Adaptive hypermedia courses : qualitative and quantitative evaluation and tool support

DOI:
10.6100/IR772869

Document status and date:
Published: 01/01/2014

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Adaptive Hypermedia Courses:

Qualitative and Quantitative Evaluation and Tool Support

Vinicius Faria Culmant Ramos

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3607-8

SIKS Dissertation Series No. 2014-19
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

MIX
Paper from responsible sources
FSC® C112051

Printed by: ipskampdrukkers.nl

Cover Design: Silvia Esteves Duarte
Cover images: http://pngimg.com/img/people/hands

http://www.sxc.hu/photo/1146139

Adaptive Hypermedia Courses:

Qualitative and Quantitative Evaluation and Tool Support

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus, prof.dr.ir. C.J. van Duijn, voor een commissie aangewezen door het College voor Promoties in het openbaar te verdedigen op dinsdag 29 april 2014 om 14.00 uur

door

Vinicius Faria Culmant Ramos

geboren te Porto Velho, Brazilië

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. E.H.L. Aarts
1e promotor:	prof.dr. P.M.E. De Bra
2e promotor:	prof.dr. G.B. Xexéo (Federal University of Rio de Janeiro)
copromotor:	dr. M. Pechenizkiy
leden:	prof.dr. G. Zimbrão da Silva (Federal University of Rio de Janeiro)
	prof.dr. D. Schwabe (Pontifical Catholic University of Rio de Janeiro)
	prof.dr. M. Specht (Open University The Netherlands)
	prof.dr. G.W.M. Rauterberg

Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
COPPE UFRJ

ADAPTIVE HYPERMEDIA COURSES: QUALITATIVE AND QUANTITATIVE EVALUATION AND TOOL SUPPORT

Vinicius Faria Culmant Ramos

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Geraldo Bonorino Xexéo
Paul Maria Emile De Bra

Rio de Janeiro
Novembro de 2013

ADAPTIVE HYPERMEDIA COURSES: QUALITATIVE AND QUANTITATIVE EVALUATION AND TOOL SUPPORT

Vinicius Faria Culmant Ramos

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. Paul Maria Emile De Bra, Ph.D.

Prof. Daniel Schwabe, Ph.D.

Prof. Geraldo Zimbrão da Silva, D.Sc.

Profa. Miriam Struchiner, Ph.D.

Dr. Mykola Pechenizkiy, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

NOVEMBRO DE 2013

Ramos, Vinicius Faria Culmant

Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support/ Vinicius Faria Culmant Ramos. — Rio de Janeiro: UFRJ/COPPE, 2013.

XXX, 198 p.: il.; 29,7 cm

Orientadores: Geraldo Bonorino Xexéo

Paul Maria Emile De Bra

Tese (doutorado) — UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2013.

Referências Bibliográficas: p. 125-134.

1. Adaptive Hypermedia. 2. Evaluation. 3. Adaptive Course I. Xexéo, Geraldo Bonorino *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Esta tese foi realizada em co-tutela entre a
Universidade Federal do Rio de Janeiro/COPPE
e a Technische Universiteit Eindhoven, aprovada
pelo CEPG, Conselho de Ensino para Graduados, de
acordo com o Processo nº 23079.064290/2013-90.

To my wife Marina and
my wonderful children
Júlia and Maurício,
you make my life easier
and happier.

Acknowledgement

First of all, I would like to thank my parents. This work would not have been possible without their support and assistance during my whole life. Thanks Dad (Sr. Claudio) and Mom (D. Sonia). Both of you deserve all my love and gratitude.

I would also like to thank my brother and sisters: Carlos, Cássia and Isabela. Thank you for your care, laughs and support in the moments I needed, even though sometimes I did not deserve that much. Many thanks to my sister Cássia and her beautiful family: Felipe (husband), Nathália (daugther) and Maria Eduarda (daugther). It is an easy task to talk about my youngest sister (Isabela), since the youngest is always the one loved by everyone, and she is certainly no exception. My brother, his wife (Ana) and their amazing happy daughter (Clara) are more than special to me. I want to thank you so much for all the funny and lovely moments we spent together.

My happiness and my life would not have been complete without these three wonderful people: Marina (wife), Júlia (daughter) and Maurício (son). They are the most special people in the world to me. I do not have words to thank you all. Júlia, you are 5 years old now and I spent most of the happiest moments of my life at your side. Even if you do not know yet, you have given me the incentives to face my challenges and to keep going over the problems every day. Maurício, you are less than a year old, and you, like your sister, also made my life happier. You are a beautiful baby and your constant smile makes my life easier. It is not easy to thank my wonderful wife, Marina, using only words, since she is the person that supports me in all my decisions, and shares all the easy and difficult moments of life at my side. Marina, you are amazing! Thank you for all your unending support, and for all the nights I had to stay awake to write the thesis or to write research papers. Thank you for helping me during such stressful moments. Thank you for sharing the greatest moments in my life. THANK YOU, Marina!

Thank you Vera (mother-in-law), Pedro (brother-in-law) and Juliana (wife's cousin) for the support and funny moments in the last three years we stayed closer.

Thank you Alexandre Stauffer. You have been a great friend for years. You are not only a good friend, but you are also the person who gave me support for continuing to study, making research and also developing all the academic skills I have today. You read each part of my thesis and papers, not because you had to, but because you are an amazing friend (and researcher). Your criticism improved all my academic qualities, and, consequently, my research. In my place you have free bier for the rest of your life, you only have to ask for them. Thank you very very very much.

There are a lot of friends I would like to thank. First of all, I would like to thank my friends from my undergraduate at UFRJ: Calisto, Izaías, Paulo Nunes, Raphael (Dill), Taísa and Targino. I would like to thank the secretaries of PESC/COPPE: Ana Paula, Claudia, Guty, Mercedes, Patrícia and Solange. Thank you, Mrs. Riet and Mrs. Ine, you were amazingly nice and kind during my stay at TU/e. Many thanks to my friends from Eindhoven: Antonino Simone, Alberto Perrota, Camille Carcoute, Evegny Knutov, Gierri, Ivelina, Jorge, Marcos, Mayla Bruso and Nicolas Monti.

I would also like to thank Eindhoven University of Technology (TU/e), specially the Department of Mathematics and Computer Science, for the support in my research and for having me as a visitor. Thanks to all those people that I met in the Web Engineering research group at TU/e. I have spent valuable moments during my stay in Eindhoven.

I am very thankful for Dr. David Smits for all your help. Since I have started my research with the AHA! system (and later the GALE system), Dr. David Smits gave me a lot of support in the development of new code to the AHA! and GALE system.

Thank you Prof. Miriam Struchiner for being in my thesis committee at UFRJ. Prof. Miriam has played a major role in my academic life. She was responsible for my first steps in research, accepting me as an undergraduate student into her lab in 2002. Prof. Miriam deserves all my gratitude and admiration.

Thank you Prof. Dr. Marcus Specht and Prof. Dr. Matthias Rauterberg for accepting to be in my thesis committee at TU/e, and thank you Prof. Dr. Daniel Schwabe and Prof. Dr. Geraldo Zimbrão for accepting to be in my thesis committee at UFRJ and TU/e.

I am thankful to Dr. Mykola Pechenizkiy to be in my thesis committee at UFRJ and TU/e, and also for giving me fundamental and important feedback on my research. He was also very kind and nice during my stay at TU/e. Thank you very much.

At last, but not less important, I am thankful to my supervisors: Prof. Dr. Paul De Bra and Prof. Dr. Geraldo B. Xexéo.

Prof. Dr. Xexéo gave me the opportunity to work with him in his research group and was the first one to encourage my mobility to TU/e in 2009. His friendship and kindness brought us into a closer contact than that between a supervisor and a student. I know I can count on him.

I have all my gratitude to Prof. Dr. De Bra. You are one of the greatest researchers I have ever met. You are more than a researcher, you are a nice and generous person. Your academic and personal support before, during and after my stay in Eindhoven were invaluable to me, and have made me have a great admiration for you as a researcher and a person. I have learned a lot with you, I am thank you for every single word, method, technique, academic stuffs etc you taught me. Thank you very very very much!

Many thanks to CNPq for the financial support during most part of this thesis.

Resumo da Tese apresentada à COPPE/UFRJ e à TU/e como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

ADAPTIVE HYPERMEDIA COURSES: QUALITATIVE AND QUANTITATIVE EVALUATION AND TOOL SUPPORT

Vinicius Faria Culmant Ramos
Novembro/2013

Orientadores: Geraldo Bonorino Xexéo
Paul Maria Emile De Bra

Programa: Engenharia de Sistemas e Computação

O foco deste trabalho é a avaliação de cursos adaptativos criados e entregues pelo AHA! (Adaptive Hypermedia Architecture) and GALE (Generic Adaptation Language and Engine desenvolvido dentro do projeto EU FP7 GRAPPLE). O objetivo destas avaliações é entender a influência da adaptação no aprendizado dos alunos de um curso adaptativo. Os métodos avaliativos são divididos em qualitativo e quantitativo. Os métodos quantitativos consistem na análise dos *logs*, testes e notas de avaliações dos alunos. As análises de questionários fazem parte dos métodos qualitativos. Também fizemos neste trabalho a avaliação da modularidade e extensibilidade do GALE como parte da preocupação em ter um sistema como esse com apenas um núcleo de adaptação que pode ser extendido para ser usado em diferentes tipos de sistemas adaptativos. Esta tese também apresenta ferramentas de análise de logs de navegação e de resultados de testes e questionários dos cursos adaptativos no GALE. O principal objetivo destas ferramentas é auxiliar os autores apresentando-os medidas estatísticas de seu próprio curso, permitindo a eles uma análise da estutura do curso do ponto de vista da navegação dos alunos. No final, discutimos os trabalhos futuros, em especial as sugestões de mudanças na configuração do GALE (para desenvolvedores que precisam extender o sistema) e na estrutura de cursos baseado nas observações dos comportamentos dos alunos e nas sugestões apresentadas por eles.

Abstract of Thesis presented to COPPE/UFRJ and to TU/e as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

ADAPTIVE HYPERMEDIA COURSES: QUALITATIVE AND QUANTITATIVE EVALUATION AND TOOL SUPPORT

Vinicius Faria Culmant Ramos
November/2013

Advisors: Geraldo Bonorino Xexéo
Paul Maria Emile De Bra

Department: Systems Engineering and Computer Science

The focus of this work is the evaluation of adaptive courses created and delivery by AHA! (the Adaptive Hypermedia Architecture) and GALE (the Generic Adaptation Language and Engine developed in the EU FP7 project GRAPPLE). The main goal of these evaluations is to understand the influence of adaptation on students' learning in an adaptive hypertext course. The evaluation methods are divided into qualitative and quantitative ones. The quantitative methods consist of analysis of the students' logs, the performed tests and assignment grades. The analysis of questionnaires is part of the qualitative method. In this work we also performed an evaluation of the modularity and extensibility of the GALE system as part of the concern in having such a system with a single core adaptation engine that can be extended in order to use it for different types of adaptation. This thesis also presents tools for the analysis of navigation logs and test and quiz results of adaptive courses in GALE. The main goal of these tools is to assist the courses' authors to retrieve statistical measurements for their own courses, allowing them to analyze the structure of the course from the point of view of the students' navigation. At the end we discuss future work, and in particular suggest changes to the setup of GALE (for developers that needs to extend the system) and to the structure of hypertext courses based on the observed student behavior as well as the student feedback.

Samenvatting van het proefschrift ingediend bij COPPE/UFRJ en de TU/e ter verkrijging van de graad van Doctor.

ADAPTIEVE HYPERMEDIA LEERMATERIAAL: KWALITATIEVE EN KWANTITATIEVE EVALUATIE EN ONDERSTEUNENDE HULPMIDDELEN

Vinicius Faria Culmant Ramos

November/2013

Promotores: Geraldo Bonorino Xexéo en Paul Maria Emile De Bra

Department: Systems Engineering and Computer Science

De focus van dit werk is de evaluatie van adaptief leermateriaal dat gemaakt is voor en aangeboden door het AHA! system (Adaptive Hypermedia Architecture) en door GALE (Generic Adaptation Language and Engine, ontwikkeld in het EU FP7 project GRAPPLE). Het hoofddoel van deze evaluaties is om de invloed te begrijpen die het gebruik van adaptief leermateriaal heeft op het leergedrag van studenten. De evaluatie bestaat uit kwalitatieve en kwantitatieve methoden. De kwantitatieve methoden bestaan uit een analyse van de logbestanden met betrekking tot de navigatie door studenten, de uitgevoerde toetsen en de eindbeoordeling van opdrachten. De kwalitatieve methode bestaat uit de analyse van vragenlijsten. In dit werk hebben we ook een evaluatie uitgevoerd van de modulariteit en uitbreidbaarheid van het GALE systeem, bedoeld om met een uitbreidbare kern verschillende types adaptatie te ondersteunen. Dit proefschrift presenteert ook hulpmiddelen voor de analyse van navigatie-logbestanden en toets en quiz resultaten behaald in GALE leermateriaal. Het hoofddoel van deze hulpmiddelen is om de auteurs van leermateriaal te helpen om statistische metingen te verkrijgen voor hun leermateriaal enom hen toe te laten de structuur van het leermateriaal te analyseren met het oogpunt van navigatie door studenten. Aan het eind van het proefschrift bespreken we toekomstig werk, waarbij we in het bijzonder uitbreidingen voorstellen in de opstelling van GALE (voor ontwikkelaars die het systeem moeten uitbreiden) en in de structuur van hypertekst leermateriaal, gebaseerd op observatie van het gedrag van studenten en op terugkoppeling door studenten.

Contents

List of Figures

List of Tables

1 Introduction

In the e-learning field, Bates [3] says that today the researchers do not have to discuss whether the technology enhances learning; instead the discussion must be about the way the technology is used for learning. Besides, Bates discusses the use of learning environments through the Web.

Traditional e-learning systems were created to make learning material available and reachable, and to present tasks, activities and evaluations, but they did not allow for user interaction. Furthermore, these systems present their content (learning material, activities and evaluation) always in the same way not taking into account the skills and the characteristics of the users. For example, in a traditional system if we present a list with the main topics of a course, all students will see the same list, and in the same way (topics in the same order, links in the same color and style etc). It is often the case that different students in a course have different learning styles, goals, interests etc. Hence, the development of personalized, extensible and generic systems, which can be adapted to students' diversity, is a very powerful tool and a very big challenge.

E-Learning systems were initially developed to assist users to reach their goals and make these systems more attractive. In this way, many educational systems were developed using artificial intelligence techniques: for example, the tutor systems [34, 6] and adaptive systems [11, 12, 19]. An example of Intelligent Tutoring Systems (ITS) is presented by Brusilovsky in [6]. In this paper, the author refers to an on-line course that presents a button to the students, which we call *suggestion button*, that links the actual content to the most relevant topic (system's suggestion) in the course. This *suggestion button* intended to guide students that find themselves lost in the system, without knowing which content they should read in the course.

In the beginning, the main goal of these systems was to assist the users to solve proposed problems and tests. The authors assumed that the users learned the content of the course outside the on-line environment, by means of books and classes. In this way, the ITS developers (and authors) decided to combine the ITS assistance with learning materials, using a hypertext and hypermedia environment. The union of the ITS and the hypermedia offered much more functionality to the system and to the users than the static educational environment. There were a few system, which we now classify as adaptive hypermedia (AH) systems, whose development was influenced by the ITS [12, 45, 37].

1.1 Motivation

The concept of AH goes back a long way, all the way to Vannevar Bush' article "As We May Think" [15] if we interpret the text in a sufficiently creative way. The "Memex" device envisioned by Bush was a form of hypertext or even hypermedia because it was not limited to text. The user could link documents, which for the device would imply that retrieving one document would automatically also retrieve and display the other. The user could build "trails of his interest through the maze of materials available to him", which is a form of personalization (a personalized guided tour in fact). This personalization was aimed at coping with information overload, an all too common problem of Internet users today. Furthermore, the personalization also included adding some kind of annotation: "he inserts a page of longhand analysis of his own". This shows that just linking information items (possibly from different authors) may not constitute a coherent story, which reveals the importance of adding the annotations or what we would call content adaptation.

The personalization envisioned by Bush was aimed at *revisiting* information (finding it again through trails), and at *recalling* a previously discovered meaning. When Bush defined *trailblazing* as a possible new profession we can understand this as preparing personalization for others.

AH systems, summarized by Brusilovsky in 1996 [7] and 2001 [8], and further updated and detailed by Knutov et al. in 2009 [49], are systems that build, for each user, an individual model and apply it to adapt the application (e.g. an on-line course) to that user. The individual model is constructed based on the preferences, goals and evolving knowledge of the users. The AH aims at automating the Bush's "trailblazing" through *link adaptation* and the annotations through *content adaptation*. The AHAM reference model [21, 80] captured these methods and techniques into an abstract adaptive hypermedia architecture.

The AHA! system [26] was developed in parallel with the AHAM reference model [21, 80, 22]. Actually, AHA! was originally designed to serve a hypertext course taught through the Web [19]. The AHA! system was under development for almost the last 20 years. The main goal of the AHA! developers was to make it as generic as possible, and to make it able to serve different adaptive methods and techniques and, consequently, different adaptive applications.

