not all users made enough operations during their web sessions to enable the algorithm to function.

The conceptual prediction model also performed better than the sequential one in the context of prediction accuracy. The comparison of prediction accuracy over algorithms A1–A6 for the four models is shown on Figure 6-12.

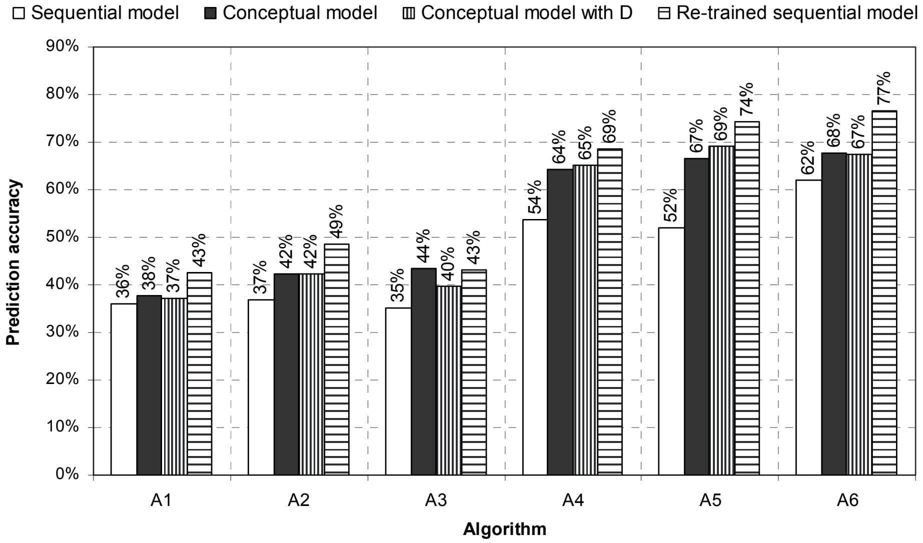


Figure 6-12. *Comparison of the prediction accuracy analysis for the sequential and conceptual modelling approach.*

Two conceptual models are presented (Figure 6-12): the original conceptual model and its sub-model which takes advantage of the conceptual distance between the last requested concept and the one being proposed with prediction, labelled as 'Conceptual model with D'. As can be seen, in proposing only one concept per prediction the conceptual model has better accuracy, however for algorithms A4 and A5 the 'conceptual model with D' provides a slight increase in accuracy. The re-trained sequential model is shown for comparison only (as no real predictions were made on it) and for obvious reasons it performed the best. Nevertheless, it enables a rather good comparison of decaying model and an updated one. In overall, the conceptual approach delivers in average 10% better accuracy than the sequential model.

The accuracy delivered by the model and algorithm combination is dependent on the amount of choices available in the model, which in turn relies on data modelling approach. Figure 6-13 depicts the quantitative classification of concepts of interest the PE had to choose from in the conceptual model in comparison to the sequential model. As the results indicate, with the conceptual model the choice over items to predict is more diverse in comparison to the sequential model. This is due to the fact that the conceptual model is based on concepts covered by user locality profile and the order in which these concepts appear does not influence the set of referred interest concepts. In overall, the average number of referred interest concepts (RIC) for user locality profiles of size w=2 was 7.01 and in case of a locality size w=3 in average 3.99 RICs for the conceptual prediction model.

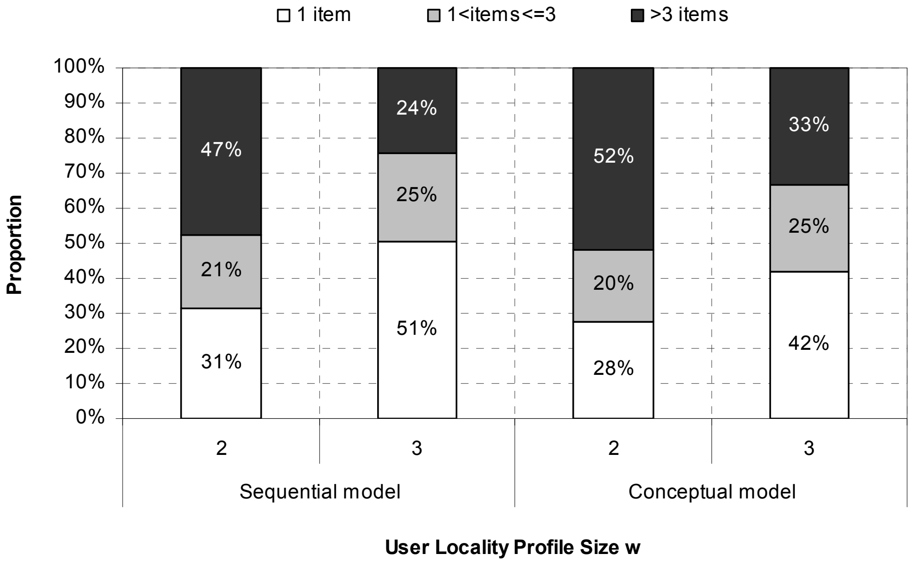


Figure 6-13. *Quantitative classification of items and concepts of interest to choose from while generating predictions based on sequential and conceptual model.*

In conclusion, the conceptual model proposed herein performs better than the sequential model, taking into account the prediction availability, and accuracy. As the evidence suggest, the sequential model must be regularly updated to avoid its aging and enable prediction engine to provide items matching users' latest access trends. Although freshly updated sequential model may have better accuracy, the conceptual model is safer to use, providing more stable prognosis and better prediction coverage even in case of 'cold start'. It is preferable to have the prediction engine to compute a prediction even if it might not be as accurate, than not to provide any prediction at all.

6.3.4 Discussion

In terms of prediction accuracy the research of other authors reports different values, dependent on approach, methodology and used data. In [58] authors have studied an approach to personalize search results using ontological user profiles and domain ontology. In their experiments with web search personalization for unknown user they gained accuracy up to approximately

30%. Based on their experiments, they concluded that applying ontological user profiles for personalizing search results is an effective approach.

In [90] a framework for predicting user's next page request was proposed. The experiments were conducted with association rules, frequent sequences and frequent generalized sequences. The authors found the possible quantity of items to be predicted per query to lie in between 2 and 30 items, depending heavily on the mined combinations. In the experiments conducted in this thesis, the minimum number of items per query to predict was 1 and the maximum 55, whilst the average for the sequential model was 3.8 and for the conceptual model 4.6 items per query. The prediction availability obtained for the approaches described in [90] was in the range 18–52%, and in favour of association rules approach, while the overall accuracy of the prediction engine was at maximum at 71% for the association rules approach and a little below 60% for frequent generalized sequences and even lower for frequent sequences approach. However, these are the maximum values attained in the experiments. In the category of predicting only one item per prediction the obtained accuracy was reported in favour of frequent sequences at accuracy rate of approximately 28% only. The drawbacks of the approaches described in [90] mainly lie in the fact that the models were unable to predict new navigations that had not occurred in the log before, thus they suffered from the 'cold-start' problem.

In [92] researchers reported prediction accuracy up to 80%, nevertheless, the average accuracy of their recommendation system based on Q-Learning problem remained at 52%. In [93] a collaborative filtering approach of providing recommendations based on semantic tagging was explored. The authors gained only an average precision of 27% in their experiments. Semantic enhancement of personalized recommender system on the other hand has shown good experimental results, delivering an improvement of 10% over collaborative filtering and accuracy up to 83% [94]. Bonino et al. [95] report maximum precision value of 85% and an average of 52% for their prediction model based on evolutionary algorithm. Davison [133] reports overall predictive accuracy of nearly 20% for top one and 40% for top three predictions of user future web actions based on HTML content.

As shown, the research and reported achievements vary a lot and are not fully comparable due to different methodologies applied. Also, the results are heavily dependent on used dataset and experimental environment, in particular, if the experiments are conducted on real web information systems or just simulated based on collected log data. Still, the aforementioned figures provide some reference values to compare the results described in this chapter to the works of other researchers.

6.4 Chapter Summary

With every click users make on a website, they leave behind pieces of information describing their preferences and intentions. Fully automated log systems enable to capture these events into web usage logs, where the data can be mined and used for predicting users' intentions on the basis of collective intelligence. Such predictions provide a valuable source of data for adaptive web systems and recommender systems.

This chapter proposed two different approaches to model users' behaviour for predicting users' actions. Firstly, a sequential prediction model was introduced, based solely on the user latest activity pattern. In the model, different user locality profiles were processed and items of interest identified. The major drawbacks of this model were its dependence on the operation sequence and decay over time. Secondly, a conceptual prediction model was proposed, where users' behaviour was mapped to concepts of web ontology and corresponding concepts of probable interest extracted. As shown, users are searching for conceptually close items. The conceptual model solves some of the problems of the sequential model and also provides better performance in sense of prediction availability and accuracy. This model is not so dependent on frequent updates. For both of the models experiments were carried out on real website with real web users. The results of this long term empirical study suggest that the conceptual model suites better for a kind of personalization service, having better prediction coverage and ability to provide prognosis even when the sequential model failed.

The main contribution of this chapter is a method for attaining conceptual model for predicting users' future actions based on conceptually modelled users' interest and web ontology, together with prediction algorithms. It is highly likely that combining the conceptual model with indicators of user interest as well as feeding the prediction engine with more knowledge about user, for instance in case of anonymous ad-hoc web user some information about the user's previous visits, the prediction precision could be enhanced in further. The prediction engine as such provided herein is seen as a part of a recommender system delivering web adaptation and personalization. The methods described in this chapter are applied in the following Chapter 7, where an approach to complement the prediction with users' domain model is provided. However, including more knowledge about anonymous ad-hoc web user to fortify the predictions remains an interest of future studies at this point.

Chapter 7
PROVIDING RECOMMENDATIONS FOR ANONYMOUS AD-HOC WEB USERS

Daily, new information is added by thousands and thousands of users, escalating the sophisticated set of information resources available in various forms over the Internet. All this has resulted in a plethora and diversity of information, introducing problems of successful information retrieval for web users. The fundamental problem is whether users should spend their valuable time for searching a piece of information they are seeking for, or should web information systems try to foresee users' needs and assist them by providing the information before users explicitly submit request for it.

In this chapter a framework for establishing viewing recommendations for anonymous ad-hoc web users based on collective intelligence mined form web usage logs is described. Two previously discussed methods are integrated to reinforce recommendation establishment. In particular, the method for learning web users' domain models discussed in Chapter 5 is used to specify the predictions provided by the conceptual prediction engine described in Chapter 6. As a result, a recommender system is proposed to enable web personalization for anonymous ad-hoc web users to enhance their web experience.

7.1 Introduction

The Web has become a crucial part of our everyday lives, being a source for information and news, a platform to connect with other users through various social web applications, a medium for citizens to communicate with state via use of e-government services, and doing business online, regardless whether it is internet banking or shopping. It has also become an important instrument for online marketing. Today, the Web is everywhere, starting from computers and ending up with portable devices such as smart-phones and tablets. Some of us would not imagine the life without it and being web-present anymore.

As more and more information is made available over the Internet, users are facing information overload making it harder for them to find information searched for at once. Studies on web search engines [8] have shown users rarely to look beyond the first page of search results. Even more, users expect search engines to provide correct results immediately, based on a few rather general keywords specified in search query. Besides the use of search engines, adaptive web and web personalization is seen as a remedy to the information overload problem by delivering refined access to information and thus meeting users' expectations. The utter aim is to provide access to the right information at the right moment in order to tailor the web to a particular user. This is achieved through smart exploitation of recommender systems and by tactical adaptation of the presentation layer of web information systems as a result of collaboration between the user and the system, representing a sort of artificial intelligence.

In general, personalization in the context of anonymous ad-hoc web users is based on overall usage information, i.e., collective intelligence and therefore seemingly transparent for users. In order to deliver recommendations, users' interests are learned by collecting information about their behaviour. Using advanced mining techniques, user profiles, representing collective intelligence harvested from WIS, are modelled. These profiles are then used to detect online users' probable intentions and informational needs using accumulated knowledge about users with behaviour alike, and personalization is delivered as a result of machine learning.

Whilst usually web personalization assumes users to be somehow identifiable, this chapter provides an approach to operate around an anonymous ad-hoc web user – a user about whom the system does not keep a record on (i.e., a user profile), nor the user needs to log into the web information system for identification. Such user can be either new or returning visitor. This makes the recommendation task rather sophisticated in comparison to providing recommendations to identifiable users about whom the system may already have learned a lot, and users as well may have explicitly provided valuable details about themselves. Herein, recommendations for ad-hoc anonymous web users are generated based on the detected user profile, and items provided by the prediction engine as a part of the recommender system (RS).

Previously, in Chapter 5, a method for learning web users' domain models was described. In this chapter, this model will be used to detect online user profile, which is then applied to adjust the prediction result. In Chapter 6 two methods for predicting users' future actions on website were discussed. Out of the two, the conceptual prediction model had the best performance and thus suites well to be exploited in a recommender system for a kind of personalization service. Moreover, as the model is based on the same web ontology as the users' domain model, the knowledge in those two complement each other. The author of thesis herein claims that the general prediction result computed by the prediction engine can be concretized by applying a user profile on it and thereby improve the recommendation set. This forms the main concept of the proposed recommender system, covered by the author's publications [128] and [134].

The following sections of this chapter are organized as follows: Section 7.2 discusses the method of establishing recommendations, while Section 7.3

describes the experimental results to confirm the recommendation approach, and Section 7.4 provides discussion on the application of the recommendation results for personalizing web view for anonymous ad-hoc web users through tactical web adaptation.

7.2 Web Recommendations Based on User's Behaviour

The proposed recommender system (RS) to provide page view recommendations for anonymous ad-hoc web users is based on the users' domain model (UDM) and the conceptual prediction model, representing users' conceptual interest model (CIM), respectively described in Chapters 5 and 6.

The recommender system consists of two major parts. The first component is the prediction engine to acquire pages that might be of interest for an anonymous ad-hoc online user, based on the users' conceptual interest model (CIM). The second component is the user classifier, which is used to determine the type of an online visitor.

The task of the recommender system is to monitor and analyze the actions of online user, and apply this behaviour data to UDM and CIM to determine user's interests for providing viewing recommendations. In particular, the RS exploits online user's latest browsing activity and feeds this data to the user classifier (UC) and the prediction engine (PE). The set of predicted items are then adapted to detected user profile. The resulting set on interest concepts are mapped to WIS pages using the metadata DB, and thereby the recommendation has been established. The implementation of the recommender system, outlining also the data flow for recommendation establishment, is described on Figure 7-1.

The recommendation process starts with monitoring online user's actions and maintaining such user's latest activity history, for instance in the form of cookies or session variables. Based on this data, active user locality profile L_a is created and used as an input for both of the components of the RS – the prediction engine and the user classifier.

The prediction engine is based on the conceptual prediction model (Chapter 6), and enables to detect probable items of user interest. To establish a set of predicted concepts c_p, the conceptual interest model (CIM) is queried for referred interest concepts (RIC) based on the interest concepts (IC) available in the active user's locality profile L_a, resulting in a set of predicted concepts $c_p \in RIC$. The selection of c_p in the prediction engine is based on the rank and probability values stored in the CIM for each RIC, and is performed according to the selected prediction algorithm. Prediction establishment in detail has been previously described in Section 6.3.1.

The selection of the prediction algorithm to be used in the RS depends on the recommendation aim and available latest activity history (the size of L_a) of the online user. In the context of the proposed RS, the prediction engine is able to start computing predictions as early as the user has performed at least two operations on a website, thus $q=w(L_a)=2$. In the latter case, only algorithms A1 and A4 are available for use. If there are more than two operations available in user's active locality profile L_a, algorithms A1–A6 are applicable for prediction computation, however algorithms A4–A6, returning a set of recommended items, instead of only one most probable item, are preferred for the task.

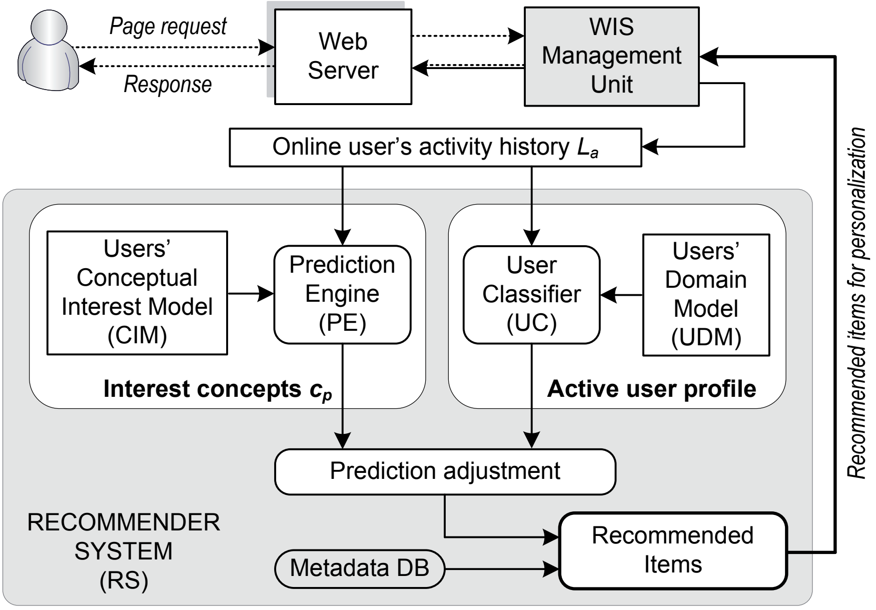


Figure 7-1. *Implementation of the recommender system.*

While in Chapter 6 the algorithms A1–A3 returned only one predicted item and A4–A6 a result-set of multiple predicted items, with the use of the PE in the RS here the sizes of predicted items sets need to be reconsidered. In particular, the sets of predicted items need to be enlarged. This is necessary due to the prediction adjustment phase, where the predicted concepts (c_p) are evaluated in the context of detected active user profile. Therefore, it is justified to let the prediction algorithms return *top N* predicted items, where $N \geq 3$. A suggested size is top 5–10 items, if available, depending on the web information system scope, size and targeted application of the recommendation result. Do note that by doing this, the prediction algorithms A1 and A4 become essentially the same, as well as A2 and A5.