A lot has happened since AHAM and AHA! and the other models and systems of the early 21^{st} century. Knutov et al. [49] describe many new adaptation techniques developed to provide a list of challenges for creating a new generic adaptive hypermedia system. That research has led to the GAF model (Generic Adaptation Framework) [50, 48], which has been designed to be capable of dealing with several types of adaptive systems and applications, from "traditional" adaptive hypermedia to personalized search and recommender systems. Designing a new adaptive system that encompasses all new techniques, from *open corpus adaptation*, through *domain models based on ontologies*, *group adaptation* (and *group formation*), *higher order adaptation* based on *web-log mining*, *context-awareness*, *information retrieval* and *recommender systems*, is an impossible task. GALE tackles this challenge through a very modular and extensible approach [67, 66, 29], consisting of an extensible generic and general purpose adaptive hypermedia engine. GALE was inspired and developed over the code of the AHA! system with a significant architectural difference: the separation of the *adaptive engine* from the *user modeling service* [28]. For this reason, we say that GALE is an evolution of the AHA! system.

It is very important to develop generic and general purpose systems, but it is also important to develop tools for the authors that create applications in such systems to evaluate them. For example, in an adaptive course it is important to the author to discover whether the course is navigated as he expected, and whether the students are learning from the proposed test.

The first works about evaluation of AH systems compared adaptive systems with non-adaptive versions of the same systems [43, 12, 16]. In general, the main goal of this kind of evaluation is to compare the efficiency between the two versions or, in some cases, to analyze the success and failure rates of users doing a task. A few years before these works, Totterdell and Boyle [70] already presented potential problems in this kind of evaluation, such as: to determine which system state of the adaptive version must be chosen to compare it with the non-adaptive version, and to specify at which point the comparison must start, since there is a gap in the adaptive version from the starting point of the user's navigation and the apprehension of his needs and characteristic. The authors also describe an evaluation of the AH systems that takes into account the layers of these systems, originally, called as layered evaluation.

The adaptation of an AH system refers to the method or technique used to adapt the system or the content to the user. The most common ones are techniques used to adapt the presentation and/or the navigation of the system to the user based on the *user model*. A good taxonomy about these techniques can be found in [49, 48].

The layers of an AH system are the core of the system. They are responsible for each step of the adaptation. In fact, layers of an AH system are implicit logical divisions of the system, where each of them is responsible for different tasks. For example, in GALE there is an *User Model* (UM) and an *Adaptation Model* (AM) layer. The first one is responsible for the creation and update of the user knowledge, interests, preferences, goals and objectives, style, and other relevant properties that might be useful for adaptation. The AM is responsible to adapt the presentation, information content, and navigation structure to the user's knowledge, interests, goals and objectives, etc. In 2000 the layered evaluation became popular, even though Totterdell and Boyle had described this type of evaluation already in the 90s [70]. Brusilovsky et al. [13] propagated the technique of layered evaluation in their revision of their evaluation of an AH system [9] where they presented the benefits of this technique.

1.2 Research Questions and Hypothesis

This work provides answers to the following research questions:

1. Does the annotation/hiding adaptive technique influence the choices of the students? Does the annotation/hiding adaptive technique constitute an effective model of adaptation and students' guidance?
   The hypothesis we have is that the adaptive technique is an effective model of adaptation and students' guide. To answer these questions we analyzed two versions of an adaptive course that we call *Hypermedia course*. The first analysis was made 3 years after the end of the course, and we used a quantitative approach to measure the influence of the annotation/hiding technique in the students' choice. We used new functions and extensions, as

well as the results of this analysis, to carry out a second evaluation of the Hypermedia course.

2. How does the interplay between the link structure of an adaptive course and the rules employed by the adaptation influence the choices of the students? The hypothesis we have is that the link structure influences the choices made by the students. Here we analyze the connection between the annotation/hiding adaptive technique and the link structure of the course, and their effect on the way the students follow the course.
   In our analysis we focus on the concept of *hubs* and *authorities* defined by Kleinberg [47]. Intuitively, *hubs* are pages that contain a large number of links to others pages. We cannot apply this concept directly in our setting, since in an adaptive application the number of links of a web page may change over time.

3. Can GALE system effectively serve different kinds of adaptive applications to reach its goal as a generic and extensible adaptive system?
   Our hypothesis is that GALE reach its goal as a generic and extensible adaptive system serving different kinds of adaptive applications. In this study, master students were asked to develop new applications and extensions to GALE as part of their master coursework, and we put a questionnaire on-line for a few weeks for the master students to answer regarding their experience. The answers were anonymous to give the students freedom to evaluate the system.

The research described in this thesis is of an exploratory nature. Although logging by the adaptive systems is used and also questionnaires the aim is not to obtain solid statistical evidence but rather indications of patterns that help us in answering the above questions and in defining some new concepts that can be used by authors of adaptive applications in the future.

1.3 Research Goal and Approach

This thesis focuses on the evaluation of adaptive hypermedia courses created and developed with AHA! and GALE. We are also interested in the evaluation of the GALE system as a modular and extensible adaptive system. An important remark is that this research is focused on the evaluation of the AHA! and GALE systems and their applications; thus our goals and inferences are based on these systems only. Our evaluations are intended to provide a better experience to students by helping authors and developers of adaptive applications in AHA! and GALE. Our research is divided in the following phases:

- Study and analyze the research methods and techniques used in empirical evaluations of adaptive systems, web-based information systems, intelligent tutoring systems, and other related fields, in order to apply (or create new ones and apply) them in the evaluation of adaptive courses.

- Evaluate an adaptive course created and delivered by the AHA! system using a selection of empirical and quantitative methods and techniques studied in the previous phase, present the results, problems and pitfalls found in the evaluation. The adaptive course was offered in 2006 and our analysis was carried out 3 years later. The data analyzed (somewhat informally) contains navigation and test logs.

- Evaluate an updated version of the adaptive course delivered by the GALE system with empirical, quantitative and qualitative methods and techniques. The methods and techniques are revisited and adapted from our first evaluation in order to eliminate some problems and pitfalls observed there. The (somewhat informal) empirical and quantitative evaluation was made using the navigation, test and quiz logs, while a questionnaire was used as a qualitative technique.

- Propose an evaluation framework to be implemented in GALE.

- Evaluate the modularity and extensibility of GALE, in order to validate the following goal of GALE: to be a generic and extensible adaptive system, which means that it can be used by researchers and educators without all being forced to use the same type of presentation and adaptation. This evaluation can only be done with small groups of students, so only anecdotal evidence is to be expected.

1.4 Important Definitions

In this section we introduce a few crucial definitions related to the AHA! and GALE systems, and that we use throughout the thesis. These definitions are related to the main ideas behind the adaptive techniques used in these systems.

- **Concept** — is an abstract representation of an information item from the application domain, e.g. subjects to be studied in a course, or artists, art styles, or art pieces (like paintings) in a museum. It has a unique identifier and any arbitrary (Java) data structure, where part of this data identifies a resource (file) to be retrieved, adapted and presented to the user as an HTML page. Some attributes have a meaning for the system, like access (a Boolean attribute that temporarily becomes true when a page is accessed), some have meaning for the author (and user), like knowledge or interest, and some have meaning for both, like visited (determining the link color) [23]. For example, the concept called *stratum* is composed by attributes, such as, *title*, *parent*, *suitability* and *availability*, and it refers to the file *stratum.xml*. The *suitability* and *availability* are Boolean Java variables. These variables are evaluated/manipulated by the system in the moment the *stratum* concept is requested. After this evaluation/manipulation, the system retrieve, adapt and present the *stratum* concept as an HTML page.

- **Page** — is the result (or representation) of a request for a concept in the system. It is typically an HTML page.

- **Link Hiding** — is an adaptation technique that can be found in Brusilovsky's taxonomy [8]. The presentation of the links in a page is defined by an author-defined requirement. The requirements can express the common prerequisite relationships between concepts but can be used for any other condition. When a page is generated, links marked as conditional are displayed differently depending on the suitability of the link destination. Typically the system uses three link classes, named *good*, *neutral* and *bad*. The use of the name *bad* is only related to the link classes, it does not imply that it is a bad choice to follow these links. The link is shown in a color that depends on its class; the standard implementation of these colors are blue for good links, purple for neutral links and black for bad links. This realizes the link hiding technique because the color black (associated to bad links) is the same color used for the text of the web page, making the link visually indistinguishable from the text.

- **Link Annotation** — is also an adaptation technique that can be found in Brusilovsky's taxonomy [8]. It differs from link hiding because it is not "hidden" in the text. For example, a list of links in a page or in a menu is annotated through different link colors. A black link anchor is not really hidden in this context. The use of different colors for link classes that are all different from the color of the text of the web page are also considered part of the link annotation technique. Another strategy of link annotation is to associate icons to each class.

1.5 Thesis Outline

This thesis fits in the context of the evaluation of AH systems. Before we present an overview of the chapters presented in this work, we would like to remark that since this thesis is not an adaptive application, it is not possible to have adaptation techniques implemented in this printed static document. However, in Section 1.6 we would like to suggest you different reading paths depending on your background knowledge and interests to give you a sense of adaptation. In Chapter 2 we briefly present the literature of AH systems and their evaluations. After that, in Chapter 3, we present the evaluation of an adaptive course (to simplify we call it as the *Hypermedia* course) created for and served by AHA!. The Hypermedia course was offered in 2006 and our analysis was made 3 years later. The main goal of the evaluation is to study the combined effect of link structure and annotation/hiding on the navigation patterns of users. This chapter is mostly based on our paper [60] at the ACM Hypertext conference in 2010. In Chapter 4 we present an evaluation of an updated version of the Hypermedia course, but at this time, it was served by GALE in 2012. We have a description of GALE in Appendix A which is based mostly on the paper [29]. Apart from the system used there are differences between the two courses, an important one being that the layout and navigation contained an optional menu view in the updated version. This difference combined with a deadline of an exam significantly changed the findings versus what we found in the earlier analysis. The main goal of this evaluation is to identify which pages and links influence the choices of the students and to contrast this with the test logs, questionnaire answers and exam grades as well as the adaptation rules employed by the adaption engine. This chapter is based mostly on our paper [63] presented at the EC-TEL conference in 2013. After this evaluation, the main goal of the Chapter 5 is to present a continuous evaluation with the objective to reveal possible problems in the adaptive courses or in the way students are using the course while the course design can still be modified. We published about continuous evaluation in our paper [61] in the "Evaluation in e-Learning" book, in 2012. In Chapter 6 we present our findings about the evaluation of the GALE modularity and extensibility. This chapter is based mostly on our paper [62] presented at the ELearn conference in 2013. After that, we present in Chapter 7 the tools developed for the authors of an adaptive application served by GALE. The main goal of these tools is to summarize the user's navigation, tests and quiz logs to present it to the authors. With this summary, the authors can analyze the logs and draw their own conclusions. These tools so far have been used only by ourselves and have not yet been published about. In the last chapter, we present our conclusions and future work. The dataset used for the evaluation performed in this thesis can be found at `http://www.cos.ufrj.br/~vfcr/rawData.html`.

1.6 Suggestion of Reading Paths

Since this thesis is not an adaptive application, we would like to suggest different reading paths depending on your interests and background. Regardless of your interests and background, we suggest the reader to start from Chapter 1 and to end with Chapter 8. For the remaining chapters, we suggest the following reading paths.

Newbies in AH systems. All Chapters. For people with little experience in the field of AH systems and/or evaluation of AH systems, we suggest to read this thesis in the order it is written. It is important to remark that Chapter 7 are more technical. Thus, a reader with not more than a minor interest in computer science, may skip these chapters.

Evaluation. Chapters 3, 4, 5, 6, and 7. For people interested in the evaluation of adaptive courses, we suggest to read Chapters 3, 4, 7 and 5, in this order. Afterwards, you may also want to read Chapter 6 if you are interested in the evaluation of the GALE system as a generic and extensible adaptive system. A possibility we find particularly good is to read Appendix A before reading Chapter 4 or Chapter 6.

2 Evaluation of Adaptive Hypermedia Systems

The main characteristic of an adaptive system is the ability of adapting itself to the needs of the users. For information services to comfortably replace human counterparts such as museum guides [54, 69, 74], teachers [45, 27] or personal tutors and guidance [45, 37, 42], the services need to take the characteristics of individual users and of user groups into account to decide what to present, how to present it and how to structure or order the presentation. For authors to easily (or at least easier than starting from scratch) create adaptive applications, like adaptive courses, it is important to have tools and frameworks to help them [12, 18, 26, 67].

The focus of this work is on the evaluation of an adaptive hypermedia course. Thus, we are specially interested in Adaptive Hypermedia (AH) systems. An AH system is also called an adaptive web-based information system (AHS for short), which means that it is developed as hypertext or hypermedia systems.

An adaptive system consists of many layers, and each of them is responsible for a specific step in the adaptation process. The three main parts of these systems are: the *domain model*, *user model* and *adaptation model*. In Knutov's thesis [48] an elaborate comparison is made between existing AHS and applications with respect to these three aspects. In Section 2.1 we briefly summarize the findings from that comparative study, and then also list other "parts" found in different types of adaptive systems.

The developers of adaptive systems are concerned with the effectivity, acceptability, efficiency and usability of the systems. Indeed, these points are a concern in all software products where the main goal is to assist users in their tasks. However an adaptive system has to go beyond that, since the system is created to understand the needs, the characteristics and the goals of the users. Considering that the adaptation takes into account the user's actions and tasks in the system, the adaptive system and application need to be evaluated constantly to confirm that creating and using the adaptation is efficient and effective for both end-users and authors. In Section 2.2 we present the different types of evaluation of an adaptive system and some examples.

2.1 Adaptive Systems Comparison

2.1.1 Domain Model Comparison

Each adaptive application must be based on a Domain Model (DM), describing how the conceptual representation of the application domain is structured. It usually consists of *concepts* and *concept relationships*. A concept represents an abstract information item from the application domain. In all systems we investigated the concepts form a hierarchy. The leaves of the hierarchy are atomic or primitive concepts and all other nodes are composite concepts that have sub-concepts. For example in Interbook [12] a textbook is structured as a hierarchy of chapters and sections with atomic presentations, tests or examples. The pages (and sections) are connected to a structure of concepts, indicating for each page what the required (prerequisite) knowledge is for the page and which outcome concepts the page teaches something about. KBS Hyperbook [40] uses a knowledge base that consists of so-called “Knowledge Items” which are essentially concepts. Each document from the document space is indexed by some concepts from the knowledge base that describes the content representation and hierarchical structure. In APeLS [18] the concepts are encapsulated into a “Narrative” structure where each narrative can be hierarchically split into sub-narratives.

Each system proposes its own way to encapsulate content information: for example like a Pagelet (in APeLS), which contains content and a content model, or it may be an Information Unit just encapsulating content information as in KBS-Hyperbook. These Information Units are indexed to map the Knowledge Items structure. In the AHAM model [21] and in the AHA! system [26] content representation is based on pages which consist of fragments that can be conditionally included by the AH system and which represent the lowest level in the concept hierarchy. (The AHAM model considers fragments to be static but in AHA! the content of fragments can be adaptive.)

A *concept relationship* is a meaningful connection between concepts. In AHAM it is represented as an object with a unique identifier, attribute-value pairs, presentation specification and a sequence of (two or more) specifiers to indicate anchors and a possible “direction” of the connection. Each concept relationship has a type (e.g. direct link, inhibitor, 'part of' or prerequisite). Such a DM structure representation captures the types of relationships that can be encountered in most AH systems. For example, in KBS-Hyperbook one may see the dependency graph of all the KI's (knowledge items), in AHA! there are binary relationships of arbitrary types, and in APeLS there is a relationships map in a Narrative Model, by which adaptive logic is represented. In the GRAPPLE authoring tool [39] a distinction is made between concept relationships that have a meaning in the *subject domain* and relationships that have a *pedagogical meaning*. DM contains subject domain relationships (like *kind-of*, *same-as*, *special-type-of*) and the Conceptual Adaptation Model contains pedagogical relationships (like *prerequisites*). Applications in very different domains make use of very differently named relationships. The CHIP art recommender [74] uses semantic databases (represented in RDF) to connect artworks with "creators", "techniques", "subjects", etc.

2.1.2 User Model Comparison

The User Model (UM) has to be created and kept up-to-date to represent user knowledge, interests, preferences, goals and objectives, action history, type, style and other relevant properties that might be useful for adaptation. Some systems also look at the environment in which an application is used, device properties, work context, etc.

UM properties are usually separated into *domain dependent* and *domain independent* properties. The user typically has domain-independent properties like identity, name and password, all with simple (atomic) values, but UM may also have more complex properties such as a collection of groups the user belongs to, preferences, a number of learning styles, work environment, and so on. The domain-dependent properties of a UM (for a given user) typically consist of some entities, objects or concepts, for which we store a number of attribute-value pairs. For each entity there may in principle be different attributes, but in practice most entities will have the same attributes.

As domain dependent properties we see that most entities in a UM represent concepts from DM, forming an *overlay* over DM, mapping the user's domain-specific characteristics like knowledge of concepts over the domain knowledge space. There may be more domain dependent properties, such as test results and learning objectives which can be problem solving tasks or short term objectives. Typically these need to be represented in a DM as well in order to be used for adaptation. Thus, even for properties like learning goals the UM will be an overlay of DM; however, not all domain dependent properties should necessarily belong to an overlay. There can be aggregation properties like an "average knowledge" or auxiliary like "has seen an introduction page", which are difficult to express in a UM as an overlay.

KBS-Hyperbook represents knowledge through a *knowledge vector*. The values represent a "confidence" or "probability" of the user's knowledge for each concept (from DM). In Interbook (and AHA! and APeLS) the meaning of a knowledge value is *how much* the user knows about the concept. Such values can be more easily aggregated into knowledge values for higher level, composite concepts.

Despite the differences there is a striking commonality between the systems (not including AHA! or GALE): UM has a fixed structure, with predefined attributes per concept, targeting one specific application area, which in these cases is learning (education). AHA!, and inspired by that also GALE, offers no predefined UM structure. Concepts in AHA! and GALE can have arbitrary attributes. A designer can choose which attributes to define and use them depending on the type of application (s)he wishes to create.