When investigating the choice overload problem in recommender systems, Bollen and colleagues [102] suggested keeping the recommendation set below ten items, outlining that the set must be attractive, varied and manageable by users. Herein, this is achieved by controlling the value of *N*, which is to be specified by an expert of area for particular WIS, taking into account the aim and target of delivered web recommendations.

These *top N* predicted items are selected by the PE only based on rank and probability values over all WIS users' behaviour. Thus, they represent the generalization of users' interests, and not taking account any particular user group interests. For successful recommendation, it is necessary to re-evaluate each of these items in the context of a particular user profile. In other words, the active user profile must be taken into account when computing recommendations for web personalization. As the experiments confirm (Section 7.3), this approach enables to improve recommendation accuracy and therefore enables to deliver more relevant recommendations. The prediction algorithms are used to select a set of the most relevant *top N* items from the conceptual interest model (CIM), representing users' general interests. In further, these *top N* items are re-evaluated in the context of an active online user.

When talking about the number of available RICs per IC in the CIM, it can be expected to be a fairly large number for medium-sized and larger corporate websites. For instance, in the studies for the DCE website, the average number of RICs per IC was 6.5 whenever there was a choice of at least two RICs for a given IC, and the maximum to be 39. Thereby, the initial selection using the prediction algorithms is essential.

Established users' domain model (UDM) and the predefined user profiles available in the model are used to reflect online user's interest area. The user classifier (UC) is provided with an active user locality profile L_a and the classification task is carried out. The result is that online user is defined through predefined user profiles in the user profiles ontology. This constitutes a user profile for an anonymous ad-hoc web user.

The next phase in the recommendation generation is prediction adjustment (Figure 7-1), where a selection of predicted interest concepts c_p is performed based on detected user profile. A predicted concept c_p with high rank and probability value could actually be of a priority interest in a user profile that does not match the current online user profile. High rank and probability values do not necessarily mean that a concept is of interest for a particular user. The concepts returned by the prediction engine represent the generalization over all users and are based on access rate. To mitigate the problem, applying prediction adjustment is necessary and justified.

During the prediction adjustment phase, eligible concepts are addressed by re-evaluating the predicted *top N* interest concepts in the context of detected user profile. For interest concepts that also belong to the detected user profile, ranking is increased by a significance factor *u*. In other words, the significance of a particular interest concept c_p is indicated by assigning to it a factor *u*, or vice versa. This provides a lift in *top N* set for concepts that are more relevant to online user according to the detected user profile. Thereby, for each concept $c_p \in RIC$ that is present in the detected user profile, the rank of that particular c_p is multiplied by the significance factor *u* (Equation 7-1). The value range of the factor *u* will be addressed in Section 7.3. Based on the new rank values, the concepts are sorted in a descending order and limited to a set of M items, where $M < N$. By this the prediction adjustment is finished. The recommended value for M is M=[3..5], as an appropriate number of items to be recommended, and which also corresponds to the set size of returned items by algorithms A4–A6 introduced in Chapter 6. In the experiments described in Section 7.3 the value of M has been chosen accordingly, that is M=1 for algorithms A1–A3, M=3 for A4 and A5, M=4 for A6. This also guarantees that the results obtained by the use of the PE described in Chapter 6 and the results obtained for the recommender system in Section 7.3 are comparable.

$$Rank_{lifted}(c_p) = Rank(c_p) \cdot u \tag{7-1}$$

In the final phase of recommendation establishment for each interest concept c_p a corresponding page $p_r \in P$ is found in the WIS, using the web metadata DB, such that recommended set of pages $P_r = \{p_{r1}, p_{r2}, ..., p_{rM}\}$ and $P_r \subset P$. In the metadata DB pages represented in the WIS are referred to concepts of web ontology by their URI references, allowing easy matching of relationships between WIS concepts and available web pages in WIS for users on the World Wide Web. This phase of recommendation generation is necessary as WIS management units generally operate around explicitly identified pages, and so does the DCE web information system. The algorithm used for recommendation establishment is outlined in Appendix F.

Having established a recommendation, it is applicable for adapting the system to online user's needs and intentions, and thereby the method enables to provide a personalized view for a user of a particular web information system.

7.3 Empirical Study on Refining Predictions for Web Recommendations

In order to prove the concept of the recommender system, and to confirm that applying user profiles on to predicted items of interest improves prognosis and recommendation process in general, a set of experiments based on the log data collected from an active prediction engine (introduced in Chapter 6) were carried out. This prediction engine embodies implementations of proposed prediction approaches and works in the background of the DCE website, logging the computed predictions and users' actual requests in real life situation.

To minimize the data amount to be analyzed, the empirical study on refining prediction accuracy was restricted to predefined user profiles *ResearcherProfile* and *StudentProfile,* and 28 552 records (belonging to more than 5 300 sessions) extracted from the active prediction engine log in the interest domain of those profiles, containing the following information:

- User's latest operation history described by locality profiles with sliding window sizes w=2 and w=3,

- Predicted concepts of interest according to algorithms A1–A6,
- Actual page requested by user,
- Reference to corresponding main log system record.

Having extracted the log records, this data was used to run the recommender system simulation for the predefined user profiles *ResearcherProfile* and *StudentProfile* to evaluate the recommender system approach. As the log already contained data about the conceptually predicted items and the actual items visitors requested, the only open question remained was the proper value for the significance factor u to be applied in case of interest match in user profile, to provide lift sufficient enough for those concepts of interest c_p to be included in the recommendation.

A series of experiments with varying values of the significance factor u from 1.0 to 3.0 was conducted (Table 7-1). The best suitable value for significance factor u was discovered to be in the range $1.8 \leq u \leq 2.0$. On the whole, this result was quite expected, as high values of factor u, applied to concepts c_p belonging to detected user profile, elevate their rank and make these concepts competitive with predicted concepts without the support of the user profile but having a high rank value nonetheless. Hence, the prevalence of predicted concepts with high rank value not supported by detected user profile becomes limited. The latter is the underlying idea of the use of the factor u. Forced discarding of such concepts however is not justified as it automatically eliminates concepts c_p that do not belong to a predefined user profile but have a high rank value, and thereby are of likelihood to be accessed by user. The experiments showed that this forced exclusion can increase the error-rate for the recommended items due to the fact that users are free to move from one locality to another. Thereby, the approach of re-evaluating rank of concepts supported by active user profile provides a smoother transition from general interest prediction to user-specific one used for recommendation generation herein. However, greater values of $u > 2.2$ caused a decline in prediction accuracy as it eliminated such unsupported concepts at all. Also, in some cases the decline was caused by the fact that a given concept had not yet been classified into the proper profile. On the other hand, low values of factor $u < 1.5$ did not produce significant lift for concepts belonging to an interest profile to appear in the final results of the computed recommendation.

Comparing the results of the original prediction engine, where no information about user profile was applied (Table 7-1, u=1) to the recommendation approach, where knowledge about active user profile is applied to refine predictions, a slight improvement was gained for algorithms A1, A2, and A3. The experiments with different values of factor u=1.8 and u=2.0 showed a small improvement in predicting user activity for algorithms as following: for A1 +1.1%, A2 +1.1%, where only one item of interest was proposed for recommendation, and algorithms quite restricted in item selection. For the algorithm A3 an improvement of 31.0% was gained, which is already a remarkable result. For algorithms A4, A5 and A6 the improvement in predicting user's next action was even better, up to 33.6%. This was quite expected, as the result set, where the actual visited item might have lied in, was greater (depending on the algorithm, either three or four items). Also, it is evident that the task to predict one item in comparison to a set of multiple items is rather troublesome, if the set of choices is of significant size. Table 7-1 presents the results of the experiments with different values of factor *u* for algorithms A1–A6, providing a comparison on the original prediction from the real time prediction engine (*u*=1), and gained improvement when applying detected user interest concepts for prediction refinement while establishing a set of recommended items for web personalization.

Table 7-1. *Improvement gained by refining prediction with knowledge of user profile.*

u	A1	A2	A3	A4	A5	A6
1.0	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
1.3	0.9%	0.8%	30.7%	18.2%	16.6%	33.2%
1.5	1.0%	0.9%	30.7%	18.3%	16.7%	33.4%
1.8	1.1%	0.9%	30.9%	18.5%	16.7%	33.5%
2.0	0.9%	1.1%	31.0%	18.3%	17.0%	33.6%
2.2	1.1%	0.8%	30.7%	18.1%	16.6%	33.4%
2.5	1.0%	0.7%	30.4%	17.7%	16.5%	33.1%
3.0	–1.8%	0.9%	27.1%	15.0%	16.0%	29.5%

The experiments, conducted with conceptual prediction engine having access to the knowledge available in user profile in addition to the user conceptual interests model, clearly showed that applying current user profile during establishing recommendation improves the prediction rate for visitor future activities, and therefore the selection of recommended items. This improvement thereby proves the approach of the recommender system to be successful.

7.4 Exploiting the Recommendation Result

Having established a set of recommended items – pages that might be of interest to an online user, it is time to apply tactical web adaptation to deliver the recommendation results to online user and thereby personalize the web session.

The adaptation process is carried out by the WIS management unit, responsible for handling users' requests and sending a response in a form of a webpage (Figure 7-1). The modifications introduced during tactical adaptation as a result of generated recommendations are short-term and applied in real-time. Therefore, the recommendation computation as well as its application must be fast enough not to cause any significant delays in page composition. In other words, users must never be kept waiting because of the personalization process. If the recommendation computation takes longer than normal page load would allow, the adaptation can be delivered through asynchronous web technologies. The modifications introduced as tactical adaptation to deliver recommendation results are user-centric and effective for a particular online user only at the time of page access.

There are numerous ways of exploiting the recommendation results for personalizing web sessions and thereby users browsing experiences. In its simplest form, a list of recommendations is provided for online user, typically in a special area on web page designed for the purpose. The presentation of personalization may also involve a refined navigation menu topology (recommended items added), highlighting of recommended items (e.g., by colour, size) to draw more user attention on them, provision of one-click shortcuts to recommended items, displaying related adverts, pulling related information from other (external) sources, such as the Web or web services, or any other means of dynamic adaptation. Thus, the reaction to the detected recommended items in the WIS management unit varies from simple methods of drawing user attention to certain elements to providing additional value and assisting user while browsing.

In terms of navigation menu, items not already present there for the current view can be added. However, these temporarily added items should be somehow distinguished. A good style example for instance is to divide the menu section into two, outlining the recommendations as '*See also'*, '*Recommended to see*', '*Related information*', '*Consider also*', '*Other users found useful*', and so on. Even a line between the original set of menu items and the recommended items can provide necessary separation of those two item groups. Distinguishing the original schema and added items is necessary to diminish the risk to get returning users confused in case they return to a page through a direct link or through an alternative path, and end up seeing only the original navigation schema without or with different set of recommendations. If the recommendation involves items already available in the layout, a simple technique is to highlight these elements. In this case, personalization is delivered through making some items more visible compared to other elements in the layout, only for a particular online user.

Yet another way to employ recommendation results is to have pages in the WIS assigned promotional resources to direct and stimulate users to further explore certain content. For instance, a user is being classified as a graduating student browsing the DCE website and therefore a page offering topics for thesis is found of interest for that user as a result of recommendation computation. In the WIS, there is an advertising content assigned to that recommended page – a banner for example. This content is now displayed to the user to stimulate browsing of that content. The user interface layout usually accommodates a special area for presenting such information. The latter example leads to one of the main drivers of web personalization – namely marketing and advertising. The described framework is suitable for delivering targeted online advertising based on detected user type and interests.

During web adaptation process the original structure (navigation) of website should remain intact. This requirement is essential, as the aim is to provide additional value in the form of information users might be looking for, and not to get users confused by random changes in the general structure or layout. Nevertheless, personalization is also used for filtering content and thereby delivering only the information found to be relevant and important to user by the system, discarding everything else. There are two main dangers involved with this approach. First of all, no recommendation engine is precise enough to fully and accurately comprehend human needs at a particular moment in time and space. Another thing is that if only targeted information is presented, presuming the recommendation is correct, users are not provided with alternatives, and it narrows down their possibilities and cognition of the information domain as a whole. A really good example of this is the use of personalization in search engines. For instance, you find an item of your interest in search results on the first page, in top 5 for example. Now, you switch computer or re-run your query later to find out that the same system profiles you a bit differently at this time and you cannot find the same result on the first page in top 5, as you were expecting. Isn't that frustrating?

Despite the ability to provide customized views to web content, as a result of web personalization, users may or may not choose to follow that information. Analysing users' actions in the context of provided recommendations would enable to evaluate different approaches of personalization, their delivery (presentation), and their efficiency. The log system proposed in this thesis, coupled with additional information about the recommendation result, can be used for the task. Regardless that this is an interesting topic of investigation, especially in the area of human-computer interaction (HCI), the evaluation of personalization methods is out of the scope of this chapter, and remains an interest of future studies for the author.

7.5 Chapter Summary

Web personalization is an emerging trend to manage the amount of available information and to deliver content relevant to user. It is achieved through the use of recommender systems and tactical adaptation of the web made available to target user. Even though the technological base today would enable to provide personalization without borders everywhere, its availability is quite limited, especially for anonymous ad-hoc web users. On one hand, there are privacy issues that set restrictions in identifying users, and these will remain as the Internet is based on the assumption of anonymity, where one has the optional ability to identify himself/herself as necessary for consuming services. On the other hand, developing and maintaining such sophisticated systems is quite costly and the target group of anonymous users might not be so tempting and motivating in sense of marketing, e-commerce, or even social networks.

Still, recommender systems are useful for sites where a large amount of information is available through several categories, starting from corporate websites and ending up with specific systems such as online museums, online stores, movie databases, news and government portals, to help users manage the information overload and problems of successful browsing. The economic effect gained from such systems is obvious – optimized and personalized websites are a better way to maintain information resources. For users it means time saving and increased productivity. Site owners however gain trust and visitor satisfaction, accompanied by customer loyalty, and in e-commerce also profit from improved sales of goods, services and advertising. Easy access to valuable information guarantees users to return to website rather than seek for alternative sources, and thereby helps to maintain and improve the rate of visitors, a key factor of success on the World Wide Web, and thus a driver for profitability.

In this chapter, a method for generating recommendations for anonymous ad-hoc web users was proposed. The approach advantages of the method for learning users' interest and domain model, and a method of predicting users' actions based on a conceptual interest model. The recommendations are provided by combining the two latter methods. Using these two methods together in a recommender system for establishing recommendations, enables to specify rather general prediction results with information available in user profile, and thereby obtain a recommendation set that better conforms to online visitors' information needs and interests. Experiments conducted on real website have confirmed the effectiveness of this approach. The described recommendation generation method to deliver personalized view to web content is not limited to just profiling anonymous ad-hoc users but can also be exploited in systems operating on identifiable users about whom a lot more information can be included in user profile, or extended to other areas such as recommending items to buy in online shops, or e-government services to consume in state information portals.

Chapter 8
OTHER DEVELOPED WEB INFORMATION SYSTEMS

While the main scope of the thesis is set on understanding web users' behaviour, providing ways of improving web information systems, and enabling web personalization for anonymous ad-hoc web users, the author has also been actively researching the e-learning domain. Many of the methods described in the previous chapters are applicable also for this domain. For instance, the log system is already exploited to log users' actions in the learning management system e-EDU and the described methods for WIS evaluation and optimization are applicable to improve web-based learning environments and digital learning objects. A sort of recommendation service could be developed to recommend students available resources they otherwise seem to ignore.

This chapter provides a short overview of the thesis author's contribution to e-learning domain, where the work has been focused on enabling contemporary ways of teaching and learning. In particular, a learning management system e-EDU developed at the Department of Computer Engineering at Tallinn University of Technology is discussed together with several digital learning objects as interactive hands-on tools and web services for e-learning.

8.1 Introduction

Today, we are living the era where our society is largely dependent on information technology and computer-based systems. Devices equipped with microelectronics can be found everywhere – from computers, mobile phones and personal music players to kitchenware. Microelectronic systems have become integral part of our lives. These systems all need some sort of software to be run, binding together the two major disciplines of the information and communication technology (ICT) world. Software systems are found everywhere where computer systems are involved. Even more, software systems also provide a platform to build other software systems on them. These systems vary in their complexity and level of sophistication. Nevertheless, failures in them may frustrate masses of people. A bug in mp3-player software may affect and disappoint thousands of users, while it is not a critical problem compared to a fault in an airplane navigation system or nuclear power plant management system, where failures may have fatal consequences.

Obviously, if young system engineers are educated with great care from the beginning, using different teaching methods, including e-learning and interactive tools made available by the rapid development of the Internet and its associated technologies, their good skills and increased interest towards study field can be assured. Thereby, proper education of future engineers can mitigate some of the risks concerned with computer systems development.

The rapid development of the Internet and its associated technologies has enabled web-based training and distance learning to become a reality and made it possible to move studies fully or partially into virtual learning environments. This has also made possible creation of digital learning objects, varying from simple textual content to images, animations and interactive simulator tools. A learning object, as defined by the IEEE, is *any entity – digital or non-digital – that may be used for learning, education or training* [135]. Clearly, digital learning objects as interactive hands-on tools and web enabled learning management systems (LMS) are taking education to the next level and pose new challenges for teaching, as users can access such systems everywhere and any time where the Internet is present, regardless whether they are at home, in the office, sitting in the park with tablet PC or smart-phone, and so forth. This rapid evolution of web technologies has now enabled to develop learning management systems with various functionalities on an affordable level.

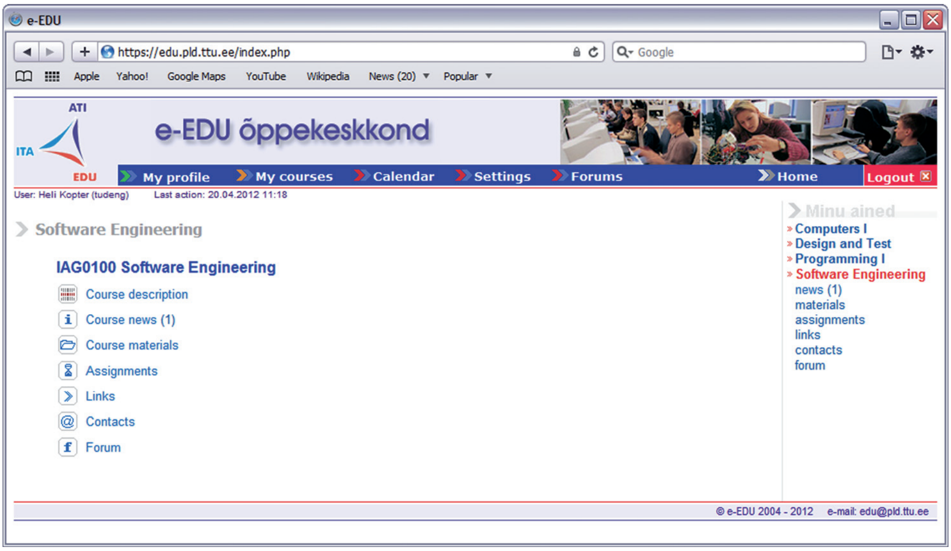
In this chapter, an overview about the authors work in the e-learning domain is provided. A learning management system e-EDU and several frameworks for creating digital learning objects are introduced. These works establish a platform for future scientific work and experiments in the e-learning domain as well as for studies of WIS usage and personalization with identifiable users.

8.2 A Lightweight Learning Management System e-EDU

Over the years, universities have developed various information systems including systems that were supposed to support learning and teaching processes via providing materials, links and other information valuable in learning process. However, most of these systems are concerned about making it possible to declare term subjects electronically, and maintaining students' assessment results. This has caused lecturers to search for applications suitable and available for them, or to develop their own systems for managing courses from the point of view of the learning-teaching process. As a result, students end up facing many heterogeneous web environments, if at all. Evidently, one coherent web-based environment, where the majority of courses students are taking or have taken are represented, is in everybody's best interest.

The e-EDU is a lightweight learning management system [120] used at the Dept. of Computer Engineering at Tallinn University of Technology (TUT) for ICT studies. The project was launched in 2002 with the aim to propose new and localized approach for daily and distance learning courses via intelligent use of modern technologies and consequently enliven and enrich ICT studies of the curricula of Computer Systems, Electronics, and Informatics at TUT.

Although, since it launch other learning management systems such as WebCT (now Blackboard[1], licence discontinued as of 2012) and later the popular course management system Moodle[2] were made available by the Estonian e-Learning Development Centre[3] (earlier known as Estonian e-University, founded in February 2003), e-EDU is still used as the primary learning environment for several hard- and software related subjects taught at the department. During this time, over 4 800 students have used the e-EDU throughout several courses. The e-EDU is available for students at https://edu.pld.ttu.ee. Figure 8-1 shows an overview of the e-EDU LMS.



User: Heli Kopter (tudeng) Last action: 20.04.2012 11:18

© e-EDU 2004 - 2012 e-mail: edu@pld.ttu.ee

Figure 8-1. *Screenshot from the e-EDU learning management system; a subject Software Engineering is selected in the view.*

Since its establishment, the e-EDU has followed the strict line to be accessible (a) with all commonly available web browsers, (b) without a need for additional plug-ins (e.g., Adobe FlashPlayer, ActiveX controls, Microsoft Silverlight, Oracle Java etc.) or software to be installed by learner, exceptions apply to special learning objects, and (c) regardless of platform being used. The e-EDU is based on the same kernel as the DCE website and has been implemented using PHP scripting language on the server side and Javascript on the client side, running on Apache web server, and using MySQL database engine for data storage. The log system introduced in Chapter 3 is also present in the e-EDU learning management system.

[1] http://www.blackboard.com
[2] http://moodle.org
[3] http://www.e-ope.ee

The following services are provided for students via the e-EDU LMS: course news, personalized assignments and progress feedback, course study materials by modalities and resources set, calendar with running academic due dates and task deadline reminder, course descriptions, lecturers' contacts, forum, and interactive hands-on tools. With the establishment and exploitation of the e-EDU LMS the following advantages have been gained:

- Courses are Internet present in a coherent environment,
- Ubiquitous access to courses is attained,
- Everything is reproducible in this environment,
- Communication and feedback on tasks,
- Reduced workload for lecturers,
- Native language interface,
- Intelligent use of ICT towards e-supported courses.

A detailed overview of the e-EDU system is available in [120] and a corresponding case study on moving studies to e-environments in [136]. In [137] personalized web services were proposed for the e-EDU, to provide easy access to course news and deadlines over the exploitation of RSS feeds. The e-EDU is averagely accessed by students 1.4 times a day and most of the accesses occur either during lecture hours or between 8PM and 10PM, though the log shows that students make use of it 24-hours a day. Detailed statistics about the e-EDU LMS have been published in [138].

In addition to the aforementioned advantages, with the introduction of the e-EDU teachers have less paperwork to handle in regards of maintaining students' results and assignments. All necessary aspects are covered with the e-EDU information system. Also, as students are usually divided into groups having different teachers, the results can be shared and are easily accessible to all lecturers involved with a particular course. Moreover, the progress of the course is now viewable in real-time, and different statistics can be generated about students and their success. Access to the e-EDU for teachers is provided via the department's intranet web information system called 'ITA'.

The concept and development of the e-EDU is mainly maintained by the author of this thesis. In particular, the kernel and the majority of the modules is developed by the author, with one exception – the credits for the assignments evaluation subsystem go to the author's colleague, research scientist Elmet Orasson. The development of the e-EDU continues.

8.3 Learning Objects as Interactive Hands-On Tools

Teaching information and communication technology is a complex area, compared to economics and social sciences, for example. There is a constant need to add value to theoretical knowledge presented in lectures. This can be achieved by using interactive hands-on tools that clearly demonstrate static knowledge in a dynamic form, exemplifying theory and enabling students to explore the problem domain through several scenarios. Several digital learning objects have been established by the author, and some under his supervision, mainly for first year bachelor level students in the fields of informatics, computers and basics of microelectronics. In [139] an interactive tool to demonstrate the behaviour of fundamental elements of digital logic (e.g., OR, AND, etc. gates) was established, and in [138] a learning object to demonstrate conversion between different numeric systems. These learning objects allow students to explore the problem domain either by entering input values and letting the system to demonstrate the problem, or setting the input and as well proposing the expected outcome, allowing students to check their knowledge with the help of an interactive tool. Interactive digital learning objects are meant to support the learning process.

In 2009 a concept of learning objects on web services was introduced with the e-EDU Web Services Initiative[1] (e-EDU WSI) [140], [141]. The e-EDU WSI provides a framework for developing interactive tools based on web services in the area of e-learning, targeting the following advantages:

- Single implementation of complex algorithms, multiple use in heterogeneous learning management systems;
- Modularity, as complex services can be composed of atomic ones;
- Interoperability by defining services via standard means;
- Realisation independency and freedom of a chosen LMS platform for learning objects;
- Enabling to supplement courses with different interactive tools, without excessive programming and costs for particular LMS;
- Automation as machine-to-machine interaction;
- Moving (heavy) computing away from client-side (e.g., Java applets) to distributed systems;
- Encapsulating service implementation (algorithms do not need to be publicly available, of importance is the use of certain functionality), with possibility of wide public use.

Web services provide standard means of communication among different software applications enabling platform and language independent interoperable networking over the Internet using standard formats [142]. Bringing web services to e-learning would enable to compose courses of learning objects distributed all over the Web. Still, there are only a few LMS's allowing to consume services or providing learning objects as web services. Appendix G outlines the framework for developing digital learning objects on web services.

Under the e-EDU WSI two web services have been made available: a web service for conversion calculator between different numeric systems, and a service for exploring behaviour of digital logic gates. Both services are implemented as SOAP and JSON servers, and coupled with example clients. Currently, a mobile application on 'Introducing the Basics of Digital Logic' on

[1] http://www.pld.ttu.ee/edu/WSI/

the Android platform for smart-phones is being developed under the supervision of the thesis author. With the wide availability and increasing use of smart-phones, digital learning objects and also learning management systems need to be made available on these platforms as well.

In [143] a learning object based on domain knowledge described by means of ontology was proposed to demonstrate the behaviour of fundamental elements of digital logic. The main didactic aim of this intelligent tool was set on teaching perception of the basics of digital logic, described from various aspects by means of ontology, illustrated with gate behaviour, notations, truth tables, and relevant details. This tool is currently under development.

8.4 Chapter Summary

Living the era, where the Internet has become an integral part of our lives, education of young system engineers also needs to be provided with following the latest trends in technology. Virtual learning environments and learning management systems accessible over the Web regardless of time and location are an essential part of today's teaching-learning process, enabling both for students contemporary ways of learning and for teachers variety of ways to communicate knowledge. Interactive digital learning objects also have a crucial role to play in educating future engineers, as they allow students to investigate problem domain on their own, and besides static theory presented in the lectures obtain a view to the knowledge in a dynamic form.

Many of the methods and frameworks discussed in Chapters 3–7 are applicable also for the e-EDU LMS, where users need to log in and identify themselves. For instance, the log system introduced in Chapter 3 has already been included in the system. The methods of user action prediction and recommendation generation can be used to recommend learning resources, or in digital learning objects to direct students to explore different aspects of provided tools.

In this chapter an overview of the author's contribution to the e-learning domain was discussed. A lightweight learning management system e-EDU, called into life and mostly maintained by the author, was introduced, followed by a discussion on developed digital learning objects. A framework for developing learning objects on web services as a way to provide ubiquitous access to functionality of educational programs for establishing learning objects in heterogeneous environments was introduced, accompanied with example web services. Also, an interactive tool on domain ontology for microelectronics education was outlined. This research has established a good foundation for future studies and experiments for both e-learning and WIS's development. The work on the e-EDU LMS and digital learning objects continues as the author is actively involved in teaching future system engineers.

Chapter 9
CONCLUSIONS AND FUTURE WORK

"Any sufficiently advanced technology is indistinguishable from magic"

Sir Arthur C. Clarke[1]

Living the Internet-enabled era, users have gained ubiquitous access to information available in various forms. However, this has also posed problems of information overload and successful information retrieval, challenging developers and researchers all over the world to find new and user-centric approaches to deliver users the useful information they are searching for or might be in search of.

This thesis has focused on understanding users and their needs with the aim to deliver improved web experience and enabling web personalization for anonymous ad-hoc web users, taking yet another step towards adaptive web.

This chapter summarises the thesis and provides directions for future work.

9.1 Conclusions

The rapid development of technologies has raised the importance of the Interned and the World Wide Web in our lives to a level it has never been before. The Internet is now present not only in computers but also on various platforms from smart-phones to TVs – the access to the Internet has become ubiquitous. This process has also had an effect on expectations users have towards the Web, where information is assumed to be easily found in a fast and convenient manner. The information overload problem and users' expectations towards the Web have led to a need for web information systems optimization and personalization, where implicit approaches seamless to users are applied and knowledge obtained from users' behaviour. The era of the 'one-size-fits-all' browsing paradigm is coming to an end.

[1] Sir Arthur C. Clarke (1917-2008), British physicist and science fiction author. Citation also known as Clarke's third law. Origin: Profiles of the Future (revised edition, 1973).

This thesis provides several frameworks to improve and personalize web information systems by advantaging from users' behaviour modelling. Methods of obtaining users' behaviour data, together with a development of a special log and analyzer system, have been investigated. The developed log system has been used to collect users' behaviour data for research on WIS improvement, and users' behaviour modelling, resulting in methods for obtaining users' domain models and profiles, methods for predicting users' forthcoming behaviour, and a framework for delivering web recommendations for anonymous ad-hoc web users as a personalization service. This research has focused on anonymous users – users about whom the system does not maintain any explicit knowledge. The latter distinguishes this research from works of other scientist of the field, where researchers have mainly focused on users who are explicitly identifiable by some means. To the best of the author's knowledge, the novel methods described in the thesis were not reported at the time these web studies were initiated.

As the author has been actively involved also in developing and maintaining web information systems and applications in the e-learning domain, the thesis also provides a short overview of the author's research in that field with outlining notable contributions.

The main contributions of the thesis are as follows:

- Establishment of an original log system to capture users' web activity, and an analyzer system to process this data into a knowledge base. As an increasing number of users access information on the Web, collecting and analyzing such data provides a great opportunity to learn from users.
- A methodology providing a new approach to evaluate and optimize web information systems through various metrics based on knowledge obtained from users' behaviour modelling. The methodology targets to minimize the conceptual gap between the model applied during web information system development and the one used by its actual users.
- A novel framework for learning users' domain models. The framework delivers a method for learning web user profiles that combines browsing behaviour modelling with ontology based profiling. The approach takes advantage from giving conceptual meaning to web usage mining results by using ontologies and automatic classification of concepts to ontologies via reasoning.
- A methodology to model web users' behaviour for anonymous visitor activity prediction based on collective intelligence. Two novel models for action prediction were proposed, out of which conceptual modelling of users' interests proved to perform better than the probabilistic sequential model. The methodology results in a conceptual interest model for web users' action prediction.

- A framework for establishing viewing recommendations for anonymous ad-hoc web users and delivering web personalization for such users. The method for computing recommendations advantages from ontology-based modelling of user profiles and user action prediction based on conceptually modelled users' interests. The framework delivers a new, lightweight and effective method for recommender system to enable web personalization for anonymous ad-hoc web users via web adaptation.
- An original learning management system e-EDU and a set of digital learning objects to improve learning-teaching process and enable contemporary ways of learning for students of information and communication technology at Tallinn University of Technology.

The delivered approaches for user profiling and recommendation generation are oriented on anonymous web users. However, they are also applicable in systems where users are identifiable and additional information about them is easily available. The target of the described methods was set to enable personalization on websites delivering front-end to various medium and large-sized web information systems, such as corporate web and information portals (e.g., governmental e-service or news portals). Yet, there are no restrictions to exploit these methods in other areas such as social networking, online banking, and applications of e-commerce. The discussed methods are also applicable for the e-learning domain.

The economic benefit gained from such intelligent and adaptive web information systems is twofold. Users gain access to information resources optimized and personalized for them, facilitating information search and browsing, while offering increased productivity and decrease in search time. Information that is easily accessible, logically structured and navigable, and maintained in a way users expect to find it, contributes to visitors browsing activity and thereby also affects users' decisions over the necessity of sought information or suitability of a product. Web-based environments where visitors feel welcomed and that are convenient to use, guarantee users to return rather than to seek for alternative sources for information, and therefore help to maintain and improve the rate of visitors, a key factor of success on the World Wide Web. Website owners and entrepreneurs on the other hand benefit from visitor satisfaction and increased engagement, accompanied by customer loyalty, and in e-commerce also profit from improved sales of goods and services, for instance online advertising. Such intelligent and adaptive systems allow customization of visitor-company interaction, and thereby make visitors sense they are personally valuable for company. Moreover, such systems enable to bring users to certain information (e.g., advertisements) they otherwise might disregard. The more is known about a user, the more personalized content can be delivered, and if properly handled, this knowledge can be turned into profit.

Through its contributions, the thesis has addressed both of the issues: web information systems improvement and personalization based on users' behaviour towards adaptive web.

9.2 Future Work

The ideas and issues to be considered to further improve and advance the provided methods constituting the domain of future work are as follows:

- Further development of the log system and analyzer system by adding features that enable collecting of web usage log over web services, hosted event logging, and advantaging of features of the HTML5 (e.g., DOM storage) standard. Extend the log system to enable capturing of web events cross-domain. In terms of web service based usage logging, a prototype tool has been established by Tanel Kerstna in his bachelor thesis 'Applying Web Services for Log System Implementation' (2011) under the supervision of the author of this dissertation. This development work continues.
- Improve the methods of capturing time spent on page by adding active TSP detection, and enabling approximate TSP discovery for the exit page. This is to be based on the new features made available with the HTML5 standard and its support in modern web browsers.
- Establishment of an online user profiler for instant detection of active user profile based on users' domain ontology.
- Study the possibilities to include and model different demographic aspects of probable users (e.g., language profiles) in web ontology.
- Investigate the possibilities to include more collective intelligence information, such as user interest indicators to improve the prediction models, and thereby also recommendation generation for personalization services.
- Establish a new adaptive DCE WIS (in parallel to existing) where the methods delivered in this thesis are made available for general public. The system is to be used for the evaluation of delivered web recommendation and personalization methods for anonymous ad-hoc web users through long-time empirical studies.
- Continuation of research and development in the e-learning domain. Further improvement of the e-EDU learning management system and development of new digital learning objects.

The research described in this thesis has paved a way for further studies on web information systems improvement and web personalization. The ideas provided for future work are just some of the possible research activities to be considered in the nearest future as a continuation of the research covered by the dissertation.