2.1.3 Adaptation Model and Adaptation Engine Comparison

Adaptive systems adapt the presentation, the information content and the navigation structure to the user's level of knowledge, interest, navigational style, goals, objectives, etc. To this end an Adaptation Model (AM) has to be provided, indicating how concept relations in a DM affect user navigation and UM properties updates (for instance whether the system should guide the user towards or away from information about certain concepts). AM actually always consists of two different aspects: rules to translate user activities into UM updates and rules to adapt the presentation to the UM state.

For computing UM updates the most popular technique is to use "forward reasoning", meaning that an event leads to a conditional action. This is expressed by means of *event-condition-action rules*. The updates lead to more conditional actions, etc. This most closely resembled *triggers* in database systems and to some extent is also comparable to "forward chaining" in rule-based reasoning systems.

Systems using forward reasoning include Interbook, AHA! (or GALE) and to some extent also APeLS. Through forward reasoning one can calculate high-level UM properties, and have their values ready immediately when needed. KBS-Hyperbook uses deduction rules which allows for "backward reasoning", trying to deduce UM values from events that have happened or from other UM values, somewhat like how rule-based reasoning systems may use "backward chaining" to find evidence for a proposition. Other systems also use backward reasoning but only for rules that determine the adaptation of the presentation. An example of information that is typically calculated (backwards) when a link to a concept needs to be presented is the *suitability* of the link destination for the user. In designing GALE the developers tried to offer a truly generic adaptation engine, which allows to define both types of reasoning for all aspects, i.e. for updating UM as well as for adapting the presentation. Instead of "forward" and "backward" reasoning some other models like the General Ontological Model for Adaptive Web Environments GOMAWE [2] and the Generic Adaptivity Model [31] use the terms "push reasoning" versus "pull reasoning".

Just like for the structures in UM most systems only have *predefined* types of adaptation rules, specialized for their application. Interbook for instance has a rule that sets a specific amount of knowledge (1.0) to a suitable concept that is studied, and adds a small amount (0.1) to an unsuitable concept, each time that that concept is accessed. AHA! was the first system to allow authors to define arbitrary rules, thus enabling the creation of very different types of application using the same base system. As an illustration of this flexibility a student created an almost perfect simulation of Interbook in AHA! [24]. GALE was developed in such a way to extend this flexibility and extensibility further by also not prescribing the language used to define the rules.

2.1.4 Other "models" in Adaptive Systems

The GAF reference model research [48] has identified other parts of adaptive systems and frameworks that have a specific role. For instance, APeLS explicitly models learning goals or required competencies. Interbook can use concepts themselves as a learning goal to then generate a "Teach Me" page that offers direct guidance. The GAF model includes a Goal Model to represent this functionality. GALE follows the

Interbook approach of not modeling goals as a separate type of object but simply as a DM concept.

Another element in GAF is that it distinguishes between the User Model and a Group Model. Recommender systems may form user groups, consisting of people with a similar interest, and select information based on the commonalities between the group members. In general users may belong to different groups at the same time. The aforementioned adaptive systems, Interbook, KBS-Hyperbook, APeLS, and also AHA!, used mainly in education, do not accommodate a group model. GALE does support group modeling but this feature is not currently enabled so as to guarantee privacy of user data.

GAF also distinguishes between an *Application Model* (with rules that are essential for the application but not for personalization), the *Adaptation Model* (with rules for updating UM) and the *Presentation Model* (with rules for creating a personalized presentation based on UM). This follows an earlier distinction between Adaptation Model, Presentation Model and Navigation model in the General Meta-model for AH system described in [65]. Although these models deal with conceptually different aspects of an adaptive application the same rule language can be used to express these three parts. In GALE we treat all these parts as being the Adaptation Model.

2.2 Evaluation of Adaptive Systems

One of the first works about the evaluation of adaptive system dates back to 1990's. Totterdell and Boyle [70] present a few metrics to be related with different components of the logical model of an adaptive system interface. In 2001 Chin [17] said that only one third of the publications of the journal User Modeling and User-Adapted Interaction (UMUAI) for the past 9 years contained at least a small part of the evaluation of a system. However, it is important to highlight that the number of works from 1998 to 2001 with some evaluation is twice that of the five years before 1998. This is an indication on how importance of the evaluation has been increasing in the area of adaptive systems. Paramythis et al. [57] say that between 2007 and 2009 all articles published in the UMUAI, with the exception of surveys and introduction to special topics, have at least a small part of evaluation of an adaptive system. The focus of this thesis is on the evaluation of an adaptive course created and served by AHA! and GALE, based on empirical evidence and questionnaires. We also propose a continuous-time layered evaluation variant of the layered evaluation described in Section 2.2.2.

There are different techniques and methods to evaluate an adaptive system. In the next subsections we describe a few of them.

2.2.1 Comparative Evaluation

The first attempts at the evaluation of adaptive systems compared adaptive systems with non-adaptive versions of the same systems. These works, that we call *comparative evaluation*, became very popular in the 90's and in the beginning of 2000 [43, 5, 53, 75, 9, 41, 16]. It is natural that this kind of evaluation became popular in the first works of adaptive system evaluation, since the systems were created to assist the users to find their goals and solutions. Typically, a comparative evaluation is made with two groups of users navigating through the system: one group using a version of the system with adaptation, and another group using a version of the system without adaptation. At the end of the navigation, the researchers compare different things, such as the navigational paths, test results and questionnaires. The main goal of this kind of evaluation is to analyze whether the users have substantially different results (less effort, better grades, less navigation etc) between the groups. Therefore a comparative evaluation should bring some insights about the effectiveness of these systems. However, in 1990 Totterdell and Boyle [70] pointed out the potential problems that this kind of evaluation could have. Paramythis et al. [57] summarized Totterdell and Boyle's work and highlighted the following problems:

- Selection of non-adaptive controls. An adaptive system has different states for different users. The authors call *state* the current goals, navigation paths, learning style and all the characteristic that the system can store about the user (the values stored in the user model) and the adaptation that the system provides at that moment. Therefore, the biggest problem is to select one of these states as the starting point for the non-adaptive version. The question is which of these states is appropriate to compare both systems? The appropriate state should be selected from the best current practice. In this case, what should be the "best" state, since the adaptive system should have a very large space of potential system states. This is a very hard task. (Often the chosen state would be equivalent to a "final" state of the adaptive system, where all prerequisites are met after the user studied everything.)

- Selection of equilibrium points. All adaptive systems require a minimum "time" and navigation to "get to know" the user characteristics, goals etc, before the system can start to adapt the content to the user. The time required depends on the system. Consequently, the evaluation must be done after the user model has been updated, and after any other effect that influences the adaptation of the system has already taken place. Another important point to be considered is to identify the exact moments when the system reaches new points of equilibrium. The point of equilibrium is considered by the authors as the interval of time wherein the adaptation does not work while the user is acting (navigating, performing tests and tasks etc) through the system.

- Dynamics of adaptive behavior. An adaptive system can adapt itself in different ways for the same user, but this adaptation can sometimes not be beneficial for another user. In this case, the evaluator must show that that state is beneficial to that user, but the system has different "optimal" states that need to be found. In this kind of evaluation, all these states should ideally be taken into consideration.

The problems observed in the comparative evaluation motivated new proposals of evaluation research, such as the layered evaluation described in Section 2.2.2. For example, the works on comparative evaluations suggest that it does not produce generalizable results, i. e., they only work for the system or application that is being evaluated. Another point is that the results do not present sufficient data to allow the researchers to detail the system behavior effects, as pointed out by [13]. Brusilovsky et al. also observed that it is important to remark that comparative evaluations do not evaluate an exact aspect of adaptation, hence it becomes very hard to point out the causes of the "success" or "failure" of the adaptation. Specifically, it is very hard to identify under which conditions and why one aspect of adaptation can be applied to reach the goal.

2.2.2 Layered Evaluation

It is common to find in the literature two kinds of evaluation of adaptive systems: the traditional (largely called "evaluation as a whole") and the layered evaluation. The traditional evaluation consists of applying the methods over the whole system, without distinguishing between parts or layers that could be evaluated separately. Consequently, the system is treated as one block only [9, 75, 60]. For this reason, it is not possible to identify in which aspect of adaptation the possible problem is (if there is any). Brusilovsky [14] emphasizes that the evaluation "as a whole" means that the system needs to be already developed completely. Thus, the required changes that could be deduced from the results of the evaluation may not be easy to implement anymore, leading to an extra effort of development. Considering only the system layers, the traditional evaluation does not give feedback for each of these layers, and then the project pattern can not be reused in other systems. On the other hand, it is possible for the layered evaluation to identify problems and pitfalls for a specific layer of the system (the layers can be evaluated in union or concurrently). Consequently, the authors and developers of the AHS are able to focus on the resolution of the problem for that specific layer. The layered evaluation became popular in the past 12 years [56, 13, 78, 14, 58, 57].

Brusilovsky et al. [13] present the benefits of a layered evaluation in an adaptive system by revisiting a traditional and comparative earlier evaluation [9]. In [9], Brusilovsky did not have important results about the course being evaluated. For this reason, the authors in [13] decided to revisit this work using the layered evaluation to have better results. In [9], Brusilovsky realized that the adaptation did not assist the student in their learning process, because the students who used the non-adaptive system version got better results in the course than that ones who used the adaptive version. Brusilovsky et al. [13] present important results using the layered evaluation. For example, the authors observed that the students spent significantly less time on pages with status "nothing new" (a page that has neither unknown outcomes nor unknown prerequisites) than on pages with status "ready and recommended" (a page that has no unknown prerequisites and at least one unknown outcome concept). In this example the results show that the students considered the status of the page they were reading, and the system worked as idealized by the system's author.

2.2.3 Empirical Evaluation

A good way to evaluate an adaptive system using a traditional or layered technique is through empirical research. An empirical evaluation observes participants performing tasks and tests as described by Chin [17] as "the appraisal of a theory by observation in experiments." It is out of the scope of this work to present the plethora of works on empirical evaluation, and here we restrict our discussion to empirical evaluation of AHS. For the AHS the most important thing is the user, since the AHS need the characteristics, knowledge and goals of each user. For this reason, the participants of empirical evaluation can give important feedback about the system if the experiment are well formed and controlled. For example, an empirical evaluation considers the frequency with which a user accesses the system and the number of errors made by a user to complete a task. Chin contributed with his paper by presenting the common errors and problems found when researchers are creating their experiments. A common error is that researchers do not consider the knowledge required for the task, the reading skill or the visual perception. All these aspects influence the measurements and the results. There are a few other problems that are present in the experiments, for example, the nonexistence of guidelines and documentation of the experiment, inconclusive results, an insufficient number of users, leading to results without statistically relevant meaning and experiments with insufficient and adequate control. All these issues lead to bigger problems, such as the impossibility to repeat the experiment and the difficulty to obtain similar results with a different group of users.

To avoid problems in experiments, Chin suggests guidelines to assist researchers elaborating an experiment. We highlight the following suggestions: randomly assign enough participants to groups, be prepared to discard participant data if the participant requires interaction with the experimenter during the experiment and run a pilot study before the main study. The author also suggests basic measures to appear in the empirical experiments, such as the quantity, the source and the users' prior knowledge, the method and technique of the analysis and the raw data. These empirical studies were also performed in different areas of research, such as software engineering [46], medicine [81, 79] and psychiatry [52].

2.2.4 Evaluation Frameworks

From 2000 onwards the researchers and developers of AH systems started thinking of the development of frameworks, guidelines and patterns for the evaluation of AH systems [56, 58, 35, 36, 73, 57]. The main goal of these works was to eliminate the problems in the evaluation methods or the results presentation.

Gena and Weibelzahl [36] present an approach to evaluate AH systems on the Web. Their approach is based on a user-centered evaluation (UCE). They focus on the evaluation of AH system based on each phase of development: requirement phase, preliminary evaluation phase, and final evaluation phase. For each phase they present evaluation techniques and examples. At the end, the authors suggest solutions and work to be done while applying the techniques to avoid problems and pitfalls.

In 2008 Van Velsen et al.[73] present a systematic review of UCE for adaptive and adaptable systems. It is important to remark that the UCE are the methods used in the evaluation, they are not the whole evaluation as we stated before when we discussed traditional and layered evaluation. For De Jong and Schellens [30] the UCE is used to reach three main goals: verify the quality of the application, detect problems and support the decisions of implementation and design.

For Van Velsen et al. an adaptive system "tailors its output, using implicit inferences based on interaction with the user" while adaptable systems "use explicitly provided input to personalize output." The authors divided the analyzed works in four categories: concerning attitude and experience, concerning actual use, concerning system adoption and concerning system output. The concern about "actual use" is presented in more than 50% of the analyzed works. The three most used variables to evaluate the concept were usability, perceived usefulness and appropriateness of the adaptation. Besides the identification of the methods used in an UCE, the authors presented a model to be used as a framework to guide the evaluation process or the presentation of the results produced by an UCE. The most commonly used method was the questionnaire. For the authors, the questionnaire was not well formulated and could not be used to evaluate an adaptive system, which means that the quality of the evaluation was not significant and should not be used for further analysis. Some of the problems found by the authors are the nonexistence of documentation about the procedure and methods used in the evaluation (making it difficult to replicate the same experiment), experiments with the developers' opinions (little criticism about the system) and only little data about the usability of the system (which means the way the users see the system).

Paramythis et al. [57] in 2010 propose a framework to guide the layered evaluation of interactive adaptive systems (IAS). The authors consider an IAS as systems that have an interactive front-end and are capable of self-adaptation (applied to, or experienced through, the aforementioned interactive front-end), which means that it is not restricted to AH systems. The proposed framework is the union of previous approaches presented in the literature (mainly the ones which is presented in [70, 13, 76, 78, 56, 58]), and it proposes the decomposition of an IAS in five layers: collection of input data, interpretation of the collected data, modeling of the current state of the "world", deciding upon adaptation, and applying (or instantiating) adaptation. The authors say that a layered evaluation does not have to evaluate all the proposed layers, it depends on the nature of the system and the layers relevance. The authors also summarize and presents the common methods and techniques used in the layered evaluation of IAS, discussing the application of these methods and techniques clearly and presenting the best practice to a layered evaluation. The authors focused on the methods and techniques used in formative evaluations. Formative evaluation aims to identify shortcomings or errors in a system in order to further improve it and to guide the system design and development. In contrast, summative evaluation aims to determine the value or impact of a system. Therefore, the authors present the variables used in the evaluation and relate them to each layer of the system, showing practical examples of that.

Paramythis et al. [57] mention that "Formative evaluations may be more time- and labour- intensive compared to most forms of summative evaluation due to relying more on qualitative methods." Indeed, the observational methods suggested by the authors for testing the system with real users take a considerable amount of time and, in most of the case, it takes more than one test session. Another point is that the *thinking-aloud* method and the *co-discovery* method, for example, may interfere with the participants, sometimes slowing them down or, even, making them behave in different ways [17].

We consider the methods and techniques proposed in [57] to elaborate our evaluation method, which we present in Chapters 3, 5, 4 and 6. For example, we do use users to evaluate the adaptive course presented in this thesis, but in that case the evaluation concerns the impact of the system on the student's behavior instead of the system's errors (correctness of implementation).

3 Preliminary Evaluation of an Adaptive Course

The *Adaptive Hypermedia for All!* (*AHA!*) development led to GALE [66, 67]. As a consequence, GALE can be considered the AHA! version 5.0. The AHA! development started in 1996 [19], and it is considered a pioneering work in the area. It is important to remark that all the applications provided by AHA! can also be served by GALE. In this chapter, we consider the evaluation of an adaptive course provided by AHA! version 3.0[1]. This chapter is mostly based on our paper [60] at the ACM Hypertext conference in 2010.

An adaptive course is an educational application built in and provided by AHA! (and also GALE). The main goal of the evaluation is to study the combined effect of link structure and annotation/hiding on the navigation patterns of users. This goal intends to answer our first two research questions stated in Section 1.2 (and repeated here to better discuss our approach): 1. Does the annotation/hiding adaptive technique influence the choices of the students? and 2. How does the interplay between the link structure of an adaptive course and the rules employed by the adaptation influence the choices of the students? To achieve our goal we present in this chapter a exploratory case study. Easterbrook et al. [33] define an exploratory case study as "(an empirical method) used for explore initial investigations of some phenomena to derive new hypotheses and build theories". We call this case study a preliminary evaluation of an adaptive course.

We decided to go for an exploratory case study since the literature of adaptive systems did not publish related works where the combined effect of the link structure and the adaptive navigation support were considered. Nonetheless the literature dies present important results in the area of the empirical evaluation of other aspects of adaptive systems. For instance, we refer to [43] and [10] where the authors suggest that the annotation/hiding adaptive technique can speed up the navigation and learning of the students. Indeed, these works use empirical evidence to show an effect of adaptive navigation support, but they do not present evidence about the influence of the structure of links to their users' application. Weibelzahl and Weber present in [77] the limits and the advantages of using empirical evaluation in AH

[1]Note that there was an AHA! version 4.0 as a precursor to GALE but that was never released or published about.

systems. They notice that empirical evaluation can find errors and problems that could have remained undiscovered had another evaluation technique been used.

Considering the role played by the structure of links in an adaptive course, Calvi presents results about the evaluation of an online Italian language course [16]. The course has an engine to enable and disable links depending on the level of knowledge of the student. Her evaluation is focused on the comparison between two groups of students, the first consisting of the students that used the adaptation engine and the second consisting of the students that used the same course without the adaptation mechanism. Although her work does not give any statistical evidence that the adaptive engine improves the quality of learning, Calvi notices that students in the first group were able to find the material they want using a smaller number of clicks. Calvi's paper plays an important role in our evaluation since it gives an insight on how the structure of links can be related to adaptive navigation technique. It is important to note that Calvi was interested in giving evidence about the role that the adaptive engine played in the quality of students' learning, but the study of the link structure was not her goal.