REFERENCES

[1] Isakowitz, T., Bieber, M., Vitali, F. Web Information Systems. *Communications of the ACM*, 1998, 41(7), 78-80.

[2] Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web. *Scientific American*, 2001, 284(5), 34–43.

[3] Shadbolt, N., Berners-Lee, T., Hall, W. The Semantic Web Revisited. *IEEE Intelligent Systems*, 2006, 21(3), 96-101.

[4] Bernard, M. L. User Expectations for the Location of Web Objects. *Proceedings of CHI'01 Conference: Human Factors in Computing Systems*, Seattle, USA, 2001, pp. 171-172.

[5] Geissler, G., Zinkhan, G., Watson, R. Web Home Page Complexity and Communication Effectiveness. *Journal of the Association for Information Systems*, 2001, 2(2), 1-48.

[6] Bernard, M.L., Chaparro, B.S. Searching within websites: A Comparison of Three Types of Sitemap Menu Structures. *Proceedings of The Human Factors and Ergonomics Society 44th Annual Meeting in San Diego*, 2000, pp. 441-444.

[7] Lee, A.T. Web Usability: a Review of the Research. *ACM SIGCHI Bulletin*, 1999, 31(1), 38-40.

[8] Speretta, M., Gauch, S. Personalized Search based on User Search Histories. *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, IEEE Computer Society, Washington, DC, USA, 2005, pp. 622-628.

[9] Mulvenna, M.D., Anand, S.S., Buchener, A.G. Personalization on the net using web mining. *Communication of ACM*, 2000, 43(8), 122-125.

[10] Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., Reidl, J. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. *Proceedings of ACM Conference on Computer Supported Collaborative Work (CSCW)*, Seattle, Washington, USA, 1998, pp. 345-354.

[11] Middleton, S.E., Shadbolt, N.R., De Roure, D.C. Capturing Interest through Inference and Visualization: Ontological User Profiling in Recommender Systems. *Proceedings of the 2nd International Conference on Knowledge Capture*, Sanibel Island, FL, USA, 2003, pp. 62-69.

[12] Shapira, B., Taieb-Maimon, M., Moskowitz, A. Study of Usefulness of Known and New Implicit Indicators and Their Optimal Combination for Accurate Inference of Users Interest. *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC '06)*, Dijon, France, 2006, pp. 1118-1119.

[13] Srivastava, J., Cooley R., Deshpande M., Tan P.N. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *ACM SIGKDD Explorations*, 2000, 1(2), 12-23.

[14] Al halabi, W.S., Kubat, M., Tapia, M. Time Spent on a Web Page is Sufficient to Infer a User's Interest. *Proceedings of the IASTED European Conference: internet and multimedia systems and applications*, Chamonix, France, 2007, pp. 41-46.

[15] Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A. User profiles for personalized information access. *The adaptive web*, P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.). LNCS, vol. 4321. Springer-Verlag, Berlin, Heidelberg, 2007, pp. 54-89.

[16] Goecks, J., Shavlik, J. Learning users' interests by unobtrusively observing their normal behavior. *Proceedings of the 5th international conference on Intelligent user interfaces (IUI '00)*, ACM, New York, NY, USA, 2000, pp. 129-132.

[17] Lieberman, H. Letizia: An Agent That Assists Web Browsing. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, 1995, pp. 924-929.

[18] Claypool, M., Le, P., Wased, M., Brown, D. Implicit interest indicators. *Proceedings of the 6th international conference on Intelligent user interfaces IUI'01*, ACM Press, New York, NY, USA, 2001, pp. 33-40.

[19] Davison, B. Web Traffic Logs: An Imperfect Resource for Evaluation. *Proceedings of Ninth Annual Conference of the Internet Society (INET '99)*, San Jose, CA, 1999, (available at http://www.isoc.org/inet99/proceedings/4n/4n_1.htm)

[20] Mobasher, B., Cooley, R., Srivastava. J. Automatic Personalization Based on Web Usage Mining. *Communications of the ACM*, 2000, 43(8), 142-151.

[21] Kimball, R., Margy, R. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modelling*. John Wiley & Sons, 2002, 2nd ed., 464 p.

[22] Tan, P.N., Steinbach, M, Kumar, V. *Introduction to data mining*. Pearson Addison Wesley, 2006, 769p.

[23] Etzioni, O., The World Wide Web: Quagmire or Gold Mine?. *Communications of ACM*, 1996, 39(11), 65-68.

[24] Srivastava, J., Desikan, P., Kumar, V. Web Mining: Accomplishments and Future Directions. *Proc. US Nat'l Science Foundation Workshop on Next-Generation Data Mining (NGDM)*, Nat'l Science Foundation, 2002, pp. 51-70.

[25] Berendt, B., Hotho, A., Mladenic, D., Someren, M., Spiliopoulou, M., Stumme, G. A Roadmap for Web Mining: From Web to Semantic Web. *Proceedings of EWMF'2003*, LNCS, vol. 3209, 2004, pp.1-22.

[26] Kosala, R. Blockeel, H. Web Mining Research: A Survey. *ACM SIGKDD Explorations*, 2000, 2(1), 1-15.

[27] Liu, B. *Web data mining: exploring hyperlinks, contents, and usage data*, Springer, 2007, 532p.

[28] Markov, Z., Larose, D. *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*, Wiley, 2007, 218p.

[29] Li, Y. Zhong, N. Mining Ontology for Automatically Acquiring Web User Information Needs. *IEEE Transactions on Knowledge and Data Engineering*, 2006, 18(4), 554-568.

[30] Mobasher, B. Data Mining for Web Personalization. *The adaptive web*, P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.). LNCS, vol. 4321. Springer-Verlag, Berlin, Heidelberg, 2007, pp. 90-135.

[31] Eirinaki, M., Vazirgiannis, M. Web mining for web personalization. *ACM Transactions on Internet Technology (TOIT)*, 2003, 3(1), 1-27.

[32] Perkowitz, M. Etzioni, O. Adaptive web sites: Concept and case study. *Artificial Intelligence*, 2001, Vol. 118 (1-2), pp. 245-275.

[33] Baraglia, R., Silvestri, F. Dynamic personalization of web sites without user intervention. *Communications of ACM*, 2007, 50(2), 63-67.

[34] Perkowitz, M., Etzioni, O. Towards adaptive Web sites: conceptual framework and case study. *Artificial Intelligence*, 2000, 118(1-2), 245-275.

[35] Shahabi, C., Banaei-Kashani, F. Efficient and Anonymous Web-Usage Mining for Web Personalization. *INFORMS Journal on Computing*, 2003, 15(2), 123-147.

[36] Resnick, P., Varian, H.R. Recommender systems. *Communications of ACM*, 1997, 40(3), 56-58.

[37] Pazzani, M., Muramatsu, J., Billsus, D. Syskill&Webert: Identifying Interesting Web Sites. *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 1 (AAAI'96)*, vol. 1, AAAI Press, 1996, pp. 54-61.

[38] Joachims, T., Freitag, D., Mitchell, T. WebWatcher: A Tour Guide for the World Wide Web. *Proc. of International Joint Conference on Artificial Intelligence*, Nagoya, Japan, Morgan Kaufmann, 1997, pp. 770-775.

[39] Guarino, N., Masolo, C., Vetere, G. OntoSeek: Content-based Access to the Web. *IEEE Intelligent Systems*, 1999, 14(3), 70-80.

[40] Markellou, P., Mousouroulli, I., Spiros, S., Tsakadilis, A. Using Semantic Web Mining Technologies for Personalized e-Learning Experiences. *Proceedings of The IASTED International Conference on Web-based Education (WBE 2005)*, V. Uskov (ed.), Grindelwald, Switzerland, 2005, pp. 461-826.

[41] Zaiane, O.R. Building a Recommender Agent for e-Learning Systems. *Proceedings of the International Conference on Computers in Education (ICCE '02)*, IEEE CS, Washington, DC, USA, 2002, pp. 55-60.

[42] Bobadilla, J., Hernando, A., Arroyo, A. e-Learning Experience Using Recommender Systems. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE '11)*, ACM, New York, NY, USA, 2011, pp. 477-482.

[43] Tiroshi, A., Kuflik, T., Kay, J., Kummerfeld, B. Recommender Systems and the Social Web. *Advances in User Modeling: selected papers from UMAP 2011 workshops*, LNCS, vol. 7138, 2012, pp. 60-70.

[44] Fijalkowski, D., Zatoka, R. An architecture of a web recommender system using social network user profiles for e-commerce. *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2011, pp.287-290.

[45] Ma, H., Zhou, T. C., Lyu, M. R., King, I. Improving Recommender Systems by Incorporating Social Contextual Information. *ACM Transactions on Information Systems (TOIS)*, 2011, 29(2), Article 9, 23 p.

[46] Hannon, J., Bennett, M., Smyth, B. Recommending twitter users to follow using content and collaborative filtering approaches. *Proceedings of the fourth ACM conference on Recommender systems (RecSys '10)*, ACM, New York, NY, USA, 2010, pp. 199-206.

[47] Ma, H., Zhou, D., Liu, C., Lyu, M. R., King, I. Recommender systems with social regularization. *Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11)*, ACM, New York, NY, USA, 2011, pp. 287-296.

[48] Official Google Blog: Search, plus Your World, (2012), Retrieved from http://googleblog.blogspot.com/2012/01/search-plus-your-world.html

[49] Guy, I., Jaimes, A., Agulló, P., Moore, P., Nandy, P., Nastar, C., Schinzel, H.. Will Recommenders Kill Search?: Recommender Systems - an Industry Perspective. *Proceedings of the fourth ACM conference on*

Recommender systems (RecSys '10), ACM, New York, NY, USA, 2010, pp. 7-12.

[50] Gaudioso, E., Boticario, J.G. User Modeling on Adaptive Web-Based Learning Communities. *7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II (KES 2003)*, V. Palade, R.J. Howlett, L.C. Jain (eds.), LNCS, vol. 2774, Springer Berlin / Heidelberg, 2003, pp. 260-266.

[51] Hofgesang, P. Web Personalization Through Incremental Individual Profiling and Support-Based User Segmentation. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '07)*, IEEE Computer Society, 2007, pp. 213-220.

[52] Cetintemel, U., Franklin, M.J., Giles, C.L. Self-Adaptive User Profiles for Large-Scale Data Delivery. *16th International Conference on Data Engineering (ICDE'00)*, IEEE Computer Society, Washington, DC, USA, 2000, pp. 622-633.

[53] Trajkova, J., Gauch, S. Improving Ontology-Based User Profiles. *Proceedings of 7th RIAO Conference*, Vaucluse, France, 2004, pp. 380-389.

[54] Castellano, G., Fanelli, A.M., Mencar, C., Torsello, M.A. Similarity-Based Fuzzy Clustering for User Profiling. *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Workshops*. IEEE Computer Society, Washington, DC, USA, 2007, pp. 75-78.

[55] Sutterer, M., Droegehorn, O., David, K. User Profile Selection by Means of Ontology Reasoning. *Proceedings of the 2008 Fourth Advanced International Conference on Telecommunications (AICT '08)*, IEEE Computer Society, Washington, DC, USA, 2008, pp. 299-304.

[56] Joung, Y., Zarki, M.E., Jain, R. A User Model for Personalization Services. *4th Int. Conference on Digital Information Management, ICDIM*, IEEE, 2009, pp. 247-252.

[57] Pan, J., Zhang, B., Wu, G. A SAM-Based Evolution Model of Ontological User Model. *Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science*, IEEE Computer Society, Washington, DC, USA, 2009, pp. 1139-1143.

[58] Sieg, A., Mobasher, B., Burke, R. Web Search Personalization with Ontological User Profiles. *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, ACM, New York, NY, USA, 2007, pp. 525-534.

[59] Pan, J., Zhang, B., Wang, S., Wu, G., Wei, D. Ontology Based User Profiling in Personalized Information Service Agent. *Proceedings of the 7th IEEE International Conference on Computer and Information*

Technology CIT '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 1089-1093.

[60] He, S., Fang, M. Ontological User Profiling on Personalized Recommendation in e-Commerce. *Proceedings of the 2008 IEEE International Conference on e-Business Engineering ICEBE '08*, IEEE Computer Society, Washington, DC, USA, 2008, pp. 585-589.

[61] Gruber, T. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal on Human and Computer Studies*, 1995, 43(5/6), 907-929.

[62] Guarino, N. Formal Ontology in Information Systems. *First International Conference on Formal Ontology in Information Systems (FOIS΄98)*, Guarino, N. (ed). IOS Press, Amsterdam, 1998, pp. 3-15.

[63] OWL Web Ontology Language, (2012), Available at http://www.w3.org/TR/owl-features/

[64] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. *The Description Logic Handbook*, Cambridge Univ. Press, 2003, 574p.

[65] Noy, N.F., McGuinness, D.L. Ontology Development 101: A Guide to Creating Your First Ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, 2001, Retrieved from http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf

[66] Haav, H.-M. A practical methodology for development of a network of e-government domain ontologies. *Building the e-World Ecosystem: 11th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2011*, IFIP Advances in Information and Communication Technology, vol. 353, Springer Boston, 2011, pp. 1-13.

[67] Atterer, R., Schmidt, A. Tracking the interaction of users with AJAX applications for usability testing. *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07).* ACM, New York, NY, USA, 2007, pp. 1347-1350.

[68] Velayathan, G., Yamada, S. Behavior-Based Web Page Evaluation. *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology (WI-IATW '06).* IEEE Computer Society, Washington, DC, USA, 2006, pp. 409-412.

[69] Shen, X., Tan, B., Zhai, C.X. Implicit User Modeling for Personalized Search. *Proceedings of the 14th ACM international conference on Information and knowledge management CIKM '05*, ACM, New York, NY, USA, 2005, pp. 824-831.

[70] Baglioni, M., Ferrara, U., Romei, A., Ruggieri, S., Turini, F. Preprocessing and Mining Web Log Data for Web personalization. *8th Italian Conference on Artificial Intelligence*, LNCS, vol. 2829, Springer, Heidelberg, 2003, pp. 237-249.

[71] Lim, E-P., Sun, A. Web Mining – the Ontology Approach. *The International Advanced Digital Library Conference (IADLC'2005)*, Nagoya, Japan, 2005, Retrieved from http://iadlc.nul.nagoya-u.ac.jp/archives/IADLC2005/Ee-Peng.pdf

[72] Chaffee, J., Gauch, S., Personal Ontologies for Web Navigation. *Proceedings of 9th International Conference on Information and Knowledge Management (CIKM'00)*, ACM, New York, NY, USA, 2000, pp.227-234.

[73] Gauch, S., Chaffee, J., Pretschner, A. Ontology-based personalized search and browsing. *Journal of Web Intelligence and Agent Systems*, 2003, 1(3-4), 219-234.

[74] Godoy,D., Amandi, A. User profiling in personal information agents: a survey. *The Knowledge Engineering Review*, 2006, 20(4), 329-361.

[75] Godoy, D., Amandi, A. Modeling user interests by conceptual clustering. *Information Systems*, Elsevier Science Ltd., 2006, 31(4), 247-265.

[76] Jung, H., Yang, J., Choi, J. Ontology-Based Web Navigation Assistant. *Proceedings of IDEAL 2004*, Z.R. Yang et al. (eds), LNCS, vol. 3177, Springer, 2004, pp. 443-448.

[77] Liu, W., Jin, F., Zhang, X. Ontology-Based User Modeling for E-Commerce System. *Third International Conference on Pervasive Computing and Applications, ICPCA 2008*, vol.1, 2008, pp.260-263.

[78] Yang, Y., Claramunt, C., Aufaure, M.-A. Towards a DL-Based Semantic User Model for Web Personalization. *Proceedings of the Third International Conference on Autonomic and Autonomous Systems ICAS '07*, IEEE Computer Society, 2007, pp. 61-66.

[79] Achananuparp, P., Han, H., Nasraoui, O., Johnson, R. Semantically Enhanced User Modeling. *Proceedings of the 2007 ACM Symposium on Applied Computing SAC '07*, ACM, New York, NY, USA, 2007, pp. 1335-1339.

[80] Ahn, J., Brusilovsky, P., Grady, J., He, D., Syn, S.Y. Open User Profiles for Adaptive news Systems: Help or Harm?. *Proceedings of the 16th international conference on World Wide Web, WWW '07*, ACM, New York, NY, USA, 2007, pp. 11-20.

[81] Sieg, A., Mobasher, B., Burke, R.D. Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. *Proceedings of IEEE Intelligent Informatics Bulletin*, vol.8, no.1, 2007, pp. 7-18.

[82] Garcia-Molina, H., Koutrika, G., Parameswaran, A. Information seeking: convergence of search, recommendations, and advertising. *Communications of ACM*, 2011, 54(11), 121-130.

[83] Olsen, K., Malizia, A. Automated Personal Assistants, *IEEE Computer*, 44(11), IEEE Computer Society, 2011, 110-112.

[84] Korth, A., Plumbaum, T. A Framework for Ubiquitous User Modeling. *IEEE International Conference on Information Reuse and Integration*, 2007, pp. 291-297.

[85] Cooley, R., Tan, P.N., Srivastava, J. WebSIFT: The Web Site Information Filter System. *Proc. of the Web Usage Analysis and User Profiling Workshop (WEBKDD'99)*, 1999.

[86] Zhou, B. Hui, S.C., Chang, K. An Intelligent Recommender System Using Sequential Web Access Patterns. *IEEE Conference on Cybernetics and Intelligent Systems*, IEEE Computer Society, 2004, pp. 393-398.

[87] Middleton, S., Roure, D.D, Shadbolt, N. Capturing Knowledge of User Preferences: Ontologies in Recommender Systems. *First Int. Conference on Knowledge Capture (K-CAP '01)*, ACM Press, New York, NY, USA, 2001, pp. 100-107.

[88] Middleton, S., Shadbolt, N., Roure, D.D. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 2004, 54-88.

[89] Middleton, S., Alani, H., Shadbolt, N., Roure, D.D. Exploiting Synergy Between Ontologies and Recommender Systems. *Proceedings of the 11th International World Wide Web Conference WWW, Workshop on Sematic Web*, 2002, pp. 41-50.

[90] Géry, M., Haddad, H. Evaluation of web usage mining approaches for user's next request prediction. *Proceedings of the 5th ACM international workshop on Web information and data management (WIDM '03)*, ACM, New York, NY, USA, 2003, pp. 74-81.

[91] Buriano, L., Marchetti, M., Carmagnola, F., Cena, F., Gena, C., Torre, I. The Role of Ontologies in Context-Aware Recommender Systems. *MDM '06: Proc. of the 7th Int. Conference on Mobile Data Management*, IEEE Computer Society, 2006, pp. 80.

[92] Taghipour, N., Kardan, A., Ghidary, S.S. Usage-Based Web Recommendations: A Reinforcement Learning Approach. *Proceedings of the 2007 ACM conference on Recommender systems (RecSys'07)*, ACM, New York, NY, USA, 2007, pp. 113-120.

[93] Zhao, S., Du, N., Nauerz, A., Zhang, X., Yuan, Q., Fu, R. Improved recommendation based on collaborative tagging behaviors. *Proceedings of*

the 13th international conference on Intelligent user interfaces (IUI '08), ACM, New York, NY, USA, 2008, pp. 413-416.

[94] Wang, R-Q., Kong, F-S. Semantic-Enhanced Personalized Recommender System. *Proceedings of the Sixth International Conference on Machine learning and Cybernetics*, 2007, vol. 7, pp. 4069-4074.

[95] Bonino, D., Corno, F., Squillero, G. A Real-Time Evolutionary Algorithm for Web Prediction. *Proceedings of the 2003 IEEE / WIC / ACM International Conference on Web Intelligence (WI'03)*, IEEE Computer Society, Washington, DC, USA, 2003, pp. 139-145.

[96] Basu, C., Hirsh, H., Cohen, W. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. *Proceedings of the 15th International Conference on Artificial Intelligence (AAAI'98)*, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1998, pp. 714-720.

[97] Messina, E., Toscani, D., Archetti, F. UP-DRES User Profiling for a Dynamic REcommendation System. *Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, LNAI, vol. 4065, Springer, 2006, pp. 146-160.

[98] Sarwar, B., Karypis, G., Konstan, J., Riedl, J. Analysis of recommendation algorithms for e-commerce. *Proceedings of the 2nd ACM conference on Electronic commerce (EC'00)*, ACM, New York, NY, USA, pp. 158-167.

[99] Breese, J. S., Heckerman, D., Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 43-52.

[100] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J. GroupLens: applying collaborative filtering to usenet news. *Communications of the ACM*, 1997, 40(3), 77-87.

[101] Liu, J., Dolan, P., Pedersen, E.R. Personalized news recommendation based on click behavior. *Proceedings of the 15th international conference on Intelligent user interfaces (IUI '10)*, ACM, New York, NY, USA, 2010, pp. 31-40.

[102] Bollen, D., Knijnenburg, B. P., Willemsen, M. C., Graus, M. Understanding choice overload in recommender systems. *Proceedings of the fourth ACM conference on Recommender systems (RecSys '10)*, ACM, New York, NY, USA, 2010, pp. 63-70.

[103] Eirinaki, M., Lampos, C., Paulakis, S., Vazirgiannis, M. Web Personalization Integrating Content semantics and Navigational Patterns. *Proceedings of the 6th Annual ACM International Workshop on Web*

Information and Data Management WIDM '04, ACM, Washington DC, USA, 2004, pp. 72-79.

[104] Baraglia, R., Silvestri, F. An Online Recommender System for Large Web Sites. *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence WI '04*, IEEE Computer Society, 2004, pp. 199-205.

[105] Vassiliou, C., Stamoulis, D., Spiliotopoulos, A., Martakos, D. Creating adaptive web sites using personalization techniques: a unified, integrated approach and the role of evaluation. *Adaptive evolutionary information systems*, IGI Publishing, Hershey, PA, USA, 2003, pp. 261-285.

[106] Heras Ballell, T.R., Legal Aspects of Recommender Systems in the Web 2.0: Trust, Liability and Social Networking. *Recommender Systems for the Social Web*. Intelligent Systems Reference Library, Springer, vol. 32, part 1, 2012, pp. 43-62.

[107] Chen, S., Williams, M.-A. Towards a comprehensive requirements architecture for privacy-aware social recommender systems. *Proceedings of the 7th Asia-Pacific Conference on Conceptual Modelling (APCCM '10)*, S. Link, A. Ghose (Eds.), vol. 110, Australian Computer Society, Darlinghurst, Australia, 2010, pp. 33-42.

[108] Hofgesang, P.I. Relevance of Time Spent on Web Pages. Proceedings of the Knowledge Discovery on the Web (KDD) ACM SIGKDD Workshop on Web Mining and Web Usage Analysis at WebKDD 2006, ACM Press, 2006.

[109] Farzan, R., Brusilovsky, P. Social navigation support in e-learning: What are real footprints? *Proceedings of IJCAI'05 Workshop on Intelligent Techniques for Web Personalization*, Edinburgh, Scotland, 2005, pp 49-56.

[110] Pan, B., Hembrooke, H.A., Gay, G.K., Granka, L.A., Feusner, M.K., Newman, J.K. The determinants of web page viewing behavior: an eye-tracking study. *Proceedings of the 2004 symposium on Eye tracking research & applications (ETRA '04)*, ACM, New York, NY, USA, 2004, pp. 147-154.

[111] Weinreich, H., Obendorf, H., Herder, E., Mayer, M. Off the Beaten Tracks: Exploring Three Aspects of Web Navigation. *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, ACM Press, Edinburgh, Scotland, 2006, pp. 133-142.

[112] Spiliopoulou, M., Pohle, C. Data Mining for Measuring and Improving the Success of Web Sites. *Data Mining and Knowledge Discovery*, 5(1-2), 2001, 85-114.

[113] Spiliopoulou, M. Web usage mining for Web site evaluation. *Communications of the ACM*, 43(8), 2000, 127-134.

[114] Olsina, L., Rossi, G. Measuring Web Application Quality with WebQEM. *IEEE MultiMedia*, 2002, 9(4), 20-29.

[115] Hartmann, J. Sure, Y. An Infrastructure for Scalable, Reliable Semantic Portals. *IEEE Intelligent Systems*, 2004, 19(3), 58-65.

[116] Jin, Y.,Decker, S., Wiederhold G. OntoWebber: Model-Driven Ontology-Based Web Site Management. *1st Int. Semantic Web Working Symposium (SWWS 01)*, 2001, pp. 529-547.

[117] Mikroyannidis, A., Theodoulidis, B. Web usage Driven Adaptation of the Semantic Web. *Proceedings of UserSWeb: Workshop on End User Aspects of the Semantic Web*, Heraklion, Crete, 2005, pp. 137-147.

[118] Coenen, F., Swinnen, G., Vanhoof, K., Wets, G. A Framework for Self Adaptive Websites: Tactical Versus Strategic Changes. *Proceedings of Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data mining, WEBKDD'2000 Web Mining for E-Commerce – Challenges and Opportunities*, 2000, pp. 1-6.

[119] Cooley, R., Mobasher, B., Srivastava, J. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems*, 1(1), 1999, 5-32.

[120] Robal, T., Kalja, A. e-EDU – An Information System for e-learning Services. *Databases and Information Systems.* Frontiers in Artificial Intelligence and Applications, vol. 118, IOS Press, Amsterdam, The Netherlands, 2005, pp. 288-298.

[121] Tan, P.-N. , Kumar, V. Discovery of Web Robot Sessions based on their Navigational Patterns. *Data Mining and Knowledge Discovery*, 2002, 6(1), 9-35.

[122] Robal, T., Kalja, A., Põld, J. Analysing the Web Log to Determine the Efficiency of Web Systems. *Proceedings of the 7th International Baltic Conference on Databases and Information Systems DB&IS'2006*, Communications, Vilnius, Lithuania, 2006, pp. 264-275.

[123] Nah, F. A study on tolerable waiting time: how long are Web users willing to wait?. *Behaviour & Information Technology*, Taylor & Francis, 2004, 23(3), 153-163.

[124] Miller, G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 1956, 63, 81-97.

[125] Peterson, L.R., & Peterson, M.J. Short-term retention of individual verbal items. *Journal of Experimental Psychology*, 1959, 58, 193-198.

[126] Rafter, R., Smyth, B. Passive Profiling from Server Logs in an Online Recruitment Environment. *Proceedings of the IJCAI Workshop on*

Intelligent Techniques for Web Personalization (ITWP 2001), Seattle, WA, USA, 2001, pp. 35-41.

[127] Ziefle, M. Effects of display resolution on visual performance. *Human Factors*, 1998, 40(4), 555-568.

[128] Robal, T., Kalja, A. Learning from Users for a Better and Personalized Web Experience. Proceedings of the *PICMET '12 Conference "Technology Management for Emerging Technologies"*, July 29 - August 2, 2012, Vancouver, Canada, PICMET: USA, 10p. [to be published]

[129] Robal, T., Kalja, A. Web systems evaluation on users' behaviour modelling. *Databases and Information Systems V : Selected Papers from the Eighth International Baltic Conference, DB&IS 2008*, Frontiers in Artificial Intelligence and Applications, vol. 187, IOS Press, Amsterdam, 2009, pp. 41-52.

[130] Robal, T., Haav, H-M., Kalja, A. Making web users' domain models explicit by applying ontologies. *Advances in Conceptual Modeling - Foundations and Applications: ER 2007 Workshops CMLSA, FP-UML, ONISW, QoIS, RIGiM, SeCoGIS*, LNCS, vol. 4802, Springer Berlin / Heidelberg, 2007, pp. 170-179.

[131] Robal, T., Kalja, A. A model for users' action prediction based on locality profiles. *The Inter-Networked World: ISD Theory, Practice, and Education*, M. Lang, W. Wojtkowski, G. Wojtkowski, S. Wrycza, J. Zupancic (eds.), Springer-Verlag, 2008, pp. 169-182.

[132] Robal, T., Kalja, A., Conceptual web users' actions prediction for ontology-based browsing recommendations. *Information Systems Development: Towards a Service Provision Society*, G.A. Papadopoulos, G. Wojtkowski, W. Wojtkowski, S. Wrycza, J. Zupancic (eds.), Springer-Verlag New York, 2009, pp. 121-129.

[133] Davison, B. Predicting web actions from HTML content. *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia (HYPERTEXT '02)*, James Blustein (Ed.), ACM, New York, NY, USA, 2002, pp. 159-168.

[134] Robal, T., Kalja, A. Applying User Domain Model to Improve Web Recommendations. *Frontiers in Artificial Intelligence and Applications. IOS Press*, Amsterdam, 14p. [to be published]

[135] IEEE Standard for Learning Object Metadata, IEEE Standard 1484.12.1-2002, 2002, pp. i-32.

[136] Robal, T., Kalja, A., Moving studies to e-environments: a case study. *Current Developments in Technology-Assisted Education*, 2006, vol. II, pp. 936-940.

[137] Robal, T., Kalja, A. Making use of personalized web services in the study process. *Proceedings of the 11th International Biennial Baltic Electronics Conference BEC 2008*, Tallinn, Estonia, 2008, pp. 211-212.

[138] Robal, T., Kalja, A. Enabling students contemporary ways of learning using e-supported courses. *Proceedings of 19th European Association for Education in Electrical and Information Engineering (EAEEIE) Annual Conference*, Tallinn, Estonia, IEEE, 2008, pp. 14-19.

[139] Robal, T., Kalja, A. Applying e-Environments in Teaching the Basics of Digital Logic. *Proceedings of the International Conference on Microelectronic Systems Education MSE'07*, IEEE Computer Society, USA, San Diego, 2007, pp. 41-42.

[140] Robal, T., Kalja, A. Interactive Hands-On Tools as Learning Objects on Web Services. *Proceedings of the International Conference on Microelectronic Systems Education MSE'09*, IEEE Press, pp. 73-76.

[141] Robal, T., Kalja, A. Creating interactive learning objects with web services. *Proceedings of the 20th EAEEIE Annual Conference*, Valencia, Spain, IEEE Publishing, 2009, pp. 1-6.

[142] Papazoglou, M.P. *Web Services: Principles and Technology*. Pearson - Prentice Hall, 2007, 752p.

[143] Robal, T., Kann, T., Kalja, A. An ontology-based intelligent learning object for teaching the basics of digital logic. *2011 IEEE International Conference on Microelectronic Systems Education (MSE)*, San Diego, CA, USA, IEEE Computer Society, 2011, pp. 106-107.

Appendix A
Log Analyzer System

This appendix outlines the database model and lists the routines implemented for the Log Analyzer System.

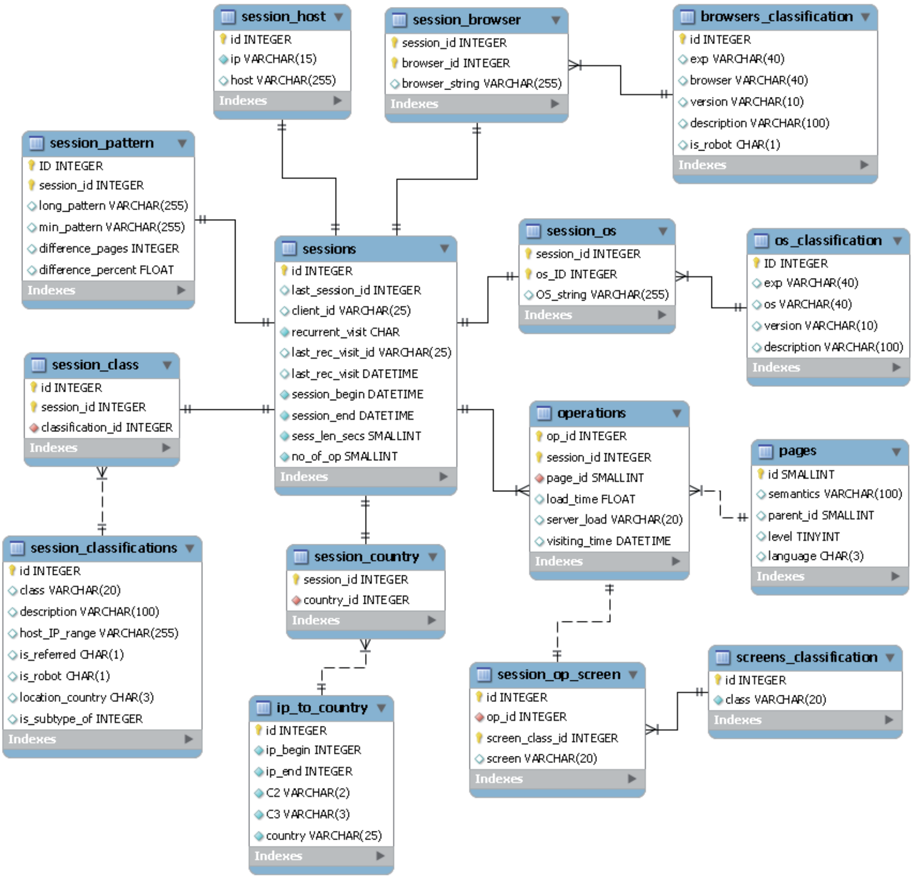


Figure A-1. *Database model for the Log Analyzer System*

Table A-1. *Data processing routines in the Log Analyzer System.*

Routine name	**Description**
`clean_log()`	Procedure: Cleans the operational log table.
`clean_tables()`	Procedure: Cleans tables holding parsed log data (except classifiers).
`debug(_msg TEXT)`	Procedure: inserts debug message into log parser log, if debug mode enabled

error(_msg TEXT)	Procedure: inserts error message into log parser log, if error logging enabled
get_browser_class_id (user_agent VARCHAR(255)) RETURNS int(11)	Function: detects user's browser from user-agent string, returns classifier ID from table browsers_classification.
get_country_id (ip VARCHAR(15)) RETURNS int(11)	Function: detects visitor's country based on the request's IP address; if none found from the IP range lookup table, ID for unknown country will be returned.
get_os_class_id (user_agent VARCHAR(255)) RETURNS int(11)	Function: detects visitor's operating system from user-agent string, returns classifier ID from table os_classification.
get_screen_class_id (screen VARCHAR(20)) RETURNS int(11)	Function: gets user's screen size, returns classifier ID from table screen_classification.
get_session_id (client_id VARCHAR(25)) RETURNS int(11)	Function: returns session ID for specified client_id. If none found, returns 0 for new session.
get_session_last_page_id (session_id INT) RETURNS smallint(6)	Function: returns the last visited page ID for a specified session. If none, returns 0.
Info (_msg TEXT)	Procedure: inserts informative message into log parser log, if info logging enabled
insert_operation (session_id INT, page_id SMALLINT, query_method VARCHAR(10), load_time FLOAT, serverload_1_5_10 VARCHAR(100), visiting_time DATETIME) RETURNS int(11)	Function: Inserts new operation for session, returns inserted operation ID.
insert_session (client_id VARCHAR(25), recurrent_visit CHAR(1), last_rec_visit DATETIME, last_rec_visit_id VARCHAR(25), session_begin DATETIME) RETURNS int(11)	Function: Inserts new session, returns inserted session ID.
load_settings ()	Procedure: loads analyzer settings for parsing into MySQL user space variables.
parse_log_entries()	Procedure: parses log rows from table log_visits, creates a view log_entries_view,

	and proceeds to parse log data into log analyzer database.
set_browser (session_id INT, user_brauser VARCHAR(200))	Procedure: sets browser classifier ID for session into table session_browser.
set_country (session_id INT, ip VARCHAR(15))	Procedure: sets country classifier ID for session into table session_country.
set_host (session_id INT, ip VARCHAR(15), host VARCHAR(255))	Procedure: sets host classifier ID for session into table session_host.
set_operation set_operation (session_id INT, page_id SMALLINT, query_method VARCHAR(10), load_time FLOAT, serverload_1_5_10 VARCHAR(100), visiting_time DATETIME, screen VARCHAR(20))	Procedure: calls insert_operation to set session operation data into table operations and set_operation_screen to insert screen classifier ID into table session_op_screen.
set_operation_screen (operation_id INT, screen VARCHAR(20))	Procedure: sets screen size classifier ID for operation into table session_op_screen.
set_os (session_id INT, user_brauser VARCHAR(200))	Procedure: sets users' operating system classifier ID for session into table session_os based on user_agent string using function get_os_class_id.
set_session_class (session_id INT, ip VARCHAR(15), site_referer VARCHAR(255))	set_session_class Procedure: sets session classification into table session_class according to classifiers listed in session_classifications.
update_last_processed_id (last_processed_id INT)	Procedure: updates last processed log id in table last_processed.
update_session_parameters (session_id INT, last_visiting_time DATETIME)	Procedure: sets session end time, length and number of operations in session.
update_session_pattern (session_id INT, page_id SMALLINT)	Procedure: updates session operations pattern.
warn (msg TEXT)	Procedure: inserts warning message into log parser log, if warning logging enabled

Appendix B
TSP IMPACT STUDY

The impact of applying time spent on page as an interest indicator to the ranking of popular pages. Threshold condition t_0 represents results with no TSP applied. Data from the DCE web usage log analysis.