Another important point about our case study is that the literature always enforces the idea that the adaptation is important for the end-users, but we are also interested in the authors' intention. Authors should not just consider the "static" link structure of the courses they create but also consider the way their students really navigate through the course, thanks to or in spite of offering adaptive navigation support.

Our case study shares the same guidelines proposed by Kitchenham et al. [46]. We first introduce the case study design in Section 3.1. This section is divided into two sections: in Section 3.1.1 we present the adaptive course by itself and its adaptation rules, and, in Section 3.1.2 we present other resources and processes involved in the study. The adaptive course plays an important role in our case study since it is considered our experimental unit, and the students are the unit of observation, as defined in [46]. Then, we present the execution of the case study and data collection in Section 3.2. In Section 3.3 we present the analysis and the results obtained in this evaluation. At the end, in Section 3.4, we present the conclusion of this evaluation.

3.1 Case Study Design

The case study design describes the products, resources and processes involved in the study, including the software being evaluated (the adaptive course), the population, the processes of intervention, and the methods used to reduce bias and determine sample size.

3.1.1 Description of the Adaptive Course

The AHA! system is heavily inspired by the AHAM reference model [21, 22]. Consequently, AHA! does not distinguish between an *Application Model* (with rules that are essential for the application but not for personalization), the *Adaptation Model* (with rules for updating UM) and the *Presentation Model* (with rules for creating a personalized presentation based on UM) as proposed in [65]. In AHA! (and also GALE) these three parts are considered the *Adaptation Model*.

An adaptive course in AHA! (and also in GALE) consists of concepts that are connected to pages (or concepts). The pages contain information (text, images, tests, and so on) and links to concepts. Links must refer to concepts instead of pages, and the link destination page may be adaptively selected by the adaptation engine. The tests are multiple choice forms, where the questions and answers can be randomized by the system (we randomize the order of the question and the answers, but we do not change the correctness of the answers and the answers are randomized for each question).

Knutov et al. [49] present a taxonomy of all the existing adaptation techniques until 2009. One of these techniques is called Link Hiding/Annotation and it is part of the navigational techniques used in AHA! (Brusilovsky [7, 8] uses the term *adaptive navigation support* in his taxonomy with a slight difference in the names of techniques: he used the term *adaptive hiding/annotation of links*). The link hiding/annotation consists of the use of different colors to present the links on the pages, perhaps with additional icons. In general, the author of the application defines rules to determine the conditions under which a presentation class is associated with a link. One of these presentation classes defines a link with the same color of the normal text, and the link is also not underlined. Consequently, the link is “hidden” in the text. The link colors suggest which links are relevant to each student. This approach is for instance discussed in [64]. In spite of other techniques provided by the AHA! system, the focus of this work is on the link hiding/annotation technique, which is part of the adaptive navigation technique shown in the taxonomy presented by [49]. For AHA! (and GALE) courses the default presentation classes are called *bad*, *good*, and *neutral*, and they have the following meaning and presentation style:

- The *bad* links point to non-recommended concepts, which means that according to the rules defined by the author, the student is expected to study something—do some reading or perform some tests—before accessing these concepts. *Bad* links are colored in black and are not underlined, which implies that they are indistinguishable from the textual information on the page. So, bad links are hidden within the text, though they are fully operational and can be clicked on at any time. An important remark about the term "bad" used here is that it refers only to the name of the link class, there is nothing inherently really *bad* about following these links.

- The *good* links point to a recommended concept that the student has not yet visited after it became recommended. *Good* links are colored in blue.

- The neutral links point to a recommended concept that the student has already visited after it became recommended. Neutral links are colored in purple.

Because of the choice of adaptation and colors the hypertext course looks like a standard (non-adaptive) website with pages and blue and purple links. Figure 1 shows a screen-shot of the *welcome* page from the hypertext course. The screen-shot shows clearly the *adaptive navigation support* technique, where *good* links are shown in blue text color (e.g., *Introduction*, *Definition of hypertext and hypermedia* and *history*), *neutral* links are shown in purple text color (e.g., *back to course topics* and *review the "readme" page*), and *bad* links are shown in black text color. Although bad links can not be distinguished from the text of the page, they appear in the list of topics in the second half of the page (e. g., the *The architecture of hypertext systems*, *navigation* and *browsing semantics on the web etc*. For more details about how AHA! performs adaptation refer to [23].

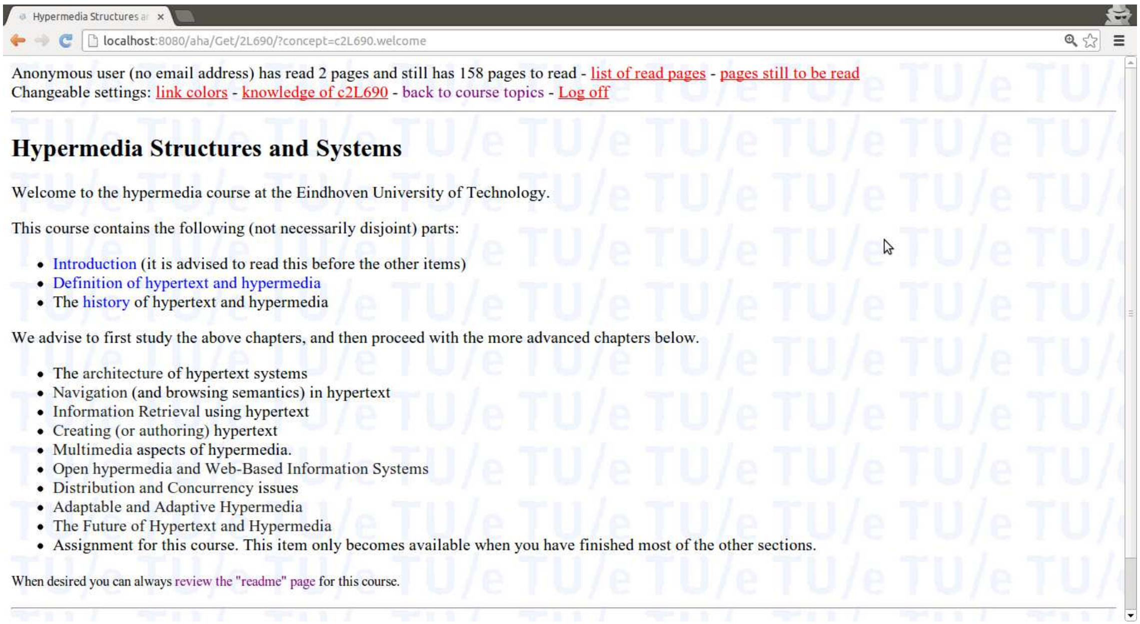


Figure 1: Screen-shot of the *welcome* page from the hypertext course, showing also *bad*, *good* and *neutral* links in the main view.

The presentation classes are used to guide the students in an AHA! adaptive course, since it shows recommended, not-recommended and neutral links. Our focus is on the students' navigation and on the course structure of links, as we present in Section 3.2.

The course is called "Hypermedia Structures and Systems" and it has been offered by the Eindhoven University of Technology since 1994. The course instance we evaluated took place during 2006 and 2007 and our analysis was made in 2009. The course was offered only through distance learning and had students from Dutch and Flemish universities. The course is composed of 171 concepts, including 11 multiple choice tests, 1 final test and the Intro concept (presented only on the very first access). From all the concepts, 12 are called the "main" concepts. We consider main concepts to be the concepts that introduce a major topic in the course; the other concepts are sub-concepts of some main concept.

The main concepts are listed in (and linked from) the course's welcome page (Figure 1 shows a screenshot of this page). The welcome page is linked to a concept and it is presented every time a student logs in. Later, when the course was moved from AHA! to GALE, students could go directly to a different concept by using the

URL of that concept. This would happen for instance when a student tries to follow a link when his session is expired. In that case, after the student logs in (again) the system redirects him to the concept he tried to access. The 2006/2007 versions of the course served by AHA! always returned to the welcome page after logging in.

The 12 main concepts have one multiple choice test each, except for the *future* concept that there is none. Initially, on the welcome page, the system presents 3 out of the 12 main concepts as good links (recommended). The first 3 recommended main concepts are *introduction*, *definition* and *history*. In the list of the recommended concepts there are 3 tests related to the 3 (already) recommended main concepts. After the student performs these first 3 tests, 90 out of 100 non-recommended links become recommended.

The student cannot repeat the tests about the main topics. The adaptation only depends on taking the test, not on the score. For the final test, the system lets the student repeat it until he scores more than 95%. The course continues with an assignment which only becomes accessible to students after they score more than 95% on the final test.

3.1.2 Other Resources and Processes of the Case Study Design

An important guideline for the design of the case study is the choice of the population from which the subjects and objects are drawn [46]. In this way, the choice of the course and, consequently, the population was made by analyzing the following requisites:

- An Adaptive course created and delivered by AHA! system and offered by TU/e.
- An online Adaptive course (no classes).
- An Adaptive course offered for undergraduate students.
- An Adaptive course offered not before the AHA! version 3.0 in 2004 [25].

Therefore, the chosen course was offered only through distance learning, with only undergraduate students from Dutch and Flemish universities, where they were registered in the "Hypermedia Structures and Systems" course offered by the

Eindhoven University of Technology (TU/e) during the year of 2006 and 2007. The evaluation was done in 2009, that is 3 years after the end of the course, which means that we did not plan the data collection specifically for this evaluation. (The system only collected data for the purpose of performing its adaptation.)

The course chosen is very representative of the population we intended to evaluate. Since 1994 the "Hypermedia Structures and Systems" course authored by Prof. De Bra [19] has been delivered over the Web, with some updates and changes every year. In the case of the hypermedia course of 2006/2007 there were no classes. All activities were done online within the course. There was no deadline. The only obligation was to complete the assignment within the course.

The course was taken only by undergraduate students from technological areas and, consequently, they all had some computing experience. Occasionally the course was also taken by graduate students (also from technological areas).

The analysis is made through the *access log* of the course. By access log we mean that the students' actions stored in a file are the navigation (clicks on links of the course) and the performed tests/assignments. Every line of the access log file (called entry) contains a reference to a user and her navigation or performed test. To avoid bias in the analysis, we exclude user entries where there is missing information about their performed tests. Specifically, it happens when there is a navigation entry for a performed test, but there is no score for that entry (when you perform a test/assignment you always get a score from 0.0 to 100, if something goes wrong, the student gets the "-" character). It means that in some cases, the logging mechanism was not working properly. Considering that the adaptive rules employed in the course are basically based on the tests, we had to disregard these entries.

Since we did not plan the data collection, we looked at the logs to find out which data we have stored in the log file. There are two important values that were not logged: the referrer, that is the information necessary to identify the link used to reach a concept or page, and the presentation class, that is a specific link color that represents recommended or non-recommended pages or concepts (for more details see Section 3.1.1). For this reason, we had to pre-process the logs to discover the referrer and the presentation class values not stored in the access log file. We explain the pre-processing of the log file in Section 3.2.

3.2 Conducting the Case Study and Data Collection

The data considered for this evaluation consists of the access log of the "Hypermedia Structures and Systems" course offered by Eindhoven University of Technology during 2006 and 2007. The evaluation was done in 2009. Therefore, the data collection in 2006 was done with no plans for an experiment. The data was stored in a file called access log. The dataset used for this evaluation can be found at `http://www.cos.ufrj.br/~vfcr/rawData.html`. The access log contains the students navigation and its performed tests/assignments, i. e., every single click a student did on a link and every performed test/assignment (multiple-choice questions) is stored in a line of the file. Every line of the access log is called a log entry. The access log is composed of the following fields:

- Student Id - an identification of the logged user.[2]
- Date - the date the student accessed a page, including hour, minutes and seconds. In other words, it is the moment of the click on a link or the moment the student performed the test/assignment.
- Concept - the concept requested.
- Session - the *login* session number provided by the server. The *login* session is the period of activity between a user logging in and logging out of a system.
- Score - the score received by the student on a test or assignment, or *null* if it is not an entry for a test or assignment.

As stated in Section 3.1.2, there are two important values that are not logged: the referrer and the presentation class. For that reason, we replayed the logs to get all information we need. The task of replaying the log was challenging. We present this task in more details in Subsection 3.2.1.

It is important to remark that no new log entries are created while we execute the process of replaying the logs. The access log contains 62,993 entries of 127 students, but we disregarded 51 users with a total of 19,886 entries for which the information

[2]The system accepts anonymous users, but in this research the few anonymous users that existed were disregarded because their test entries were missing.

regarding the test they performed was missing. We also disregarded 8,257 entries for which the "referrer" information necessary to identify the link used to reach a concept or page could not be inferred by the process of replaying the log. There are several possible causes for the "referrer" problem such as the Web browser (e. g. Opera[3]) which may reuse pages from its local cache or the students who may open a page in a new browser tab or window. In the end, we analyzed 34,850 entries of 76 students.

After replaying the log, the access log has the following fields: student id, date, concept, session, score, link class and referrer. This "new" access log file is then processed to get the measures presented in Table 1.

The first 7 rows in Table 1 are obtained by a simple counting method in the "new" access log. For example, the method to obtain the "number of accesses per link class per concept" iterates over the whole access log file grouping the concepts for each of the existing link classes. The first 7 measures are calculated analogously, i. e., by grouping fields.

The EHC measure (row 8 in Table 1) is a new measure created for this specific research. In Section 3.3.1.1 we describe the main idea behind this measure.

The last 3 rows of Table 1 are related to the time spent by the students on a concept. It is considered the *reading* time of the concept for that student, i. e., for how long did the student stay on a page before leaving that page.

With this measure we can analyze the data to verify the combined effect of link structure and annotation/hiding on the navigation patterns of students. Precisely, we have two propositions: 1) the adaptation rules, created by the author of the course, influence the navigation of the students; and 2) the rules employed by the adaptation engine contrasted with pages and links influence the choices of the students.

The first three measures are used to assess the first proposition above, since these measures are related to the link classes which is basically what we want to evaluate. The analysis of the link classes and its measures will validate our hypothesis that this adaptation technique is an effective means of adaptation and of guiding the

[3]AHA! assured that the most common Web browsers, such as Firefox and Internet Explorer, did not use their local cache, but Opera ignored these settings and continued to use its cache.

	Measure
1.	Number of accesses per link class
2.	Number of accesses per link class per concept
3.	Number of accesses per link class per concept per user
4.	Number of outgoing links for each concept, known as out-degree of a concept
5.	Number of incoming links for each concept, known as in-degree of a concept
6.	Number of accesses for each concept
7.	Number of outgoing accesses
8.	The *Empirical Hub Coefficient* (*EHC*) of a page X, which is the ratio between the number of times that students clicked on a link of X to go to a different concept and the number of times that students accessed X. (See Section 3.3.1.1 for a more detailed explanation.)
9.	The time spent by students per concept in the very first access, which consists of the very first access of a student to the concept provided he followed a link or pressed the back button
10.	The time spent by students per concept on the first time they followed a link
11.	The time spent by students per concept on the very first time with click, which is the intersection of the previous two

Table 1: Proposed measures to be used in the case study analysis.

students. If this validation failed, we would expect to obtain some insights about why it fails.

The EHC is a new measure we are proposing. This measure lies between -1 and 1, where 1 means the maximal correlation (i.e., large out-degree implies large EHC, and vice-versa) and -1 means maximal inverse correlation (i.e., large out-degree implies small EHC, and vice-versa). To assess the second proposition above, we intend to analyze the correlation between the EHC and the out-degree of a concept. We intend to relate the three last measures that are considering the reading time of a concept to the out-degree of that concept. This relationship is also used to assess the second proposition. After all these analysis, we expect to validate the hypotheses that the link structure influences the choices made by the students.

The procedure we idealized for the analysis is divided into three steps: 1. analyze the number of accesses of the link classes and its relations to concepts and users (here we can use the first three measures presented in Table 1); 2. analyze the EHC and its relationship with the out-degree of a concept (at this point we use the measures from 4 to 8 in Table 1); and 3. analyze the reading time and its relationship with the out-degree of a concept.

At the end, considering that this is an exploratory case study, even if we do not assess the hypotheses about the combined effect of link structure and annotation/hiding on the navigation patterns of students, we expect to obtain some insight on what can be changed in the research questions and hypothesis in order to run a new turn of experiments to better understand the main role of the adaptation and the link structure in the student's choices.

In Section 3.3 we present our analysis and results obtained over the measures shown in Table 1 and using the procedures presented in this section.

3.2.1 Processes for Replaying the Log

The first problem that we encountered before starting our analysis was what we called the "replaying the log" problem. This problem is related to the missing data in the access log file as presented earlier in this section. By *missing data* we mean the data that were not stored in the access log while the data collection was taking place. Our main evaluation goal is to study the combined effect of link structure and annotation/hiding on the navigation patterns of users. To start the analysis the important data is the link presentation class of each log entry, disregarding the test and assignment entries. That was the missing data. The link presentation class data indicates, as explained in Section 3.1.1, whether the student followed a recommended or non-recommended link. It is important to note that the analysis was made three years after the course ended. It also means that we did not plan this evaluation three years before, consequently we did not plan which data should be logged. The log file also missed the referrer of the page, which is the information necessary to identify the link used to reach a concept or page. If we had stored the referrer, the replaying log process would be easier. For example in this case, if a log entry stated that the referrer of *pageB* is *pageA*, then a simple access to pageA at that moment would identify the presentation class of pageB. But, unfortunately, we had no access to that information.