N	Threshold condition applied for TSP (Equation 4-3)									
	t_0		t_1		t_2		t_{2a}		t_{2b}	
	Page ID	Trend	Page ID	Trend	Page ID	Trend	Page ID	Trend	Page ID	Trend
1	400		400	$\updownarrow$	400	$\updownarrow$	4110	$\uparrow^{+1}$	200	$\uparrow^{+3}$
2	4110		200	$\uparrow^{+2}$	4110	$\updownarrow$	400	$\downarrow_{-1}$	400	$\downarrow_{-1}$
3	1505		470	$\uparrow^{+3}$	410	$\uparrow^{+5}$	1505	$\updownarrow$	4110	$\downarrow_{-1}$
4	200	$\rightarrow t_3$	300	$\uparrow^{+3}$	1505	$\downarrow_{-1}$	477	$\uparrow^{+6}$	1505	$\downarrow_{-1}$
5	480		4110	$\downarrow_{-3}$	470	$\uparrow^{+1}$	410	$\uparrow^{+3}$	480	$\updownarrow$
6	470		1505	$\downarrow_{-3}$	477	$\uparrow^{+4}$	171	$\uparrow^{+6}$	171	$\uparrow^{+6}$
7	300	$\rightarrow t_4$	150	$\uparrow^{+2}$	150	$\uparrow^{+2}$	480	$\downarrow_{-2}$	470	$\downarrow_{-1}$
8	410	$\rightarrow t_4$	480	$\downarrow_{-3}$	300	$\downarrow_{-1}$	472	$\uparrow^{+6}$	476	$\uparrow^{+9}$
9	150		410	$\downarrow_{-1}$	200	$\downarrow_{-5}$	4711	$\uparrow^{+7}$	500	$\uparrow^{+6}$
10	477	$\rightarrow t_4$	210	$\uparrow^{+1}$	480	$\downarrow_{-5}$	476	$\uparrow^{+7}$	472	$\uparrow^{+4}$
11	210	$\rightarrow t_3$ $\rightarrow t_4$	477	$\downarrow_{-1}$	133	$\uparrow^{+2}$	473	$\uparrow^{+8}$	4711	$\uparrow^{+5}$
12	171		220	$\uparrow^{+6}$	171	$\updownarrow$	150	$\downarrow_{-3}$	4712	$\leftarrow t_0$
13	133	$\rightarrow t_4$	472	$\uparrow^{+1}$	210	$\downarrow_{-2}$	133	$\updownarrow$	474	$\leftarrow t_0$
14	472		500	$\uparrow^{+1}$	476	$\uparrow^{+3}$	470	$\downarrow_{-8}$	460	$\leftarrow t_0$
15	500		230	$\leftarrow t_0$	473	$\uparrow^{+4}$	300	$\downarrow_{-8}$	600	$\uparrow^{+5}$
16	4711	$\rightarrow t_1$	171	$\downarrow_{-4}$	500	$\downarrow_{-1}$	4712	$\leftarrow t_0$	471	$\leftarrow t_0$
17	476	$\rightarrow t_1$	471	$\leftarrow t_0$	472	$\downarrow_{-3}$	474	$\leftarrow t_0$	473	$\uparrow^{+2}$
18	220	$\rightarrow t_2$ $\rightarrow t_3$ $\rightarrow t_4$	133	$\downarrow_{-5}$	4711	$\downarrow_{-2}$	460	$\leftarrow t_0$	230	$\leftarrow t_0$
19	473		473	$\updownarrow$	600	$\uparrow^{+1}$	500	$\downarrow_{-4}$	150	$\downarrow_{-10}$
20	600	$\rightarrow t_1$	474	$\leftarrow t_0$	310	$\leftarrow\ t_0$	600	$\updownarrow$	6305	$\leftarrow t_0$

Trend legend:

$\updownarrow$	No change in position in the top hit list
$\uparrow^{+x}$	Increase in top hit by X positions
$\downarrow_{-x}$	Decrease in top hit by X positions
$\rightarrow t_x$	Page moved out from top N list by TSP condition t_x
$\leftarrow t_x$	Page moved into top N list by TSP condition t_x in comparison to t_0

Appendix C
EXAMPLE WEB ONTOLOGY

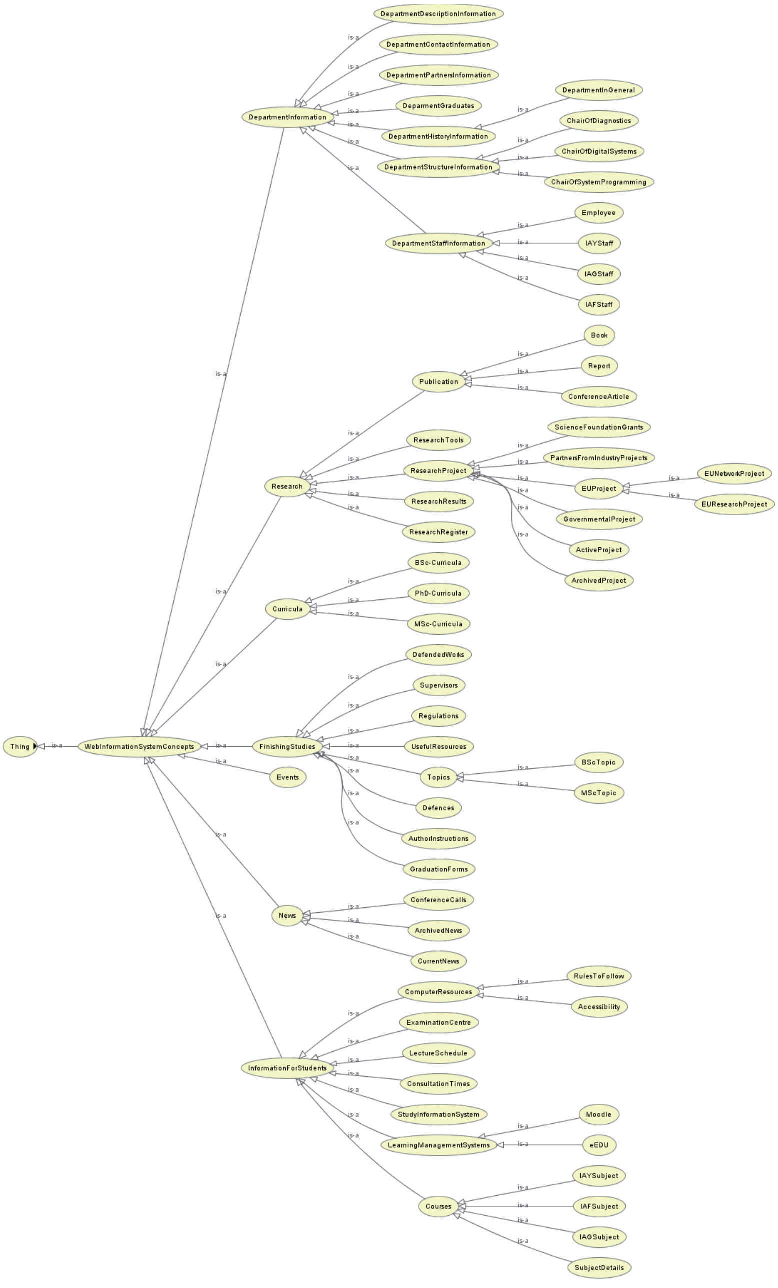


Figure C-1. *An overview of the web ontology created for the DCE website. Ontology visualization by the OWLViz plugin in the Protégé ontology editor.*

Appendix D
OWL2 Description Example

```
<owl:Class rdf:about="http://www.owl-ontologies.com/Ontology1171915362.owl#P_410_430_450">
   <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#ExtractedProfiles"/>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#hasItem"/>
         <owl:someValuesFrom rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#Courses"/>
      </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#hasItem"/>
            <owl:someValuesFrom rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#ComputerResources"/>
      </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#hasItem"/>
         <owl:someValuesFrom rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#LectureSchedule"/>
      </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#hasItem"/>
         <owl:allValuesFrom>
             <owl:Class>
                 <owl:unionOf rdf:parseType="Collection">
                     <rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1171915362.owl#ComputerResources"/>
                     <rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1171915362.owl#Courses"/>
                     <rdf:Description rdf:about="http://www.owl-ontologies.com/Ontology1171915362.owl#LectureSchedule"/>
                 </owl:unionOf>
             </owl:Class>
         </owl:allValuesFrom>
      </owl:Restriction>
   </rdfs:subClassOf>

   <owl:disjointWith rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#P_430_410_4110"/>
   <owl:disjointWith rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#P_472_473_474"/>

   <!--     ...  and so on for each disjoint class -->

   <owl:disjointWith rdf:resource="http://www.owl-ontologies.com/Ontology1171915362.owl#P_474_476_477"/>
   <rdfs:comment rdf:datatype="&xsd;string">410-430-450</rdfs:comment>
</owl:Class>
```

Figure D-1*. A fragment of generated OWL 2 description in RDF/XML format of the extracted user profile class ‘P_410_430_450’.*

Appendix E
GENERAL PREDICTION ALGORITHM

```
Function getItemsOfInterest(localityProfile, limit, minW, maxW)
  /*
  localityProfile: array            = online user’s current locality
  limit: int                        = max size of predicted itemset
  minW: int                         = min locality size w to apply
  maxW: int                         = max locality size w to apply
  interestItems: array(array)       = returned set of items of interest
  getItemsFromRepository(): array   = method to query DB
  sliceProfile():array              = method to get only the last subpart of
                                      locality profile, defined by offset
  runningLocalityProfile: array     = sliced locality profile
  detectedItemsOfInterest: array    = final resultset
  */

    BEGIN
        if maxW > count(localityProfile)
            maxW = count(localityProfile)
        end if
        if (minW >= 2) and (minW <= maxW)
            w = minW
        else
            w = 2
        end if

        do
            if count(localityProfile) >= w
                offset = count(localityProfile) - w
            else
                offset = 0
            end if
            runningLocalityProfile = sliceProfile(localityProfile, offset)
            interestItems[w] = getItemsFromRepository(runningLocalityProfile, limit)
            increment w
        while w < maxW
        decrement w

        if minW == maxW
            detectedItemsOfInterest = interestItems[w]
        else
            i = 2
            while i <= w do
                j = 0
                while j < count(interestItems[i]) do
                    add interestItems[i][j] to detectedItemsOfInterest
                    increment j
                end while
                increment i
            end while
            for each key=>row in detectedItemsOfInterest
                      add item to list1
                      add confidence to list2
            next
            multisort (list2 desc, list1, detectedItemsOfInterest)
            detectedItemsOfInterest = extract(detectedItemsOfInterest, limit)
        end if
        return detectedItemsOfInterest
    END
```

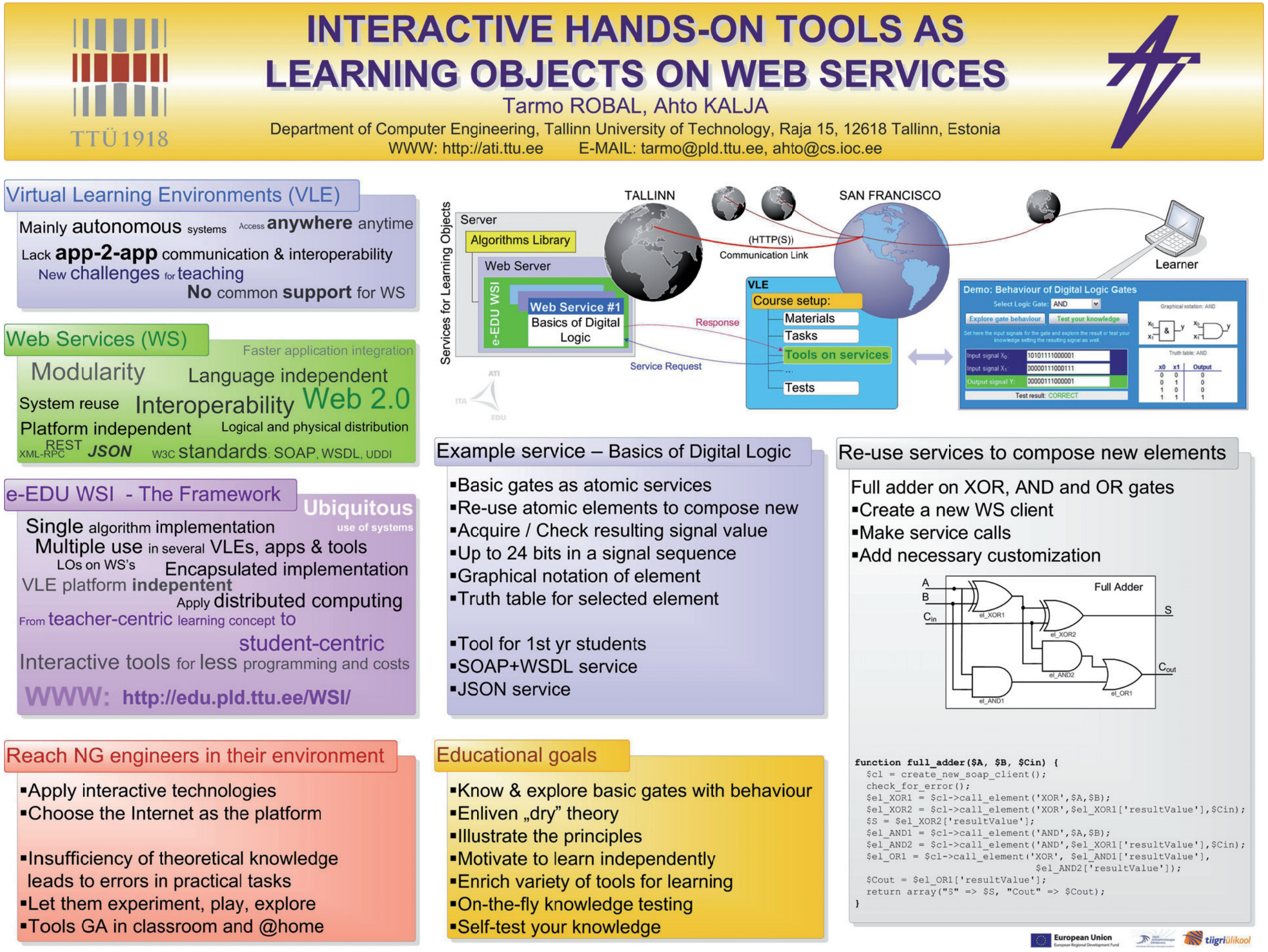
Figure E-1. *General algorithm for providing predictions based on algorithms A1-A6.*

Appendix F
RECOMMENDATION ALGORITHM

```
Function getRecommendedItems(userLocality, algorithm, N, M, u)
 /*
   userLocality: array                   = latest operations in user session
   algorithm: int                        = identificator for prediction algorithm in PE
   N: int                                = number of items to predict by PE
   M: int                                = number of items to recommend
   u: double                             = significance factor
   userICProfile: array                  = set of user interest concepts (profile)
   predictedRIC: array                   = set of predicted interest concepts
   userProfile: string                   = predefined user profile (described in ontology)
   recommendationResult: array           = set of recommended items
   mapToConcepts(array): array           = method to perform mapping to ontology concepts
   detectUserProfile(array): string      = method to get current user profile
   isConceptInProfile(string, string)    = method to check if a concept is present on
                                           specified user profile
   reEvalOrder(array): array             = method to re-evaluate item order (DESC)
   mapToPages: array                     = method to map concepts to WIS pages
 */

    BEGIN
        if M > N then M = N
        userICProfile = mapToConcepts(userLocality)
        predictedRIC = getPredictedConcepts(userICProfile, algorithm, N)
        userProfile = detectUserProfile(userICProfile)

        for each element in predictedRIC
            if (isConceptInProfile(element['concept'], userProfile)) = True
                element['rank'] = element['rank'] * u
            end if
        next

        predictedRIC = reEvalOrder(predictedRIC)
        recommendationResult = mapToPages(predictedRIC)
        recommendationResult = array_slice(recommendationResult, M)

        return recommendationResult
    END
```

Figure F-1. *Algorithm used for recommendation establishment based on user action prediction and user profiling.*

Appendix G
Learning Objects on Web Services



INTERACTIVE HANDS-ON TOOLS AS LEARNING OBJECTS ON WEB SERVICES

Tarmo ROBAL, Ahto KALJA

Department of Computer Engineering, Tallinn University of Technology, Raja 15, 12618 Tallinn, Estonia
WWW: http://ati.ttu.ee E-MAIL: tarmo@pld.ttu.ee, ahto@cs.ioc.ee

Virtual Learning Environments (VLE)

Mainly autonomous systems Access anywhere anytime

Lack app-2-app communication & interoperability

New challenges for teaching

No common support for WS

Web Services (WS)

Faster application integration

Modularity Language independent

System reuse Interoperability Web 2.0

Platform independent Logical and physical distribution

XML-RPC REST JSON W3C standards: SOAP, WSDL, UDDI

e-EDU WSI - The Framework

Ubiquitous use of systems

Single algorithm implementation

Multiple use in several VLEs, apps & tools

LOs on WS's Encapsulated implementation

VLE platform indepentent

Apply distributed computing

From teacher-centric learning concept to student-centric

Interactive tools for less programming and costs

WWW: http://edu.pld.ttu.ee/WSI/

Reach NG engineers in their environment

- Apply interactive technologies
- Choose the Internet as the platform

- Insufficiency of theoretical knowledge leads to errors in practical tasks
- Let them experiment, play, explore
- Tools GA in classroom and @home

Demo: Behaviour of Digital Logic Gates

Select Logic Gate: AND

Set here the input signals for the gate and explore the result or test your knowledge setting the resulting signal as well.

Input signal X_0: 10101111000001

Input signal X_1: 00000111000111

Output signal Y: 00000111000001

Test result: CORRECT

Graphical notation: AND

Truth table: AND

x0	x1	Output
0	0	0
0	1	0
1	0	0
1	1	1

Example service – Basics of Digital Logic

- Basic gates as atomic services
- Re-use atomic elements to compose new
- Acquire / Check resulting signal value
- Up to 24 bits in a signal sequence
- Graphical notation of element
- Truth table for selected element

- Tool for 1st yr students
- SOAP+WSDL service
- JSON service

Educational goals

- Know & explore basic gates with behaviour
- Enliven „dry" theory
- Illustrate the principles
- Motivate to learn independently
- Enrich variety of tools for learning
- On-the-fly knowledge testing
- Self-test your knowledge

Re-use services to compose new elements

Full adder on XOR, AND and OR gates

- Create a new WS client
- Make service calls
- Add necessary customization

```
function full_adder($A, $B, $Cin) {
  $cl = create_new_soap_client();
  check_for_error();
  $el_XOR1 = $cl->call_element('XOR',$A,$B);
  $el_XOR2 = $cl->call_element('XOR',$el_XOR1['resultValue'],$Cin);
  $S = $el_XOR2['resultValue'];
  $el_AND1 = $cl->call_element('AND',$A,$B);
  $el_AND2 = $cl->call_element('AND',$el_XOR1['resultValue'],$Cin);
  $el_OR1 = $cl->call_element('XOR', $el_AND1['resultValue'],
                                     $el_AND2['resultValue']);
  $Cout = $el_OR1['resultValue'];
  return array("S" => $S, "Cout" => $Cout);
}
```

European Union
European Regional Development Fund
tiigriülikool

Figure G-1. *Framework for developing digital learning objects based on web services [140].*

CURRICULUM VITAE

Personal data

Name	Tarmo Robal
Date of birth	14.06.1979
Place of birth	Estonia
Citizenship	Estonian

Contact information

Address	Raja 15, 12618 Tallinn, ESTONIA
Phone	+372 620 2263
E-mail	tarmo.robal@ati.ttu.ee

Education

2003 – ...	Ph.D. student in Information and Communication Technology, Tallinn University of Technology (TUT)
2001 – 2003	M.Sc. in Computer Engineering, TUT
1997 – 2001	B.Sc. in Computer Engineering, TUT
1986 – 1997	Secondary Education from Tallinn Lilleküla High School

Career

2004 – ...	Tallinn University of Technology, Faculty of Information Technology, Department of Computer Engineering, *Research Scientist*
2003 – 2007	The Estonian Information Technology College, *Lecturer*
2003 – 2004	Tallinn University of Technology, Faculty of Information Technology, Dept. of Computer Engineering, *Senior Engineer*
2001 – 2002	Sampo Pank A/S, *Software designer*
1998 – 2002	Tallinn University of Technology, Faculty of Information Technology, Dept. of Computer Engineering, *Engineer*

Scientific Work

PC Member for the Following International Events

- Int'l. Conf. on Advances in Semantic Processing SEMAPRO, since 2009
- Annual ACM Symposium on Applied Computing, The Semantic Web and Applications SWA, since 2008
- Int. Conference on Information Systems Development ISD, 2008, 2009
- The Joint International Workshop on Metamodels, Ontologies, Semantic Technologies, and Information Systems for the Semantic Web MOST-ONISW, 2009
- The 2nd International workshop on Ontologies and Information Systems for the Semantic Web ONISW, 2008

Reviewer

- 10th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2012)
- 12th Biennial Baltic Electronics Conference (BEC2010)
- East-European Conference on Advances in Databases and Information Systems (ADBIS 2010, ADBIS 2012)

Publications

Robal, T., Kalja, A., Learning from Users for a Better and Personalized Web Experience, PICMET '12 Conference "Technology Management for Emerging Technologies", Jul 29 - Aug 2, 2012, Vancouver, Canada, 10p. [to be published]

Robal, T., Kalja, A. Applying User Domain Model to Improve Web Recommendations. Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 14p. [to be published]

Kalja, A., Põld, J., Robal, T., Vallner, U. Modernization of the e-government in Estonia. PICMET '11 : Proceedings Technology Management in the Energy-Smart World. D. Kocaoglu, T. Anderson, T. U. Daim (Eds). Portland, OR, USA. PICMET, 2011, pp. 3151 - 3157.

Robal, T., Kann, T., Kalja, A. An ontology-based intelligent learning object for teaching the basics of digital logic. Proceedings of 2011 IEEE International Conference on Microelectronic Systems Education (MSE). IEEE Computer Society, 2011, pp. 106-107.

Kalja, A., Robal, T., Vallner, U. Towards information society: Estonian case study. Proceedings of PICMET '09: Technology Management in the Age of Fundamental Change. D. Kocaoglu, T. Anderson, T. U. Daim (Eds). Portland, Oregon USA. PICMET, 2009, pp. 3218-3225.

Robal, T., Kalja, A. Web systems evaluation on users' behaviour modelling. Databases and Information Systems V: Selected Papers from the 8th Intl. Baltic Conference, DB&IS 2008. H-M. Haav,A. Kalja (Eds). Amsterdam, IOS Press, Frontiers in Artificial Intelligence and Applications, 187, 2009, pp. 41-52.

Robal, T., Kalja, A. Conceptual web users' actions prediction for ontology-based browsing recommendations. Information Systems Development: Towards a Service Provision Society. New York: Springer-Verlag, 2009, pp. 121-129.

Robal, T., Kalja, A. A model for users' action prediction based on locality profiles. Information Systems Development: Challenges in Practice, Theory and Education. C. Barry, K. Conboy, M. Lang, G. Wojtkowski, W. Wojtkowski, Wita (Eds). New York: Springer, Vol. 1, 2009, pp. 169-182.

Robal, T., Kalja, A. Creating interactive learning objects with web services. Proceedings of the 20th EAEEIE Annual Conference. IEEE, 2009, pp. 1-6.

Robal, T., Kalja, A. Interactive Hands-On Tools as Learning Objects on Web Services. Proceedings of International Conference on Microelectronic Systems Education MSE'09. San Francisco, CA, USA. IEEE, 2009. pp. 73-76.

Robal, T., Kalja, A. Evaluations for improving web systems on the basis of users behaviour modelling. Databases and Information Systems: Proceedings of the 8th Intl Baltic Conference, Baltic DB&IS 2008. H-M. Haav, A. Kalja (Eds). Tallinn: Tallinn University of Technology Press, 2008, pp. 229-240.

Põld, J., Kalja, A., Robal, T. A five step refactoring process: improving software design properly. Databases and Information Systems: Proceedings of the 8th Intl Baltic Conference, Baltic DB&IS 2008. H-M. Haav, A. Kalja (Eds). Tallinn: Tallinn University of Technology Press, 2008, pp. 37-48.

Robal, T., Kalja, A. Enabling students contemporary ways of learning using e-supported courses. 19th EAEEIE Annual Conference, Formal Proceedings. The 19th European Association for Education in Electrical and Information Engineering Annual Conference. Tallinn, Estonia, IEEE, 2008, pp. 14-19.

Robal, T., Kalja, A. Making use of personalized web services in the study process. Proceedings: 11th Biennial Baltic Electronics Conference BEC 2008. Tallinn, Estonia, 2008, pp. 211-212.

Robal, T., Kalja, A. Evaluating Web systems on the basis of users' behavior modeling. IKTDK Annual Conference, Estonia, TUT Press, 2008, pp. 64-67.

Põld, J., Kalja, A., Robal, T. A five step refactoring process: improving software design properly. IKTDK Annual Conference, Estonia, TUT Press, 2008, pp. 93-96.

Kalja, A., Kindel, K., Kivi, R., Robal, T. eGovernment Services: How to Develop Them, How to Manage Them? Proceedings of PICMET'07: Management of Converging Technologies: PICMET '07 Portland International Center for Management of Engineering and Technology. Portland OR, USA, IEEE, 2007, pp. 2795-2798.

Robal, T., Kalja, A. Applying user profile ontology for mining web site adaptation recommendations. Local Proceedings of the 11th East-European Conference on Advances in Databases and Information Systems (ADBIS 2007).

Y. Ioannidis, B. Novikov, B. Rachev (Eds). Varna, Bulgaria: Technical University of Varna, 2007, pp. 126-135.

Robal, T., Haav, H-M., Kalja, A. Making web users' domain models explicit by applying ontologies. Advances in Conceptual Modeling - Foundations and Applications : ER 2007 Workshops CMLSA, FP-UML,ONISW, QoIS, RIGiM, SeCoGIS. J.-L. Hainaut (Ed). Auckland, New Zealand, Berlin: Springer, 2007, Lecture Notes in Computer Science, vol. 4802, pp. 170-179.

Robal, T., Kalja, A. Applying e-environments in teaching the basics of digital logics. Proceedings of the IEEE International Conference on Microelectronic Systems Education MSE 2007. P. Kellenberger (Ed), San Diego, CA, USA. IEEE Computer Society Press, 2007, pp. 41-42.

Robal, T., Kalja, A. Applying ICT for Improving the Study Process. Proceedings of 18th EAEEIE Annual Conference, Innovation in Education for Electrical and Information Engineering. Prague, Czech Technical University in Prague, 2007, pp. 1-5.

Robal, T. Applying User Profile Ontology for Mining Web Site Adaptation Recommendations. IKTDK Annual Conference, Estonia, 2007, pp. 125-128.

Robal, T., Kalja, A., Põld, J. Analysing the Web Log to Determine the Efficiency of Web Systems. Seventh International Baltic Conference on Databases and Information Systems. O. Vasilecas, J. Eder, A. Caplinskas (Eds). Vilnius, Lithuania, Technika, 2006, pp. 264-275.

Robal, T., Kalja, A. Moving studies to e-environments: a case study. Current Developments in Technology-Assisted Education. A. Méndez-Vilas, M. Solano Martín, J.A. Mesa González, J. Mesa González. Badajoz, Spain: Formatex, 2006, pp. 936-940.

Robal, T., Brik, M., Aarna, M. e-Learning of Digital Logic. Using e-Environments in Teaching Digital Logic. Proc: 6th International Workshop on Microelectronics Education EWME 2006. Stockholm, Sweden, Royal Institute of Technology KTH, 2006, pp. 120-123.

Robal, T. Analysing the Web Log to Determine the Efficiency of Web Systems. IKTDK Annual Conference, 2006, pp. 64-67.

Robal, T., Kalja, A. e-EDU - an information system for e-learning services. Selected Papers from Sixth International Baltic Conference DB&IS'2004. J. Barzdins, A. Caplinskas (Eds). Amsterdam: IOS Press, Frontiers in artificial intelligence and applications, vol. 118, 2005, pp. 288-298.

Kruus, H., Orasson, E., Robal, T., Ubar, R. Investigating Defects in Digital Circuits by Boolean Differential Equations. The 4th Int. Conference "Distance Learning - Educational Sphere of XXI Century" DLESC'04. Minsk, 2004, pp. 432-435.

Robal, T., Kalja, A. e-EDU - an information system for e-learning services. Acta Universitatis Latviensis, vol. 672, 2004, pp. 469-480.

Robal, T., Kruus, H. e-Bibliothecula - A Virtual Library Service. 13th Intl. Conference on Information Systems Development. Advances in Theory, Practice and Education, Lithuania, 2004, pp. 139-148.

Robal, T., Viies, V., Kruus, M. The Rational Unified Process with the "4+1" View Model of Software Architecture - a Way for Modeling Web Applications. Proc. of the Fifth Int. Baltic Conference: BalticDB&IS 2002, Tallinn, Estonia, 2002, pp. 119-132.

Defended Theses

2003 Master of Science in Computer Engineering, *Using the RUP with 4+1 Views of Software Architecture for Web Applications Development*. TUT, supervisor: assoc. professor V. Viies

2001 Bachelor of Science in Computer Engineering, *Dynamic web application for Department of Computer Engineering at Tallinn University of Technology*. TUT, supervisor: assoc. professor M. Kruus

Main Areas of Scientific Work

Web information systems, web mining, web personalization, adaptive web, learning management systems, digital learning objects.

Supervised Theses

Janari Põld, *Improving software development and maintainability: A refactoring process model for use in evolving systems*, 2007, M.Sc. in Computer Engineering, *Cum Laude*.

Awards

The Jaan Poska joint scholarship of the City of Tallinn and Tallinn University of Technology, 2007

AS Eesti Energia scholarship, Development Fund of TUT, 2006

Ustus Agur scholarship, Estonian Association of Information Technology and telecomminications (ITL), 2006

Estonian Information Technology Foundation "Tiger University" grant for ITC students, 2006

The Jaan Poska joint scholarship of the City of Tallinn and Tallinn University of Technology, 2005

Port of Tallinn scholarship, Development Fund of TUT, 2005

Estonian Information Technology Foundation "Tiger University" grant for ITC students, 2004

ELULOOKIRJELDUS

Isikuandmed

Ees- ja perekonnanimi	Tarmo Robal
Sünniaeg	14.06.1979
Sünnikoht	Eesti
Kodakondsus	Eesti

Kontaktandmed

Aadress	Raja 15, 12618 Tallinn
Telefon	620 2263
E-post	tarmo.robal@ati.ttu.ee

Hariduskäik

2003 – ...	Doktorantuur, Info- ja kommunikatsioonitehnoloogia, Tallinna Tehnikaülikool (TTÜ)
2001 – 2003	Tehnikateaduste magister (M.Sc.), TTÜ
1997 – 2001	Tehnikateaduste bakalaureus (B.Sc.) TTÜ
1986 – 1997	Keskharidus, Tallinna Lilleküla Keskkool

Teenistuskäik

2004 – ...	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond Arvutitehnika instituut, Süsteemitarkvara õppetool *teadur*
2003 – 2007	Eesti Infotehnoloogia Kolledž, *lektor*
2003 – 2004	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond Arvutitehnika instituut, *vaneminsener*
2001 – 2002	AS Sampo Pank, *tarkvaradisainer*
1998 – 2002	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond Arvutitehnika instituut, *insener*

Teadustegevus

Programmkomitee liige

- Int’l. Conf. on Advances in Semantic Processing SEMAPRO, al. 2009
- Annual ACM Symposium on Applied Computing, The Semantic Web and Applications SWA, alates 2008
- Int. Conference on Information Systems Development ISD, 2008, 2009
- The Joint International Workshop on Metamodels, Ontologies, Semantic Technologies, and Information Systems for the Semantic Web MOST-ONISW, 2009
- The 2nd International workshop on Ontologies and Information Systems for the Semantic Web ONISW, 2008

Retsenseerimine

- 10th International Baltic Conference on Databases and Information Systems (Baltic DB&IS 2012)
- 12th Biennial Baltic Electronics Conference (BEC2010)
- East-European Conference on Advances in Databases and Information Systems (ADBIS 2010, ADBIS 2012)

Publikatsioonid

Robal, T., Kalja, A., Learning from Users for a Better and Personalized Web Experience, PICMET '12 Conference "Technology Management for Emerging Technologies", Jul 29 - Aug 2, 2012, Vancouver, Canada, 10p. [to be published]

Robal, T., Kalja, A. Applying User Domain Model to Improve Web Recommendations. Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 14p. [to be published]

Kalja, A., Põld, J., Robal, T., Vallner, U. Modernization of the e-government in Estonia. PICMET '11 : Proceedings Technology Management in the Energy-Smart World. D. Kocaoglu, T. Anderson, T. U. Daim (Eds). Portland, OR, USA. PICMET, 2011, pp. 3151 - 3157.