To tackle the "replaying the log" problem, we developed software to read the log entries and process them. The log entries are first ordered by *student ID*. The log is replayed on a different server than the one used three years earlier to ensure that the user models are properly initialized and that the replaying has no effect on the production server. The software reads each log entry and distinguishes whether it is a request for a concept or a test (answer). For each concept entry, the software processes the following steps: 1. Send an HTTP request to the server telling AHA! to get a concept and update the user model; 2. Process the HTTP answer getting all the outgoing links, saving the presentation state of each link into what we call *processed database*; for each test entry, the software sends an execution code to the server updating the user model with the test score.

The first step is important to get information about the concept, including the content and the presentation class of the links. These data enable us to process the links attributes, like the link class (bad, good or neutral), and infer the referrer of the request. That processing is made by a newly developed application. This application navigates through the log entries, and for each log entry, it gets the concept name (we refer to it as ConceptA), date and session. For the same user and session, it looks for the closest earlier entry (we refer to it as EarlierConceptB) into the processed database, gets the link class of the outgoing link pointing to ConceptA, and save EarlierConceptB as the referrer of ConceptA request into a new database called *results database*.

The processing has other rules to be followed:

1. If the session of EarlierConceptB is not the same as the session of ConceptA, it means that ConceptA is the first accessed concept in the session. In consequence of that, the referrer and the link class are *undefined*;

2. If EarlierConceptB does not have an outgoing link pointing to ConceptA, we treat the link class as *not defined*;

3. Consecutive accesses to the same concept are considered "reloading", that is a student is reloading the page;

It is important to note that in this process we are only complementing the access log with two new fields; we do not create new log entries.

3.3 Analysis and Results: a Case Study Approach

At this point we start our analysis using the measures presented in the earlier section. The procedure we idealized starts with the analysis of the first three rows in Table 1, that is: 1. number of accesses per link class; 2. number of accesses per link class per concept; and 3. number of accesses per link class per concept per user.

One guideline proposed by Kitchenham et al. [46] is to perform sensitivity analysis as they say "It is important to look first at the organization of the data, to determine whether any results might be due to outliers or data points that have an unreasonable influence." In our analysis we looked for patterns and/or outliers presented in our dataset. This led to the elimination of close to 20,000 entries coming from short (some anonymous) sessions with missing test information as described earlier. We only kept the 34,850 entries of 76 students who accessed most or all of the course and who performed the tests. Within the data that was kept we looked for patterns and investigated interesting outliers (see below).

We start our analysis by measuring the number of accesses per link class. We notice that even though links associated with the presentation class *bad* were hidden within the text, the number of accesses via bad links was substantial: roughly 6.58% (2,293/34,850) of all the links followed were *bad* links, while 42.29% (14,738/34,850) were *good* links, and 51.13% (17,819/34,850) were *neutral* links. The issue of users following non-recommended links has been observed in other systems but was never studied closely. Our goal in the sequel is to analyze this issue further in order to discover why students go against the recommendation of the author of the course and access *bad* links.

Figure 2 was generated based on the *number of accesses per link class per concept* measure. This figure shows the presentation classes used to access the 12 main concepts of the course. We consider main concepts to be the concepts that introduce a topic in the course; the other concepts are subconcepts of some main concept. The main concepts are listed in (and linked from) the course's welcome page. Since this page provides a bullet list of topics the link hiding technique does not really succeed in hiding links here, but it does provide annotation by not showing blue links for the non-recommended links.

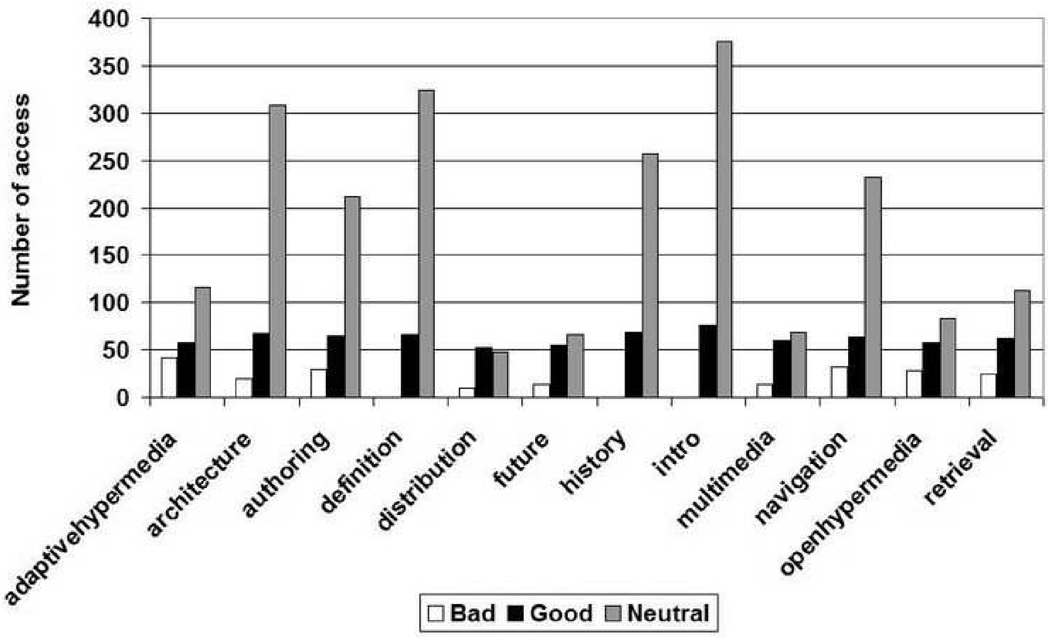


Figure 2: Total number of accesses performed by the students to each main concept of the course. The colors represent the presentation class of the link used to access the concept.

Note in Figure 2 that the concepts *definition*, *history*, and *intro* are never accessed through bad links because these links are always recommended. These concepts represent the first three main concepts or chapters of the course. Each chapter contains a multiple choice test. The "advanced" topics only become recommended after the student has completed the tests of the introductory chapters. (Only taking the tests is considered, not the score that is obtained.)

The eight "advanced" concepts shown in Figure 2 have sometimes been accessed through a bad link. The *Adaptive Hypermedia* concept has 42 accesses via bad links and 174 accesses via good or neutral links, giving roughly 20% of its accesses via bad links. This is an interesting outlier example. As we said before, we will investigate this outlier in depth. Therefore, we now focus on the bad accesses to the *Adaptive Hypermedia* concept. We start our investigation considering the referrer(s) for this concept. A new measure is proposed to treat bad accesses from a given concept: *number of accesses per link classes per concept per incoming links*, i. e., we count the number of visits for a given link class for a given concept based on the concepts used to access the given concept. For example, given the *Adaptive Hypermedia* concept, the measure presents the number of visits from all concepts used to access the given concept via bad links. One might think that most bad accesses to any concept would come from the "welcome" page where the non-recommended links are still visible in a bullet list. However, the collected data shows that less than only 12% (5/42) of the bad accesses to *Adaptive Hypermedia* came from the welcome page; hence, most bad accesses came from pages in which the link to *Adaptive Hypermedia* is effectively invisible (colored black and appearing in the main text). This suggests a behavior that we refer to as *curious browsing*: through the welcome page the students are hinted about the existence of the Adaptive Hypermedia concept, but they are not immediately drawn to it. But throughout the text, on different pages, the concept is mentioned again and again, and at some point the student's curiosity prevails and she finds out that the term is actually a hidden link and clicks on it. In addition to "being mentioned" the term Adaptive Hypermedia is named as a technology used in the hypertext course itself and students may wish to find out what that actually does.

We dwell on the idea of curious browsing a little further, and show in Figure 3 that 24 students accessed the Adaptive Hypermedia concept via bad links out of a total number of 42 accesses, which shows that some students accessed this concept more than once via bad links. This figure was generate based on the measure *number of accesses per link classes per concept per user* presented in row 3 of Table 1. We also point out that 76 students appear in the (cleaned) access log, so roughly 30% (24/76) of the students accessed Adaptive Hypermedia via bad links.

The concepts *hierarchies*, *distancematrix*, and *usermodel* can be reached from the concept Adaptive Hypermedia; so Figure 3 shows that students who follow a bad link to read about Adaptive Hypermedia do not stop at the first page but look for more hidden links to detailed information from that concept. Aside from Adaptive Hypermedia, the other concepts had roughly an average number of 1.5 accesses via bad links per student, which suggests that after revisiting the page, the students refrained from following the bad link again and waited until the link became good.

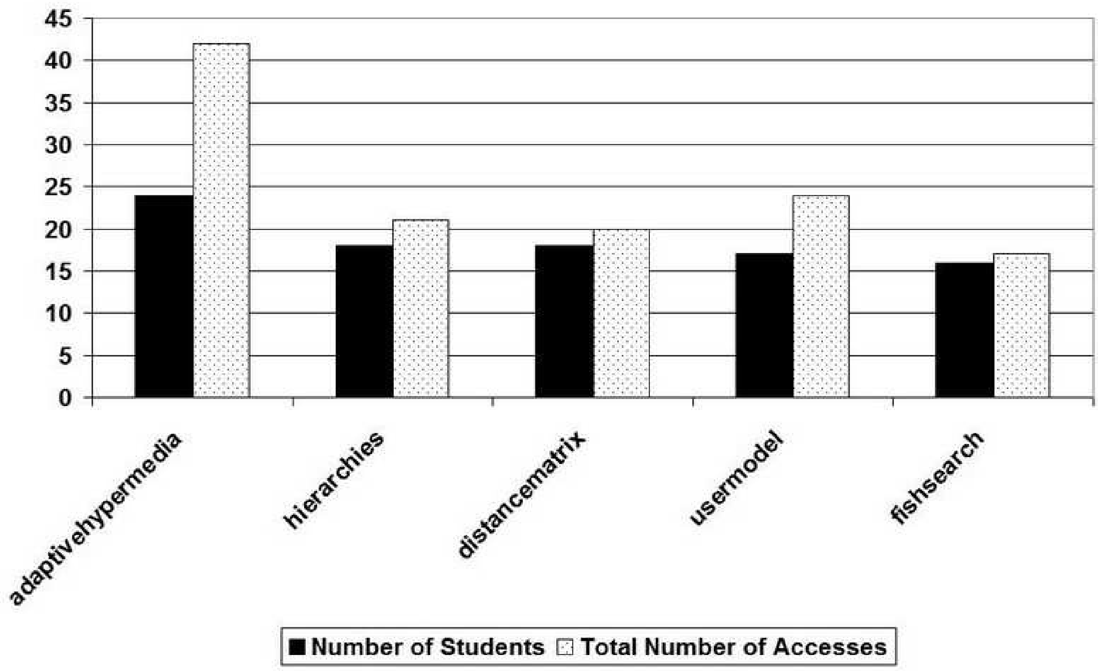


Figure 3: Number of students that accessed a concept via bad links. We show the five concepts with the largest number of student accesses via bad links.

This gives evidence that the students generally abode by the adaptation rules and that the adaptive engine of AHA! was to a large extent successful in guiding the students' navigation. In order to reinforce our suggestion that curious browsing for "Adaptive Hypermedia" was caused by the topic being mentioned in several places, we note that the in-degree (number of links pointing to the concept regardless of the link's presentation class) of the "Adaptive Hypermedia" concept is 14, which is the fifth largest in-degree in the whole course. This measure is based on row 5 of Table 1.

The observations in this section show a connection between the link adaptation and link structure: the adaptive guidance offered by the system is generally followed by the students, except for concepts that have many incoming links. Kleinberg created an important algorithm called HITS that uses ideas from random walks and measures each page according to two classes, called *hubs* and *authorities* [47]. Intuitively, good authorities are considered the interesting Web pages (i. e., Web pages that people want to refer to and that as a result have many incoming links and probably also get many visits), while good hubs are Web pages that have not just many links but many links to authorities. (The expectation is then also that hubs get many visits because they are gateways to interesting pages.) In the next section we study link structure further to find out how adaptation and link structure influence each other.

3.3.1 HUBS and Informative Pages

Earlier in this chapter, we noted that authorities may counteract the effect of link adaptation: authorities are visited even when the system recommends against them. Now let us consider the (possible) role of hubs in adaptive courses. Hubs are pages containing a large number of links (i. e., they have a large out-degree). However, in the adaptive course that uses link hiding the number of *visible* links can change. We need a new definition for hubs that takes the difference between recommended and non-recommended (invisible) links into account.

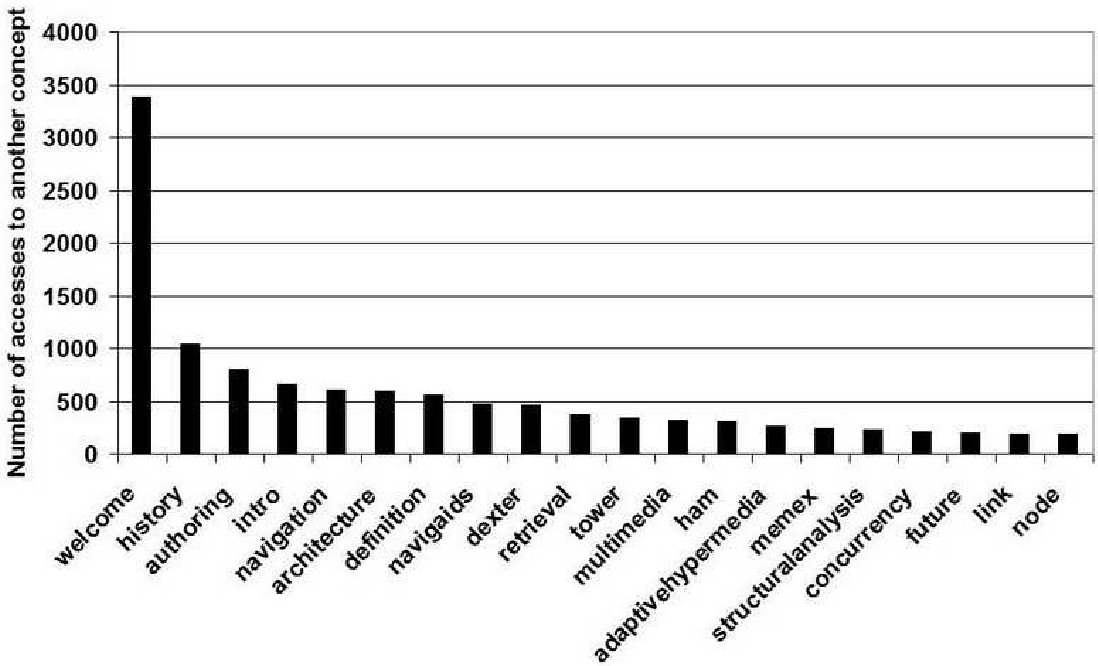


Figure 4: Concepts whose links were most frequently used to access another concept.

Figure 4 shows a rank of the concepts whose links were most frequently used to access other concepts. We remark that twelve of the concepts presented in Figure 4 have the largest out-degree in the course. This shows how a concept is used as a hub, since some concepts not only have large out-degree, but also contain links that were indeed used by the students to access another concept. This figure is based in measure *number of outgoing accesses* shown in row 7 in Table 1, i. e., the number of times a concept was used to visit another concept.

3.3.1.1 The Empirical Hub Coefficient

To investigate the idea of hubs for an adaptive course, lets define the *Empirical hub coefficient* (EHC) of a page X, which is the ratio between the number of times that students clicked on a link of X to go to a different concept (row 7 in Table 1) and the number of times that students accessed X (row 8 in Table 1). Intuitively, pages with large EHC are the ones used as hubs. Note that the EHC is a number between 0 and 1, where 1 means that each time a student visited page X she clicked on a link to another concept and 0 meaning that nobody ever clicked on a link in X that leads to a different concept (most likely because X has no links).

EHC \ Out-degree	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
0 - 0.05	19	3																
0.05 - 0.1	11	4	2		1													
0.1 - 0.15	1	8	1	2	2													
0.15 - 0.2		1	1		1	1												
0.2 - 0.25		5	5		2					1								
0.25 - 0.3		1	3		1	1		1										
0.3 - 0.35		1	2		1	1	1		1	1								
0.35 - 0.4		2	1	3		1		1										
0.4 - 0.45		3	3	1	1			1										
0.45 - 0.5			2	1	1	2	1	1	2			1						
0.5 - 0.55	1		1			2	1		1	1								
0.55 - 0.6			1	1	1	2		1	1		2							
0.6 - 0.65				1	3		1	1						1				
0.65 - 0.7				2	1	1		2	1						2			
0.7 - 0.75						1	1										1	
0.75 - 0.8										1								1
0.8 - 0.85																		
0.85 - 0.9																		
0.9 - 0.95																		
0.95 - 1																		

Table 2: Number of pages with EHC according to the row and out-degree represented by the column.

Table 2 contrasts the out-degree with the EHC of the pages of the course. The rows represent the EHC split into intervals of 0.05, and the columns represent the out-degree. Entry (i, j) of the table then corresponds to the number of pages having

EHC falling into the interval represented by row i and out-degree represented by column j.

In the hypertext course, links to a given concept could be found in many other concepts. The course is quite densely linked; links to a single concept can even be found in the subconcepts of different *main concepts* or *chapters*. This tends to decrease the EHC of the concepts since students are unlikely to revisit concepts that they have already visited while studying another chapter. On the other hand, Table 2 shows that there are many pages with out-degree between 3 and 9 and with EHC larger than 0.5. This may be another consequence of the absence of a centralized menu. In this course, when finding a link, students seemed to prefer to follow it immediately instead of waiting for the link to appear in subsequent pages. Such behavior ("let's visit the page now because we risk not finding it again later") contributes to having pages with large EHC.

	Out-Degree	Concept	Total Number Of Users	First Access		First Click		First Access with Click	
				Students	Time	Students	Time	Students	Time
1	18	history	76	59	86	68	73	57	79
2	17	authoring	76	60	89	68	88	57	98
3	15	intro	76	50	182	73	116	49	174
4	14	dexter	75	57	135	64	148	56	145
5	14	welcome	76	61	71	75	62	61	71
6	12	openhypermedia	74	52	99	57	74	46	87
7	11	definition	76	49	159	70	145	49	171
8	11	www	76	62	159	59	178	51	184
9	10	link	76	60	106	60	128	51	119
10	10	microcosm	66	48	78	19	53	17	48
11	10	navigaids	72	58	24	64	26	54	25
12	10	scripting	68	50	89	14	81	14	81
13	9	dynamicview	67	45	109	33	119	29	123
14	9	hyperties	75	59	130	55	115	49	129
15	9	intermedia	74	61	93	57	92	51	94
16	9	navigation	76	54	86	69	80	52	93
17	9	sculptural	66	46	113	40	103	37	114
18	9	wwwserver	68	54	91	48	85	43	87

Table 3: First access information for some concepts. The table shows the out-degree, the number of students who accessed each concept, and three types of first access: *first access*, which consists of the very first access of a student to the concept provided she followed a link or pressed the back button of the browser after visiting the concept, *first click*, which is the first time a student followed a link from the concept, and *first access with click*, which is the intersection of the previous two. For each type of access, the table shows the number of students and the average time in seconds they spent on the page.

A standard statistical measure that can be used to see the correlation between the out-degree of a page and its EHC is the correlation coefficient [38]. For the data shown in Table 2, the correlation coefficient is 0.71, which means a large out-degree implies large EHC.

3.3.1.2 Informative Pages

As Table 2 shows, this course has many pages with a large out-degree, which can be a problem since a myriad of links on a page may generate confusion about which link the student should follow. On the other hand, pages with large out-degree are beneficial for decreasing the depth of the link structure of the course, i. e., the minimum number of links that must be followed between any two pages that are furthest apart in terms of number of links.[4]

The hypertext course has 18 concepts with out-degree larger than 8, which is a quite large number. These concepts are easily found when presenting the results of the measure *Number of outgoing links for each concept* shown in row 4 in Table 1. We examine this issue further with Table 3, which shows concepts ordered by out-degree. Table 3 is created based on the last three measures presente in Table 1. For each concept, it shows the total number of students that accessed that concept and three types of first access: first access, first click, and first access with click. For each of the three types of access, Table 3 shows the number of students that had an access of that type to the concept, and the average time in seconds that the students spent in the Web page during such access.

Table 3 reveals an important characteristic that motivates us to define *informative pages*. We classify a page as an *informative page* if it contains information that can explicitly guide the student on the decision about which link (s)he should follow. Therefore, the presence of *informative pages* mitigates the problem inherent to pages with large out-degree. An example of an *informative page* is the *history* concept. This concept contains textual information regarding the order according to which the student should follow its links. Note that for the *first access with click*, students spent 79 seconds on the page on average, which gives evidence that the student indeed read the textual information in the page before

[4]In the terminology of [4] this would imply a high *compactness* value.

following a link. On the other hand, consider the concepts *welcome* and *navigaids*. The concept *welcome* is special because on the first visit (when just starting the course) it actually shows a "readme" page which has a lot of information and just one link, to itself. After that it becomes a page with almost no information and a large number of links. The "readme" is studied carefully, requiring 71 seconds on average. The concept *navigaids* is a typical example of a non-informative page: it has very little text, and a large number of links. There is not enough text to really guide the student into choosing a particular link. The reading time of the first access is only 25 seconds.

Now, consider the concepts *microcosm* and *scripting*. For *first access*, note that 48 students accessed *microcosm* and 50 students accessed *scripting*. However, for *first click*, only 19 and 14 students accessed *microcosm* and *scripting*, respectively. This means that most students that accessed these concepts never followed their links; they most likely clicked on the back button of the browser. We see this kind of concepts as (de-facto) dead ends. (They either have no links or only links the user already followed.)

Table 3 corroborates the idea that *informative pages* play an important role guiding the students through the maze of links that Web pages with large out-degree can create. We observe that students spent a large amount of time in their first access to the concepts, suggesting that they stopped to read the information provided in the page. *Informative pages* combine the functionality of a *hub* with a normal course page, whereas other pages that are just *hubs* but have a short reading time offer a lot of choice but no guidance. Such pages are to be avoided. The author should add information to the pages that can guide the student through the many links they offer.

3.4 Discussion and Conclusion

This chapter presented the evaluation of an adaptive course based on the exploratory case study method. The design of our case study was made through an empirical evaluation sharing the same guidelines presented in [46] and influenced by [33]. Kitchenham et al. [46] and Easterbrook et al. [33] share the same point of view: an exploratory case study is used as initial investigations and can only draw weak conclusions. We are aware of that and some of our conclusions are educated guesses based on inferences that might not be correct. A more rigorous statistical analysis of the data would be needed to come to more definite conclusions, but our findings are clear enough to form a basis for further research.

Lets first regard our hypotheses that lead us to our experiment. Our hypotheses is: 1. the adaptive navigation technique is an effective model of adaptation and of guiding students, and 2. the link structure influences the choices made by the students.

Our investigation about the adaptive navigation technique used in the course shows that 6.58% (2,293/34,850) of the concepts visits in the course were made via bad links. The most accessed concept via bad links is the *adaptive hypermedia* concept with 42 accesses, roughly 20% (42/216) of its accesses. The collected data shows that approximately 88% (37/42) of its visits came from pages in which the link to *adaptive hypermedia* concept is effectively invisible (colored black and appearing in the main text). We called this behavior as *curious browsing*, where students are hinted about the existence of a concept and they found this concept in many other concepts, i.e., the concept is mentioned again and again. At some point, the student's curiosity prevails and (s)he finds out that the term is a hidden link and clicks on it. We also measured the in-degree of the *adaptive hypermedia* concept, that is 14, which is the fifth largest in-degree in the whole course.

To corroborate the curious browsing idea, we also found evidence that some subconcepts, such as *hierarchies*, *distancematrix*, and *usermodel* concepts were also visited via bad links, specifically from the *adaptive hypermedia* concept. This is evidence that students who follow a bad link do not stop at the first page but look for hidden links to detailed information about that concept.

These educated guesses are very important to the course and to our hypothesis that says that the adaptive navigation technique is effective in guiding the student because of the following arguments: 1. the course had less than 7% of the accesses via bad links; and 2. most of the bad links accesses were made by curiosity, which seems that students were aware of the adaptivity technique. An important resource that AHA! authors can incorporate in their courses is the use of cognitive styles in their adaptation rules [68]. Thus, the system would be able to anticipate the curious browsing behavior to avoid navigation through bad links.

In our investigation about the link structure of the adaptive course we were able to show the correlation between the Empirical Hub Coefficient (EHC) and the out-degree of a concept. The EHC is an important measure originally proposed in this work. The EHC of a page X is the ratio between the number of times students clicked on a link of X to go to a different concept. For the data data shown in Table 2, the correlation coefficient between the EHC and the out-degree of a concept is 0.71, which means a large out-degree implies large EHC. This is an important finding that we should investigate in later research.

Going further in our analysis of the link structure of the adaptive course we investigated the reading time of the concepts with highest out-degrees in the course (shown in Table 3). Our analysis of the link structure and the reading time lead us to a new definition of an page with many links, that is the *informative page*. We classify a page as an *informative page* if it contains information that can explicitly guide the student on the decision about which link (s)he should follow. These pages mitigates the problem inherent to pages with too many links: disorientation. An example of an informative page is the *history* concept, where students spent 79 seconds on the page on average when they accessed the first time and clicked on a link in the page. It gives evidence that the students indeed read the textual information in the page before following a link. This evidence combined with the correlation between the EHC and the out-degree of a concept, gives evidence that the link structure influences the choices made by the students.

Considering all educated guesses and evidence to provide only a weak validation of our hypotheses, we decided to design a new turn of empirical evaluation taking into account the successes, problems and lessons learned from this evaluation, such as:

- Plan and implement the experiment design.
- Prepare the data collection to avoid missing information.
- Include a cross-validation method with questionnaires to validate our educated guesses.
- Regarding the tests are an important mechanism to the adaptation rules, include the tests in the design and analysis stages.

- Repeat the hypotheses.
- Strong validation for hypotheses.

In Chapter 4 we present the design of a new turn of evaluation for an updated version of the same adaptive course evaluated in this chapter.

4 Second Evaluation of an Adaptive Course

This chapter presents and discusses in detail the results obtained through an evaluation of a newer version of the adaptive course. The chapter is based mostly on our paper [63] presented at the EC-TEL conference in 2013. This course is served by GALE, which is an evolution of AHA! system (the platform used for the adaptive course evaluated in the first evaluation in Chapter 3). More details about GALE can be found in Appendix A and in [29] and other papers.

Three years have passed since our first evaluation presented in Chapter 3. The empirical evaluation of adaptive systems still does not have works relating the link structure of adaptive application with the adaptive navigation technique. Our motivation to this new turn of evaluation remained the same: to validate our earlier hypotheses. These hypotheses are presented in Section 1.2 and repeated here to better discuss our choices in this chapter: 1. the adaptive technique of hiding/annotating links is an effective model of adaptation and guiding students; and 2. the link structure influences the choices made by the students.

In Chapter 3 we presented an analysis of the influence of adaptation and link structure on the students' behavior (an overview can also be found in [60]). The earlier analysis was based solely on log data because we analyzed the logs after the end of the course and were therefore unable to ask the students' opinions. Obtaining meaningful information from the logs posed a challenge because the logs did not contain all data we wished to analyze. Fortunately it was possible (for almost all the access log entries) to retrieve or discover the desired data by replaying the logs for each student. In this chapter, we tackle these challenges by developing a few tools to interact with the students and by saving more data about the adaptation and the student's navigation in the logs. These implementations allowed us to analyze the logs without replaying them. We present these implementations in more detail in Chapter 7.

To understand the way users navigate through an adaptive course, our goal is to identify which pages and links influence the choices of the students and to contrast this with the test logs, questionnaire answers and exam grades as well as the adaptation rules employed by the adaption engine. Another goal is to verify how the adaptation rules created by the author of the course influence the navigation by the students.

In this chapter we present a mixed-method approach that employs data collection and analysis techniques associated with both qualitative and quantitative data [33]. There are different strategies to apply the mixed-method approach into the research. We chose the so-called *concurrent triangulation strategy*, which Easterbrook et al. [33] define as being "probably the most familiar and widely used among the mixed-methods approaches. This strategy uses different methods concurrently, in an attempt to confirm, cross-validate or corroborate findings." Our mixed-method choice is based on our results from the preliminary evaluation, such as: students seems to be aware of the adaptation technique, even when they followed non-recommended links; and, students seems to be curious about some concept that is mentioned again and again during the course, and their curiosity prevails over the adaptive navigation guidance that says that a link is not recommended. Such results would be better understood if we could ask the student about his behavior. A cross-validation approach using questionnaires and log analysis would help us understanding the student behavior and, consequently, the combined effect of link structure and annotation/hiding on the navigation patterns of users.

Our experiment shares the same guidelines proposed by Kitchenham et al. [46]. We first introduce the experiment design in Section 4.1. This section is divided into two subsections: in Section 4.1.1 we present a brief description of the revised adaptive course, and in Section 4.1.2 we present other resources and processes involved in the experiment.

In Section 4.2 we describe the execution of the experiment and data collection for this evaluation revisiting our preliminary evaluation summarized in [60]. In Section 4.3 we present the analysis and the results obtained in this evaluation, revisiting in Section 4.3.1 our study about hubs and informative pages presented in Chapter 3. At the end, in Section 4.4, we discuss the results and the conclusions obtained by our evaluation.

4.1 Experimental Design

The experimental design describes the products, resources and processes involved in the experiment, including the software being evaluated (the adaptive course), the population, the processes of intervention, and the methods used to reduce bias and determine sample size. We start our description presenting the adaptive course, in Section 4.1.1, used in this evaluation experiment. In Section 4.1.2 we present the population, processes of intervention and all methods used to reduce bias in the experiment.

4.1.1 Description of the Adaptive Course

GALE is an extensible generic and general purpose adaptive hypermedia engine [67, 66]. The GALE system is presented in detail in Appendix A. GALE was inspired and developed over the code of the AHA! system with a significant architectural difference: the separation of the adaptation engine from the user modeling service [28]. For this reason, we say that GALE is an evolution of the AHA! system. Furthermore, an adaptive course in GALE consists of concepts that are connected to pages, like in the AHA! system (for more detail refer to Section 3.1.1).

One approach used in GALE (and also in AHA!) to adapt a course is to use different colors for the links on the pages. The author of the course defines rules to determine the conditions under which a presentation class is associated with a link. This approach was used in the "Hypermedia Structures and Systems" course (to simplify we call this the *Hypermedia* course) presented in Chapter 3. Indeed, the "Design-Based Learning Hypermedia" course (to simplify we call this the *DBL* course) presented in this chapter is an update of the *Hypermedia* course. This course has existed for a long time but the content was updated several times and the course as a whole was moved from AHA! to GALE.

Moving the course from AHA! to GALE also brings more flexibility and functionality to the authors of the courses and their students. It is out of the scope of this section to describe all the important characteristics of GALE, for a concise description of GALE see Appendix A or [29]. This section aims to describe the main characteristics of GALE and the important updates made in the *Hypermedia* course presented in Chapter 3 and delivered in GALE as the *DBL* course.

From the *Hypermedia* course to the *DBL* course all the concepts were reviewed by the author during the course update. The author also added five new concepts: CSS, Empty Fragment, FOHM, Mundaneum and Ties/Hyperties. The big differences between the two courses are in the tests, logs and (possible) presentation of a navigation menu. In the earlier AHA! version each test was associated with a concept and the result of the test was a "knowledge" score for that concept. In the new *DBL course* each question of a test is linked to a concept, allowing for more detailed feedback. In the old course students were required to repeat the final test until they scored 95%, at this new course the percentage was decreased to 90%. The logs store new data, besides the date and time, the user, the concept and the score (if it is a test):

- view - indicates from which view the user followed the link (the navigation menu or the content view showing the course page);
- link class - indicates the link class of the followed link (good, neutral or bad);
- referrer - indicates the concept the student came from.

These data are logged to solve the problems we had in our earlier experiment, for more details about the problems see Chapter 3. Briefly, the *view* is logged to know whether the student follows links from the main content part or from the menu of the course. We may observe a different effect of the adaptation or different overall user behavior by knowing whether students mainly use the navigation menu to navigate through the course or whether they mainly use the links embedded in the pages (as in the old course). The *link class* and the *referrer* are logged to enable analysis of link sources and of users clicking on non-recommended links without us having to replaying the logs to retrieve that information. This last part was a big challenge in earlier evaluation, for more detail see Section 3.2.1.

In our evaluation, an important characteristic of GALE courses is the way GALE presents the colors of the links, which we call presentation classes. These classes are presented in Chapter 3, but it is very important for our evaluation that we recall them briefly here. If this thesis were adaptive, not on paper, we, based on your previous knowledge, would adapt (enable or disable) the next paragraph. Here, we intentionally show the next paragraph with the definition of the presentation classes in GALE. These classes are named “bad”, “good”, and “neutral”, and have the following meaning and presentation style:

1. The *bad* links point to non-recommended concepts, which means that according to the rules defined by the author, the student is expected to study something else—do some reading or perform some tests—before accessing these concepts. *Bad* links are colored in black and are not underlined, which implies that they are indistinguishable from the textual information of the page. So, *bad* links are hidden within the text, though they are fully operational and can be clicked on at any time. An important remark about the term “bad” used here is that it refers only to the name of the link class, there is nothing inherently *bad* about following these links.

2. The *good* links point to a recommended concept that the student has not yet visited after it became recommended. *Good* links are colored in blue.

3. The *neutral* links point to a recommended concept that the student has already visited after it became recommended. *Neutral* links are colored in purple.

Because of the choice of adaptation and colors the hypertext course looks like a standard (non-adaptive) website with pages and blue and purple links. In Figure 5 we show an example screenshot of the DBL course served by GALE (and called upon from the Sakai learning management system).

At the beginning of the course (upon the very first access, and only on that access), the student gets a page with some explanation about the course and adaptation that is used. An important message to the student in this *intro* page is that in (almost) all the cases when the student requests a page, the adaptation process is done, even when the student clicks on the *back* button of the browser. The message in this *intro* concept presents the following sentences:

The course pages are adapted each time you request them from the server. This should also happen when you use the “back” button of the browser, but some browsers (or browser configurations) may refuse to do so in which case you don’t see the adaptation. There is nothing we can do about this.

...