Robal, T., Kann, T., Kalja, A. An ontology-based intelligent learning object for teaching the basics of digital logic. Proceedings of 2011 IEEE International Conference on Microelectronic Systems Education (MSE). IEEE Computer Society, 2011, pp. 106-107.

Kalja, A., Robal, T., Vallner, U. Towards information society: Estonian case study. Proceedings of PICMET '09: Technology Management in the Age of Fundamental Change. D. Kocaoglu, T. Anderson, T. U. Daim (Eds). Portland, Oregon USA. PICMET, 2009, pp. 3218-3225.

Robal, T., Kalja, A. Web systems evaluation on users' behaviour modelling. Databases and Information Systems V: Selected Papers from the 8th Intl. Baltic Conference, DB&IS 2008. H-M. Haav,A. Kalja (Eds). Amsterdam, IOS Press, Frontiers in Artificial Intelligence and Applications, 187, 2009, pp. 41-52.

Robal, T., Kalja, A. Conceptual web users' actions prediction for ontology-based browsing recommendations. Information Systems Development: Towards a Service Provision Society. New York: Springer-Verlag, 2009, pp. 121-129.

Robal, T., Kalja, A. A model for users' action prediction based on locality profiles. Information Systems Development: Challenges in Practice, Theory and Education. C. Barry, K. Conboy, M. Lang, G. Wojtkowski, W. Wojtkowski, Wita (Eds). New York: Springer, Vol. 1, 2009, pp. 169-182.

Robal, T., Kalja, A. Creating interactive learning objects with web services. Proceedings of the 20th EAEEIE Annual Conference. IEEE, 2009, pp. 1-6.

Robal, T., Kalja, A. Interactive Hands-On Tools as Learning Objects on Web Services. Proceedings of International Conference on Microelectronic Systems Education MSE'09. San Francisco, CA, USA. IEEE, 2009. pp. 73-76.

Robal, T., Kalja, A. Evaluations for improving web systems on the basis of users behaviour modelling. Databases and Information Systems: Proceedings of the 8th Intl Baltic Conference, Baltic DB&IS 2008. H-M. Haav, A. Kalja (Eds). Tallinn: Tallinn University of Technology Press, 2008, pp. 229-240.

Põld, J., Kalja, A., Robal, T. A five step refactoring process: improving software design properly. Databases and Information Systems: Proceedings of the 8th Intl Baltic Conference, Baltic DB&IS 2008. H-M. Haav, A. Kalja (Eds). Tallinn: Tallinn University of Technology Press, 2008, pp. 37-48.

Robal, T., Kalja, A. Enabling students contemporary ways of learning using e-supported courses. 19th EAEEIE Annual Conference, Formal Proceedings. The 19th European Association for Education in Electrical and Information Engineering Annual Conference. Tallinn, Estonia, IEEE, 2008, pp. 14-19.

Robal, T., Kalja, A. Making use of personalized web services in the study process. Proceedings: 11th Biennial Baltic Electronics Conference BEC 2008. Tallinn, Estonia, 2008, pp. 211-212.

Robal, T., Kalja, A. Evaluating Web systems on the basis of users' behavior modeling. IKTDK Annual Conference, Estonia, TUT Press, 2008, pp. 64-67.

Põld, J., Kalja, A., Robal, T. A five step refactoring process: improving software design properly. IKTDK Annual Conference, Estonia, TUT Press, 2008, pp. 93-96.

Kalja, A., Kindel, K., Kivi, R., Robal, T. eGovernment Services: How to Develop Them, How to Manage Them? Proceedings of PICMET'07: Management of Converging Technologies: PICMET '07 Portland International Center for Management of Engineering and Technology. Portland OR, USA, IEEE, 2007, pp. 2795-2798.

Robal, T., Kalja, A. Applying user profile ontology for mining web site adaptation recommendations. Local Proceedings of the 11th East-European Conference on Advances in Databases and Information Systems (ADBIS 2007).

Y. Ioannidis, B. Novikov, B. Rachev (Eds). Varna, Bulgaria: Technical University of Varna, 2007, pp. 126-135.

Robal, T., Haav, H-M., Kalja, A. Making web users' domain models explicit by applying ontologies. Advances in Conceptual Modeling - Foundations and Applications : ER 2007 Workshops CMLSA, FP-UML,ONISW, QoIS, RIGiM, SeCoGIS. J.-L. Hainaut (Ed). Auckland, New Zealand, Berlin: Springer, 2007, Lecture Notes in Computer Science, vol. 4802, pp. 170-179.

Robal, T., Kalja, A. Applying e-environments in teaching the basics of digital logics. Proceedings of the IEEE International Conference on Microelectronic Systems Education MSE 2007. P. Kellenberger (Ed), San Diego, CA, USA. IEEE Computer Society Press, 2007, pp. 41-42.

Robal, T., Kalja, A. Applying ICT for Improving the Study Process. Proceedings of 18th EAEEIE Annual Conference, Innovation in Education for Electrical and Information Engineering. Prague, Czech Technical University in Prague, 2007, pp. 1-5.

Robal, T. Applying User Profile Ontology for Mining Web Site Adaptation Recommendations. IKTDK Annual Conference, Estonia, 2007, pp. 125-128.

Robal, T., Kalja, A., Põld, J. Analysing the Web Log to Determine the Efficiency of Web Systems. Seventh International Baltic Conference on Databases and Information Systems. O. Vasilecas, J. Eder, A. Caplinskas (Eds). Vilnius, Lithuania, Technika, 2006, pp. 264-275.

Robal, T., Kalja, A. Moving studies to e-environments: a case study. Current Developments in Technology-Assisted Education. A. Méndez-Vilas, M. Solano Martín, J.A. Mesa González, J. Mesa González. Badajoz, Spain: Formatex, 2006, pp. 936-940.

Robal, T., Brik, M., Aarna, M. e-Learning of Digital Logic. Using e-Environments in Teaching Digital Logic. Proc: 6th International Workshop on Microelectronics Education EWME 2006. Stockholm, Sweden, Royal Institute of Technology KTH, 2006, pp. 120-123.

Robal, T. Analysing the Web Log to Determine the Efficiency of Web Systems. IKTDK Annual Conference, 2006, pp. 64-67.

Robal, T., Kalja, A. e-EDU - an information system for e-learning services. Selected Papers from Sixth International Baltic Conference DB&IS'2004. J. Barzdins, A. Caplinskas (Eds). Amsterdam: IOS Press, Frontiers in artificial intelligence and applications, vol. 118, 2005, pp. 288-298.

Kruus, H., Orasson, E., Robal, T., Ubar, R. Investigating Defects in Digital Circuits by Boolean Differential Equations. The 4th Int. Conference "Distance Learning - Educational Sphere of XXI Century" DLESC'04. Minsk, 2004, pp. 432-435.

Robal, T., Kalja, A. e-EDU - an information system for e-learning services. Acta Universitatis Latviensis, vol. 672, 2004, pp. 469-480.

Robal, T., Kruus, H. e-Bibliothecula - A Virtual Library Service. 13th Intl. Conference on Information Systems Development. Advances in Theory, Practice and Education, Lithuania, 2004, pp. 139-148.

Robal, T., Viies, V., Kruus, M. The Rational Unified Process with the "4+1" View Model of Software Architecture - a Way for Modeling Web Applications. Proc. of the Fifth Int. Baltic Conference: BalticDB&IS 2002, Tallinn, Estonia, 2002, pp. 119-132.

Kaitstud lõputööd

2003 Magistritöö, *Using the RUP with 4+1 Views of Software Architecture for Web Applications Development.*
TTÜ, Arvutitehnika instituut, juhendaja dots. V. Viies.

2001 Bakalaureusetöö, *Internetirakendus Tallinna Tehnikaülikooli arvutitehnika instituudi dünaamiline veebileht.* TTÜ, Arvutitehnika instituut, juhendaja dots. M. Kruus.

Teadustöö põhisuunad

Veebiinfosüsteemid, veebikaevandamine, adaptiivne veeb, veebi personaliseerimine, e-õppekeskonnad, veebipõhised õppevahendid.

Juhendatud lõputööd

Janari Põld, *Improving software development and maintainability: A refactoring process model for use in evolving systems (Tarkvara parendamine läbi refaktoriseerimise protsessi)*, 2007, tehnikateaduste magister, *Cum Laude.*

Teaduspreemiad ja tunnustused

TTÜ ja Tallinna Linnavalitsuse Jaan Poska nimeline stipendium, 2007

AS Eesti Energia stipendium, TTÜ Arengufond, 2006

Ustus Aguri nimeline stipendium, ITL, 2006

Eesti Infotehnoloogia Sihtasutuse Tiigriülikooli stipendium, 2006

TTÜ ja Tallinna Linnavalitsuse Jaan Poska nimeline stipendium, 2005

AS Tallinna Sadam stipendium, TTÜ Arengufond, 2005

Eesti Infotehnoloogia Sihtasutuse Tiigriülikooli stipendium, 2004

DISSERTATIONS DEFENDED AT TALLINN UNIVERSITY OF TECHNOLOGY ON *INFORMATICS AND SYSTEM ENGINEERING*

1. **Lea Elmik**. Informational Modelling of a Communication Office. 1992.

2. **Kalle Tammemäe**. Control Intensive Digital System Synthesis. 1997.

3. **Eerik Lossmann**. Complex Signal Classification Algorithms, Based on the Third-Order Statistical Models. 1999.

4. **Kaido Kikkas**. Using the Internet in Rehabilitation of People with Mobility Impairments – Case Studies and Views from Estonia. 1999.

5. **Nazmun Nahar**. Global Electronic Commerce Process: Business-to-Business. 1999.

6. **Jevgeni Riipulk**. Microwave Radiometry for Medical Applications. 2000.

7. **Alar Kuusik**. Compact Smart Home Systems: Design and Verification of Cost Effective Hardware Solutions. 2001.

8. **Jaan Raik**. Hierarchical Test Generation for Digital Circuits Represented by Decision Diagrams. 2001.

9. **Andri Riid**. Transparent Fuzzy Systems: Model and Control. 2002.

10. **Marina Brik**. Investigation and Development of Test Generation Methods for Control Part of Digital Systems. 2002.

11. **Raul Land**. Synchronous Approximation and Processing of Sampled Data Signals. 2002.

12. **Ants Ronk**. An Extended Block-Adaptive Fourier Analyser for Analysis and Reproduction of Periodic Components of Band-Limited Discrete-Time Signals. 2002.

13. **Toivo Paavle**. System Level Modeling of the Phase Locked Loops: Behavioral Analysis and Parameterization. 2003.

14. **Irina Astrova**. On Integration of Object-Oriented Applications with Relational Databases. 2003.

15. **Kuldar Taveter**. A Multi-Perspective Methodology for Agent-Oriented Business Modelling and Simulation. 2004.

16. **Taivo Kangilaski**. Eesti Energia käiduhaldussüsteem. 2004.

17. **Artur Jutman**. Selected Issues of Modeling, Verification and Testing of Digital Systems. 2004.

18. **Ander Tenno**. Simulation and Estimation of Electro-Chemical Processes in Maintenance-Free Batteries with Fixed Electrolyte. 2004.

19. **Oleg Korolkov**. Formation of Diffusion Welded Al Contacts to Semiconductor Silicon. 2004.

20. **Risto Vaarandi**. Tools and Techniques for Event Log Analysis. 2005.

21. **Marko Koort**. Transmitter Power Control in Wireless Communication Systems. 2005.

22. **Raul Savimaa**. Modelling Emergent Behaviour of Organizations. Time-Aware, UML and Agent Based Approach. 2005.

23. **Raido Kurel**. Investigation of Electrical Characteristics of SiC Based Complementary JBS Structures. 2005.

24. **Rainer Taniloo**. Ökonoomsete negatiivse diferentsiaaltakistusega astmete ja elementide disainimine ja optimeerimine. 2005.

25. **Pauli Lallo.** Adaptive Secure Data Transmission Method for OSI Level I. 2005.

26. **Deniss Kumlander**. Some Practical Algorithms to Solve the Maximum Clique Problem. 2005.

27. **Tarmo Veskioja**. Stable Marriage Problem and College Admission. 2005.

28. **Elena Fomina**. Low Power Finite State Machine Synthesis. 2005.

29. **Eero Ivask**. Digital Test in WEB-Based Environment 2006.

30. **Виктор Войтович**. Разработка технологий выращивания из жидкой фазы эпитаксиальных структур арсенида галлия с высоковольтным p-n переходом и изготовления диодов на их основе. 2006.

31. **Tanel Alumäe**. Methods for Estonian Large Vocabulary Speech Recognition. 2006.

32. **Erki Eessaar**. Relational and Object-Relational Database Management Systems as Platforms for Managing Softwareengineering Artefacts. 2006.

33. **Rauno Gordon**. Modelling of Cardiac Dynamics and Intracardiac Bio-impedance. 2007.

34. **Madis Listak**. A Task-Oriented Design of a Biologically Inspired Underwater Robot. 2007.

35. **Elmet Orasson**. Hybrid Built-in Self-Test. Methods and Tools for Analysis and Optimization of BIST. 2007.

36. **Eduard Petlenkov**. Neural Networks Based Identification and Control of Nonlinear Systems: ANARX Model Based Approach. 2007.

37. **Toomas Kirt**. Concept Formation in Exploratory Data Analysis: Case Studies of Linguistic and Banking Data. 2007.

38. **Juhan-Peep Ernits**. Two State Space Reduction Techniques for Explicit State Model Checking. 2007.

39. **Innar Liiv**. Pattern Discovery Using Seriation and Matrix Reordering: A Unified View, Extensions and an Application to Inventory Management. 2008.

40. **Andrei Pokatilov**. Development of National Standard for Voltage Unit Based on Solid-State References. 2008.

41. **Karin Lindroos**. Mapping Social Structures by Formal Non-Linear Information Processing Methods: Case Studies of Estonian Islands Environments. 2008.

42. **Maksim Jenihhin**. Simulation-Based Hardware Verification with High-Level Decision Diagrams. 2008.

43. **Ando Saabas**. Logics for Low-Level Code and Proof-Preserving Program Transformations. 2008.

44. **Ilja Tšahhirov**. Security Protocols Analysis in the Computational Model – Dependency Flow Graphs-Based Approach. 2008.

45. **Toomas Ruuben**. Wideband Digital Beamforming in Sonar Systems. 2009.

46. **Sergei Devadze**. Fault Simulation of Digital Systems. 2009.

47. **Andrei Krivošei**. Model Based Method for Adaptive Decomposition of the Thoracic Bio-Impedance Variations into Cardiac and Respiratory Components. 2009.

48. **Vineeth Govind**. DfT-Based External Test and Diagnosis of Mesh-like Networks on Chips. 2009.

49. **Andres Kull**. Model-Based Testing of Reactive Systems. 2009.

50. **Ants Torim**. Formal Concepts in the Theory of Monotone Systems. 2009.

51. **Erika Matsak**. Discovering Logical Constructs from Estonian Children Language. 2009.

52. **Paul Annus**. Multichannel Bioimpedance Spectroscopy: Instrumentation Methods and Design Principles. 2009.

53. **Maris Tõnso**. Computer Algebra Tools for Modelling, Analysis and Synthesis for Nonlinear Control Systems. 2010.

54. **Aivo Jürgenson**. Efficient Semantics of Parallel and Serial Models of Attack Trees. 2010.

55. **Erkki Joasoon**. The Tactile Feedback Device for Multi-Touch User Interfaces. 2010.

56. **Jürgo-Sören Preden**. Enhancing Situation – Awareness Cognition and Reasoning of Ad-Hoc Network Agents. 2010.

57. **Pavel Grigorenko**. Higher-Order Attribute Semantics of Flat Languages. 2010.

58. **Anna Rannaste**. Hierarcical Test Pattern Generation and Untestability Identification Techniques for Synchronous Sequential Circuits. 2010.

59. **Sergei Strik**. Battery Charging and Full-Featured Battery Charger Integrated Circuit for Portable Applications. 2011.

60. **Rain Ottis**. A Systematic Approach to Offensive Volunteer Cyber Militia. 2011.

61. **Natalja Sleptšuk**. Investigation of the Intermediate Layer in the Metal-Silicon Carbide Contact Obtained by Diffusion Welding. 2011.

62. **Martin Jaanus**. The Interactive Learning Environment for Mobile Laboratories. 2011.

63. **Argo Kasemaa**. Analog Front End Components for Bio-Impedance Measurement: Current Source Design and Implementation. 2011.

64. **Kenneth Geers**. Strategic Cyber Security: Evaluating Nation-State Cyber Attack Mitigation Strategies. 2011.

65. **Riina Maigre**. Composition of Web Services on Large Service Models. 2011.

66. **Helena Kruus**. Optimization of Built-in Self-Test in Digital Systems. 2011.

67. **Gunnar Piho**. Archetypes Based Techniques for Development of Domains, Requirements and Sofware. 2011.

68. **Juri Gavšin**. Intrinsic Robot Safety Through Reversibility of Actions. 2011.

69. **Dmitri Mihhailov**. Hardware Implementation of Recursive Sorting Algorithms Using Tree-like Structures and HFSM Models. 2012.

70. **Anton Tšertov**. System Modeling for Processor-Centric Test Automation. 2012.

71. **Sergei Kostin**. Self-Diagnosis in Digital Systems. 2012.

72. **Mihkel Tagel**. System-Level Design of Timing-Sensitive Network-on-Chip Based Dependable Systems. 2012.

73. **Juri Belikov**. Polynomial Methods for Nonlinear Control Systems. 2012.

74. **Kristina Vassiljeva**. Restricted Connectivity Neural Networks based Identification for Control. 2012.