Links in this course are not underlined (no, there is nothing wrong with your *browser). Links appear mainly in three colors: **blue**, **purple** and **black**. Since the main text is also black there may be links you don't see. These links are black because the system thinks these links are not (sufficiently) appropriate for you at the time you view the page containing them. Don't be alarmed, when you revisit the page later these links may become blue (or purple). The link colors are determined by the server, not your browser. You should not configure your browser to override the colors requested by the server.*

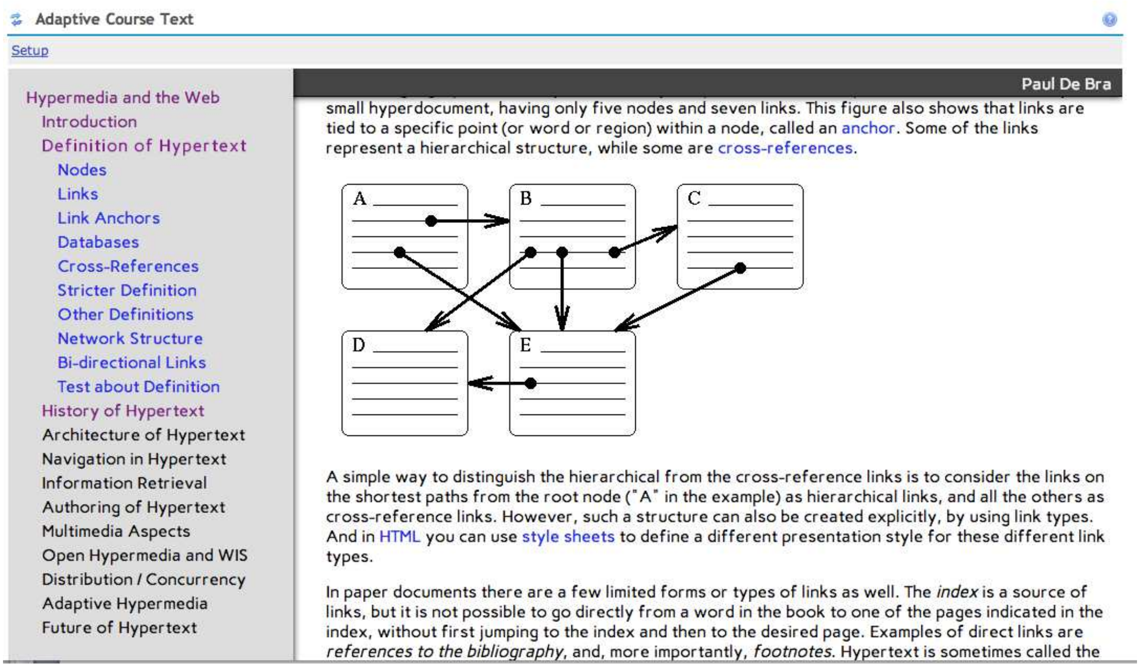


Figure 5: Screenshot of a page from the hypertext course, showing also the navigation menu (left) and a header that gives access to settings and progress information.

This message in the intro alerts the students about the importance of the link adaptation. The author suggests that students should not follow black (bad) links. The black links, as the course presents, "...are not (sufficiently) appropriate for you...". The author's intention is to tell the student to follow the structure of the course in a way that at some point they will have all the links colored in blue or purple.

The course is composed of 176 concepts, including: 11 multiple choice tests, 1 final test and the Intro concept (presented only on the very first access). It is important to remark that this course has 5 concepts more than in the course presented in Chapter 3. The content of the other 171 concepts was updated, including the multiple choice tests.

From all the concepts, 12 of them are called the main concepts. We consider main concepts to be the concepts that introduce a major topic in the course; the other concepts are sub-concepts of some main concept. Thinking of the course as a book, the main concepts would be the chapters with an introduction and the others would be the book's sections and pages. If we consider the adaptive course as a pure hypertext course, with no adaptation, the big difference between it and a book is that the student could find the same section in different chapters. Another difference is that the students do not follow a fixed page order, they follow embedded links in concepts in any way they like. In the adaptive case, the embedded links and their presentation class suggest how students should navigate through the course.

The main concepts are listed in (and linked from) the course's welcome page. The welcome page is linked to a concept and it is presented every time a student logs in, unless the student tries to access a concept with a direct URL but he is not logged in (for instance when his session is expired and he clicks on a link to a concept). In that case, after the student logs in the system redirects him to the concept he tried to access.

The students can set up the course to present a menu with the main concepts and their sub-concepts. The menu links are also adapted like the links in the pages.

The 12 main concepts have one multiple test each, except for the *future* concept. Initially, on the welcome page, the system presents 3 out of the 12 main concepts as good links (recommended). The first 3 recommended main concepts are: introduction, definition and history. At the beginning of the course, there are 74 recommended concepts (good links), 1 visited concept (the Intro concept) and 101 non-recommended concepts. In the list of the 74 recommended concepts there are 3 tests related to the 3 (already) recommended main concepts. After the student performs these first 3 tests, 91 out of 101 non-recommended links become recommended.

The welcome page also shows a sentence saying that the student has to perform all of the tests in order to be able to perform the final test. Indeed, to access the final test the student has to perform all the 11 tests in the course. It is important to remark that the final test is not just presented as a non-recommended link, it is blocked until the student performs the other tests. At the final test, the system lets the student repeat it until he scores more than 90% (instead of 95% from the earlier course).

4.1.2 Other Resources and Processes of the Experiment

At this point of the research we tried to choose an adaptive course to evaluate based on the same analysis we have made and presented in Section 3.1.2. We present our requisites for the choice of the course for this evaluation:

- An update of the adaptive course evaluated earlier.
- Offered for undergraduate students.
- Delivered by GALE.

The chosen course was offered only to undergraduate students from TU/e, where they were registered in the "Design-Based Learning Hypermedia" starting in February of 2013. The big difference between the chosen course from the earlier course is that students all started the course at the same time and they were aware that after three weeks there would be an exam about the course's content. Consequently, they had 21 days to learn about hypertext by studying the on-line course text. They were also aware that they would be allowed to consult and navigate through the course text during the exam. (In previous years students were free to take the course at their own pace and complete it with an assignment for which there was no deadline. This led to students postponing the work, concentrating on other courses with fixed deadlines and exams first.)

The data sources of the experiment are the *access log* of the course and an online available questionnaire. The access log contains students' actions stored in a file, such as the navigation (clicks on links of the course), the performed tests/assignments and the answers of a quiz about the use of bad links. The questionnaire is presented in Appendix B.

The system accepts anonymous users, but we disregard these log entries because we want to assure that all the users be consider for our analysis are registered bachelor students.

The next section presents the execution of the experiment and data collection.

4.2 Execution of the Experiment and Data Collection

The data considered for this chapter consists of the access log of 46 bachelor students of the adaptive course offered in 2013 and a questionnaire answered by 28 of the same group of students. The dataset (access log) used for this evaluation can be found at `http://www.cos.ufrj.br/~vfcr/rawData.html`. The questionnaire can be found in Appendix B. These were first year students in the programs of Web Science, Software Science and Psychology and Technology. The on-line course text was intended to prepare the students for the exam and for a later (group) assignment. The questionnaire was made available online to all the students, shortly before the exam (and left opens until a few days after the exam). The students were not required to complete the questionnaire and the system did not log their identity. The anonymous nature of the questionnaire should give students more "freedom" to answer the questions honestly. The access log is composed by the following fields:

- Student Id - an identification of the logged user.[5]
- Date - the date the student accessed a page, including hour, minutes and seconds. In other words, it is the moment of the click on a link or the moment the student performed the test/assignment.
- Concept - the concept requested.
- Session - the *login* session number provided by the server. The *login* session is the period of activity between a user logging in and logging out of a system.

[5] The system accepts anonymous users, but in this research the few anonymous users that existed were disregarded as stated in Section 4.1.2.

- Score - the score received by the student on a test or assignment, or *null* if it is not an entry for a test or assignment.

- Link class - the presentation class indicating a recommended or non-recommended concept.

- Referrer - indicates which page (can also be an external page) was used to access the concept.

- View - the name of the view from which the user followed the link.

The course's access log contains 17,318 entries, where 609 entries are answers for the tests and 1,001 entries are accesses to the test concepts (the questions). So clearly students sometimes requested a (multiple-choice) test, realized they did not yet know the answers and went back to study some course pages before trying to complete the test. Note that the tests are generated randomly from a larger collection of questions, so when a student views a test but does not answer the questions the next visit to the test may show different questions. This is also true for the final test that can be repeated: each time the student will get (some) different questions.

Table 4 presents the measures used in the analysis of the case study.

The first 6 rows in Table 4 are obtained by a simple counting method in the "new" access log. For example, the method to obtain the "number of accesses per link class per concept" iterate over the whole access log file grouping the concepts for each of the existing link class. The 7 first measures are calculated analogously, i.e., grouping fields.

The EHC measure (row 8 in Table 4) was created for the case study research presented in Chapter 3. We repeat this measure here to have a strong evidence that a large EHC implies large out-degree.

Rows 9, 10 and 11 of Table 4 are related to the time spent by the students in a concept. It is considered the *reading* time of the concept for that student, i.e., for how long time the student stayed in a page before leaving that page.

The last five rows of Table 1 were created to correlate them to the students grade in the final exam.

	Measure
1.	Number of accesses per link class
2.	Number of accesses per link class per concept
3.	Number of accesses per link class per concept per user
4.	Number of outgoing links for each concept, known as out-degree of a concept
5.	Number of incoming links for each concept, known as in-degree of a concept
6.	Number of accesses for each concept
7.	Number of outgoing accesses
8.	The *Empirical Hub Coefficient* (*EHC*) of a page X, which is the ratio between the number of times that students clicked on a link of X to go to a different concept and the number of times that students accessed X. (See Section 3.3.1.1 for a more detailed explanation.)
9.	The time spent by students per concept in the very first access, which consists of the very first access of a student to the concept provided he followed a link or pressed the back button
10.	The time spent by students per concept on the first time they followed a link
11.	The time spent by students per concept on the very first time with click, which is the intersection of the previous two
12.	Number of students who performed the first three tests (introduction, definition and history tests)
13.	Number of students who gained access to the final test
14.	Students who performed the final test
15.	Number of times a student performed the final test
16.	Highest score per test per student

Table 4: Proposed measures to be used in the case study analysis.

With this measure we can analyze the data to verify the combined effect of link structure and annotation/hiding on the navigation patterns of students. Precisely, we have two propositions: 1) the adaptation rules, created by the author of the course, influence the navigation of the students; and 2) the rules employed by the adaptation engine contrasted with pages and links influence the choices of the students.

The first three measures are used to assess the above first proposition, since these measures are related to the link classes which is basically what we want to evaluate. The analysis of the link classes and its measures will validate our hypothesis that the adaptive technique is an effective model of adaptation and students' guide. To improve the quality of this validation, we have developed a plugin in GALE (for more detail about a plugin see Appendix A called *quiz*. When a student follows more than five bad links the system presents the following question "Why are you following black links (not suitable yet)?" We expect to have a deeper understanding on the motivation of the student to follow a black (bad) link. If this validation fails, we expect to have some insights about why it fails.

The EHC is a measure that lies between -1 and 1, where 1 means the maximal correlation (i. e., large out-degree implies large EHC, and vice-versa) and -1 means maximal inverse correlation (i. e., large out-degree implies small EHC, and vice-versa). To assess the second proposition above, we intend to analyze the correlation between the EHC and the out-degree of a concept. We intend to relate the three last measures that are considering the reading time of a concept to the out-degree of that concept. This relationship is also used to assess the second proposition.

All answers to the questionnaire must be considered in the analysis, including the incomplete answers and questionnaires (students who start answering and stop in some point). The students were not required to complete the questionnaire and the system did not log their identity to give them "freedom" to answer in the way they want. The questionnaire was inserted within the Sakai Learning Management System, which allows only DBL course members to answer them. The questionnaire has basically a few questions about the adaptation and the influence of the adaptation in students choice. We would like to measure mainly three things: 1. if the students were aware of the link hiding/annotation adaptive technique; 2. if a student followed bad links and why he did that; and, 3. whether students consider the adaptation influence in their choices. At the end of the log analysis and the questionnaire analysis, a cross-validation should be made to evaluate whether what the students "do" (log analysis) is similar to what the students "say" (questionnaire).

Regarding that after three weeks of online studies the students get a multiple choice exam about the content of the course. The students grades in the exam will be correlated with his navigational logs and performed tests (analysis using the five last rows of Table 4 separately. Our expectation is to have the following correlations: high exam grades implies high test score and high exam grades implies low number of bad link accesses.

4.3 Analysis and Results: Case Study and Questionnaire

We start our analysis by relating the test log with the exam grades (or to simplify, just grade). Students were told that the tests that are part of the adaptive course test are only there in preparation for the exam and do not contribute to the course grade. The grades were directly retrieved from Sakai system. The grades that were obtained have an average of 5.2 out of a maximum of 10. (Grades were integers.) The highest grade was 7 and the lowest is 0 (only 1 student); the second lowest grade was 2 (only 2 students). 20 students obtained a grade above the average (i. e., a 6 or 7).

To understand Table 5 with the attempts and scores for the final test it is important to understand the operation of the course: only students who completed the tests embedded in each of the main concepts (chapters) were allowed (and had access) to the final test. Out of the 46 students only 12 performed all preliminary tests and gained access to the final test. (data retrieved from measure 13 in Table 4.) Of these 12 only 8 students actually attempted to complete the final test. (data retrieved from measure 14 in Table 4.) Students were told they should attempt to score at least 90% on that final test. The test could be repeated as often as desired (until the score of 90% was reached). The preliminary tests could be done only once and their score was not used for deciding on access to the final test and did not influence the adaptation. The final test was tried 319 times (by only 8 students). (data retrieved from measure 15 in Table 4.) It is interesting that 5 out of 8 students who tried the final test got an exam grade above the average. This was made by a simple match of the student grade with measure 14 of Table 4. We also note that all the students who got a score higher than 90% in the final test got a grade of 6.0 or 7.0. These findings suggest that if a student followed the course and tried to get a high score in the final test, he would be better prepared for the final exam than the "average" student. Although this is just anecdotal evidence (with few students taking and passing the final test) it is good to know because before the students started the on-line course they were told that the final test would be a good preparation for the exam. The anecdotal evidence "proves" us right. (Of course "proves" is not an appropriate term to use here as the number of samples is way too low for that, but we were at least happy that we did not find students scoring well in the final test and then failing the exam.)

Student	# Tries	Highest Score	Exam Grade
Std1	1	27	7.0
Std2	1	27	7.0
Std3	3	40	3.0
Std4	34	100	7.0
Std5	47	47	3.0
Std6	49	93	7.0
Std7	51	60	4.0
Std8	133	93	6.0

Table 5: Number of times a student answered the final test, his highest score and the exam's grade

The first three tests associated with the three recommended main concepts (introduction, definition and history) were performed by 40 students. (data obtained from measure 12 in Table 4.) Performing these tests turned the "advanced" concepts into recommended concepts, so that was a clear motivation for taking these tests. These 40 students thus clearly saw the link color changes after completing the first three tests. The questionnaire has the question: *Regarding LINK ANNOTATION, where the system presents links in different colors, perhaps with additional icons, which of the following statements applies?* For this question the answers were (in parentheses there is the number of students who chose that answer):

- I was not aware of link annotation. (1)
- I only notice that visited links became purple. (6)

- The link annotation was clearly intended to offer guidance through the course. (10)
- The link annotation was mainly intended to avoid going to some pages. (2)
- No answer. (9)

The answers show that 12 out of 19 students were influenced by the link annotation, since they recognized the importance of the link annotation to guide them through the course or to avoid them following links.

Even though students who performed the first 3 tests then got many more recommended links, they did perform some navigation through bad links before the advanced topics became recommended. The logs contain a total of 471 visits via bad links by 36 different students. This data was obtained from measure 3 of Table 4. To understand why the students would visit non-recommended concepts, the system presented a question to the students who accessed more than 5 concepts via bad links: *Why are you following black links (not suitable yet)?* The possible answers were:

1. I am curious to see what happens if I click on a black link.
2. I would like to explore the course a little bit before learning.
3. The system presented a lot of black links and I would like to know what it means.
4. I read many times about one concept and it is not suitable yet, but I would like to learn about it.
5. I do not know what a black link means.
6. Other (please specify).

15 students visited fewer than 5 concepts via bad links and did not get this question. We only wanted to question the "repeat offenders".

Table 6 presents the summary of the students' answers. It is interesting to note that 13 out of 18 answers were given on the day of the exam, including 3 of them during the exam, showing that students studied the course text just in preparation

Answer	Before Exam Day	Exam Day	**Total**
1	2	2	**4**
2	2	6	**8**
3	1	0	**1**
4	0	2	**2**
5	0	1	**1**
6	0	2	**2**
Total	**5**	**13**	**18**

Table 6: Summary of the answers about why students follow bad links.

for the exam, not really considering that the most important aspect of the course was that it prepared them for the later assignment. One student, called *Std1*, chose the *other* answer during the exam and he wrote: "Trying to find an answer to a question". Std1 had already answered this question a few hours before, and at that time he chose the answer 2. This student reset his profile after the first "exploration phase", which means that the system deleted everything about what he did from his user model, including the performed tests and visited status of concepts. (The log however remained unaffected by a reset.) We can also estimate the time students spent reading pages, by considering the time difference between two log entries for that student. Std1 showed an average reading time of roughly 9 seconds per page. This is a new measure that was not designed before. We included this measure here since it is an outlier. Clearly in the exploration he did not spend enough time on the pages to actually read and study the text. The system adaptation influenced Std1's steps in the course in spite of the fact that he was not concerned with learning the material and was only exploring the course. The adaptation in the course only depends on access, not on reading time. Because of the profile reset this student received the question about navigating through black links twice. And it turns out he gave a different answer the second time.

Table 6 also summarizes how we could groups students who do not follow the system recommendation:

- students who have curiosity about what would happen if they do not follow the adaptive structure. It is represented by students who answered 1, 3 and 4;

- students exploring the course: students who want to explore the course before they start learning. It is represented by students who answered 2;

- late students: students who start studying the course when it is (too) late. Three of them were still following bad links during the exam.

As it appears that many students started studying for the exam quite late it is interesting to look specifically at the log for the day of the exam.

It is important to notice that there are more entries in the day of the exam (4,588 entries), including the exam, than during the whole period before the day before the exam (3,976 entries). There are 8,917 log entries (51.5%) in the last day of learning. On the day of the exam the bad links access log shows 278 accesses (including during the exam), which is roughly 60% (278/471) of the all bad link accesses. All data presented in this paragraph can be inferred from the first three measures presented in Table 4. During the exam there are 168 accesses via bad links, representing roughly 36% (168/471) of all bad links accesses. These 168 accesses are concentrated in 9 different students, where 3 of them accessed only 1 bad link each and the others have an average of 27.5 (4 students above the average). Teachers often state without proof that some students go to an open book exam without any preparation, thinking they can look everything up as needed during the exam. Our log proves that some students indeed did not prepare for the exam and tried to search for the answers during the exam, and thus often going through bad links. Before the exam day the bad link accesses are mostly concentrated in the log for 5 students who performed 120 of these accesses, representing roughly 25% (120/471) of all bad links accesses. These students are in the category of student curiosity or student exploring the course according to their answer on the question about why they followed bad links.

In the questionnaire, the students answered the question: *Regarding LINK HIDING, where the main text is black, the black link is hidden in the main text but you can still click on it. Have you tried clicking on these links?* For this question, we have similar results to the one presented in Table 6: 67% of the students who admitted to have followed black links talked about the curiosity; 25% admitted to explore the course before learning; and the 8% said that they did not remember if they followed black links. (These percentages are not very precise because the absolute numbers are small.)

Another point to corroborate the idea of the students' curiosity is the fact that the first two most accessed concepts via bad links are the first two of the main non-recommended concepts appearing in the list of topics on the welcome page. (These are "The architecture of hypertext systems" (*architecture* concept) and "Navigation (and browsing semantics) in hypertext" (*navigation* concept) as shown in Figure 6.) The referrer of a visited concept indicates the concept the student came from. So we could also analyze whether other non-recommended concepts were visited directly from these concepts or whether they were accessed through cross-reference links from other main concepts. So we looked at the all pages from which bad links were followed. Again, the navigation and the architecture concepts appear in the top four rank, losing the first and second positions only to the welcome concept that lists all the main topics and the *TO DO* list (a page containing a list of all the concepts that were not visited by the student before.), two links that are always recommended. This indicates that when a student followed a bad link to a main topic he was not considered ready for, he did not stop at the first page of the non-recommended topic but kept following bad links on that topic. When asked about this behavior the answer to the bad link question was "by curiosity".

The use of link colors in the adaptive course text is described as *link hiding* because black link anchors appearing in a paragraph of black text in fact causes the link anchor to be hidden. However, the black links also appear more clearly visible in three places: on the welcome page that shows a list of links to the main concepts (Figure 6 show a screenshot of this page), in the optional navigation menu of the course (also shown in Figure 6), also showing bad links and in the TO DO list (of all unvisited concepts). In these places the differences in color should be referred to more correctly as *link annotation* instead of *link hiding*. For example, Figure 6 illustrates the presence of bad links. In the menu on the left, the items from "Architecture of Hypertext" to "Future of Hypertext" are web links in black color. Similarly, in the welcome concept (right) shown in Figure 6, the list of topics from "The architecture of hypertext systems" to "Assignment for this course" are web links in black color that can be clicked by the student. We can thus distinguish between bad links followed through annotated links versus bad links followed through hidden links. It turns out that 87% of all bad link accesses are through the three mentioned places that use *link annotation*. When we use the same three sources (welcome, menu and TO DO list) to check the accesses via good links, the percentage falls down to 57%. The accesses that remain thus indicate that the students follow many links from the course pages but few bad links in the course pages. Link hiding is thus effective in keeping students away from non-recommended topics but link annotation is not. Or in other words, students look for good links when they are navigating through the course pages but consider all the links that appear in menu-like structures.

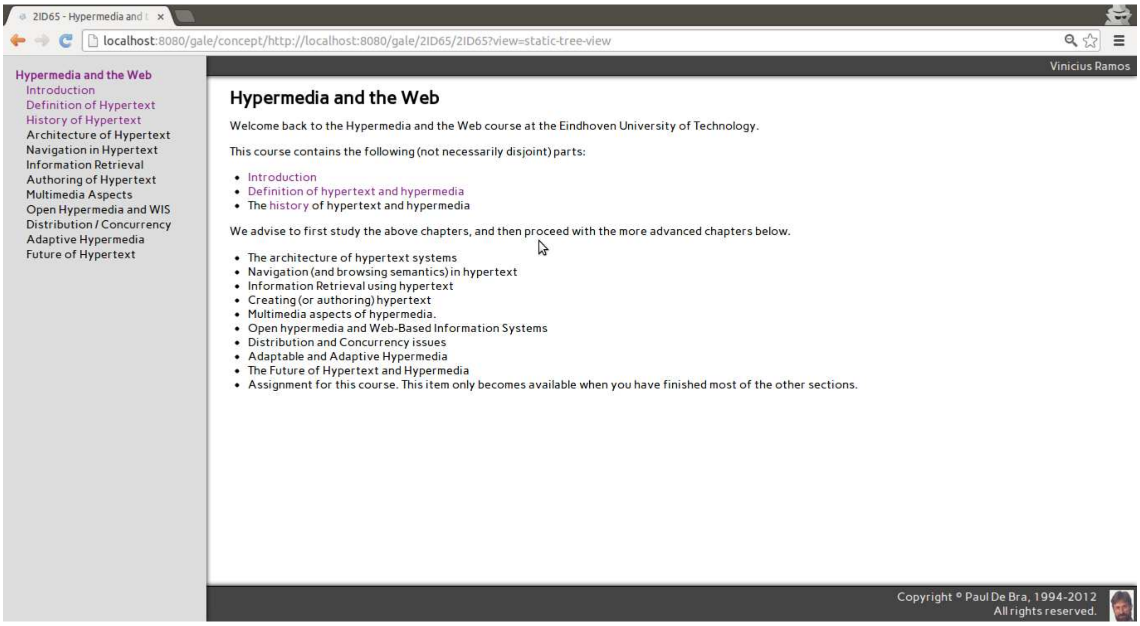


Figure 6: Screenshot of the *welcome* page from the hypertext course, showing also *bad* and *neutral* links in the menu (left) and in the main view (right).

We asked the students: *Regarding adaptation, what was your user experience?* The students answers seem to contradict the logs: 8 out of 28 students (28%) marked the option: "I hardly noticed it", while 9 others marked the option "I found it patronizing", and only one "found it helpful". 10 students did not give an answer at all. It is important to recall that roughly 90% of the students completed the first 3 tests of the course. Consequently, they got 90 new recommended concepts, of which 9 are presented in the menu and in the welcome page. For this reason, it is easy to see that the adaptation is working. Yet 28% of the students said they "hardly noticed it". Apparently link hiding and link annotation as used in this course (having links become recommended but never making a recommended link become non-recommended) is a fairly unobtrusive adaptation technique.

When students were asked to talk about their feelings about the adaptation in the course, two of them talked about the menu that could be enabled or disabled. They would like the menu to be enabled by default, whereas in the course it was disabled by default. They see an advantage of following the menu while following links, instead of relying on links that appear within the pages. Another student said that *It's not my cup of tea. I see the advantages, but I prefer an old-fashioned paper book.* Two students also suggested the possibility of having a back link/button within each page of the course, to be used instead of the browser's back button, so as to allow them to keep navigating through the pages' content.

In Section 4.4, we discuss the results presented in this chapter and the suggestions made by the students in the questionnaire.

4.3.1 HUBS and Informative Pages

In Chapter 3 we started an investigation to find out how adaptation and link structure influence each other. If this thesis were an adaptive application we would recommend (in a blue or purple link, considering a GALE application) or not (black link) the reading of Section 3.3.1 based on your previous knowledge. Another way of adapting the current section in a GALE application is to use the conditional inclusion of fragments, where the system includes (or not) part of the content depending on the user's knowledge. Since this thesis is written on paper and is not an adaptive application, we (intentionally) write (or recall, depending on what you have read before) the definitions of *Empirical Hub Coefficient* (EHC) and *Informative Pages* in this section.

4.3.1.3 The Empirical Hub Coefficient

We define the EHC of a concept X as the ratio between the number of times that students clicked on a link of X to go to a different concept and the number of times that students accessed X. Intuitively, concepts with large EHC are the ones used as hubs. Note that the EHC is a number between 0 and 1, where 1 means that each time a student visited page X he clicked on a link to another concept and 0 meaning that nobody ever clicked on a link in X that leads to a different concept (most likely because X has no links).

It is important to remark that the adaptive course evaluated in this chapter is an update of the *Hypermedia* course presented in Chapter 3. We refer to the first course evaluated as the *preliminary course*. There is a big difference in the layout of the two courses: a centralized menu (as seem in Figures 5 and 6). The menu presents a list of concepts in a tree hierarchy view, where the main concepts (chapters) are the roots of the tree and the subconcepts are within the chapters.

Table 7 contrasts the out-degree with the EHC of the concepts of the course. The rows represent the EHC split into intervals of 0.05, and the columns represent the out-degree. Entry (i, j) of the table then corresponds to the number of pages having EHC falling into the interval represented by row i and out-degree represented by column j. The out-degree of a page does not consider the menu, since it is a different view in the course. In this case, our EHC definition fits this restriction. Another important remark about the menu is that we do not log the moment the student enabled that. We are able to know which view the student came from, for example, from within the menu, and in that case we know that he has enabled the menu. Therefore, to calculate the EHC we restricted the clicks within the *content* view. For example, if the *StudentA* followed a link within the content view from *conceptD* to *conceptC* and later he followed a link within the menu view from *conceptC* to *conceptB*, we consider that *conceptC* and *conceptB* were visited but we do not consider that the *StudentA* clicked on a link of *conceptC* to navigate to *conceptB*.

In our preliminary evaluation we stated that the student's behavior "let's visit the page now because we risk not finding it again later" contributes to having pages with large EHC. At this course we have a menu, which means that the student can revisit the concept later and does not have to visit that concept at the moment he sees a link. In the *preliminary course* the correlation coefficient between the EHC and the out-degree was 0.71 (refer to Section 3.3.1.1). Thus, we would expect a decrease of the correlation coefficient in this course. Although, we still have a high correlation coefficient that is 0.65, which means a large out-degree implies large EHC.

EHC \ Out-degree	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0 - 0.05	12	16	2	2	2					1										
0.05 - 0.1	6	3	1	3	1	1				1										
0.1 - 0.15	1	6	3	1	2	2	2	2	1				1							
0.15 - 0.2	1	1	1	2	2	1	1	3												
0.2 - 0.25	1		2		2	2	3													
0.25 - 0.3	2		1	3	4	1		1	2					1						
0.3 - 0.35			1	1	2		1			1								1		
0.35 - 0.4				2		1						1								
0.4 - 0.45	1			1					1	2										
0.45 - 0.5							1	1	2					1						
0.5 - 0.55				1																
0.55 - 0.6																				
0.6 - 0.65																				
0.65 - 0.7																1				
0.7 - 0.75																				1
0.75 - 0.8													1							
0.8 - 0.85																				
0.85 - 0.9																				
0.9 - 0.95																				
0.95 - 1																				

Table 7: Number of concepts with EHC according to the row and out-degree represented by the column.

4.3.1.4 Informative Pages

The data presented in Table 8 and Table 7 reveals an important characteristic of concepts with large out-degree. The adaptive course has 19 concepts with out-degree larger than 8, which is quite large. On the one hand, a concept with a large out-degree may generate confusion about which link the student should follow. On the other hand, pages with a large out-degree are beneficial for decreasing the depth of the link structure of the course, i. e., the minimum number of links that must be followed between any two pages that are furthest apart in terms of number of links.

Table 8 shows concepts ordered by out-degree. For each concept, it shows the total number of students that accessed that concept and three types of first access: *first access* (consists of the very first access of a student to the concept provided she followed a link or pressed the back button of the browser after visiting the concept), *first click* (consist of the first time a student followed a link from the concept), and *first access with click* (it is the intersection of the previous two). For each of the three types of access, Table 8 shows the number of students who had an access of that type to the concept, and the average time in seconds that the students spent on the Web page during such access.

Kitchenham et al. [46] say that it is important to present measures of validity that could affect conclusions about the results of the experiment. We present in Table 9 the standard deviation of the data used to generate the mean measures of Table 8. Note that standard deviation of all values is quite high because there is a large variation in reading speed between students. This is likely due not only to individual differences but also to the difference between students studying carefully, ahead of time and students studying rapidly just before the exam.

	Out-Degree	Concept	Total Number Of Users	First Access		First Click		First Access with Click	
				Students	Time	Students	Time	Students	Time
1	20	history	46	28	51	44	48	27	53
2	18	authoring	40	29	36	38	33	28	38
3	16	intro	48	27	128	46	123	27	128
4	14	definition	46	24	81	46	85	22	88
5	14	dexter	34	23	111	30	104	22	116
6	13	welcome	48	19	44	48	35	19	44
7	13	openhypermedia	38	31	25	36	39	29	26
8	12	www	40	21	102	33	99	21	102
9	10	hyperties	35	17	78	24	70	17	78
10	10	microcosm	19	18	97	19	92	17	103
11	10	navigation	45	32	50	43	40	32	50
12	10	scripting	25	17	9	19	10	17	9
13	10	www-servers	35	26	81	30	71	26	81
14	9	about-html	36	18	50	29	57	18	50
15	9	intermedia	40	17	57	31	57	17	57
16	9	link	41	26	70	38	67	26	70
17	9	navig-aids	36	31	25	35	40	31	25
18	9	retrieval	39	30	76	38	68	29	78
19	9	sculptural	18	14	67	16	60	14	67

Table 8: First access information for some concepts. The table shows the out-degree, the number of students who accessed each concept, and three types of first access: *first access*, which consists of the very first access of a student to the concept provided she followed a link or pressed the back button of the browser after visiting the concept, *first click*, which is the first time a student followed a link from the concept, and *first access with click*, which is the intersection of the previous two. For each type of access, the table shows the number of students and the average time in seconds they spent on the page.

We define a page as *informative page* if it contains information that can guide the student on the decision about which link she should follow. It is important for this thesis to revisit the evaluation presented in Chapter 3. Thus, we start our analysis of Table 8 revisiting the *informative pages* we analyzed in this thesis before.

The *history* and the *intro* concepts are good example of *informative pages.* Note that for the *first access with click*, students spent 53 and 128 seconds on the page on

	Out-Degree	Concept	First Access	First Click	First Access with Click
			Std. Deviation	Std. Deviation	Std. Deviation
1	20	history	79	78	80
2	18	authoring	81	73	82
3	16	intro	165	160	165
4	14	definition	104	115	106
5	14	dexter	167	152	169
6	13	welcome	68	63	68
7	13	openhypermedia	60	91	62
8	12	www	148	156	148
9	10	hyperties	106	91	106
10	10	microcosm	153	146	156
11	10	navigation	94	82	94
12	10	scripting	20	19	20
13	10	www-servers	114	107	114
14	9	about-html	74	90	74
15	9	intermedia	103	95	103
16	9	link	112	106	112
17	9	navig-aids	38	86	38
18	9	retrieval	139	125	141
19	9	sculptural	89	82	89

Table 9: The standard deviation of the *First Access*, *First Click* and *First Access with Click* data presented in Table 8.

average, for *history* and *intro* concepts, respectively. It gives us evidence that the student indeed read the textual information in the page before following a link. On the other hand, *navig-aids* is a typical example of a non-informative pages: it has very little text, and a large number of links. The reading time of the *first access* is only 25 seconds.

The lowest average time for *first access with click* is for the *scripting* concept. As we noticed in our earlier experiment, this concept is a dead end, where all 17 students that accessed this concept, did not follow a link. We can see in Table 7 that there is one concept with EHC lower than 0.05 with out-degree equals to 10, this is the *scripting* concept. Another concept treated as a dead end is the *microcosm* concept. This concept has also a very low EHC of 0.07. Indeed, only 3 users clicked on a link from within this concept.

The *authoring* and the *openhypermedia* concepts are main chapters of the adaptive course. They were created to be *informative pages*, but, as presented in Table 7 they have EHC 0.33 and 0.13, respectively, which means that only a few students follows links from within these concepts, and most of them without even reading the concepts' content.

We reinforce the idea that *informative pages* combine the functionality of a *hub* with normal course page, whereas other pages that are just *hubs* but have a short reading time offer a lot of choice but no guidance. In this way, authors should add information to the pages that can guide the student through the many links they offer.

4.4 Discussion and Conclusion

The focus of the case study and the questionnaire have been to analyze qualitatively and quantitatively whether the adaptation mechanism influences the students' learning and navigation behavior. We approached these topics in two ways: verifying the access log of the course with the exam grades and analyzing a questionnaire answered by the students. Lets first present our hypotheses: 1. the adaptive technique is an effective model of adaptation and guiding students; and 2. the link structure influences the choices made by the students.

The main analysis of the access log relates to the visits of concepts via the so-called *bad* links (a link that points to a non-recommended concept) and whether the presence of these links affected the student's behavior. We found that students who start studying late in the course period tend to follow more bad links than students who use the entire period. On that day of the exam the bad links access log shows 278 accesses (including during the exam), which is roughly 60% (278/471) of the all bad link accesses. We also noticed in the log and questionnaire analysis that students often follow bad links (471 bad link accesses) because they are curious about the linked non-recommended concept or because they are exploring the course to get to know all the material available before they start studying the topic more in depth. The *quiz* logs shows that 7 out of 18 students answered that they were curious to know about what happen if they do not follow the adaptive structure and 8 out of 18 students answered that they would like to explore the course first before starting learning. In the questionnaire answers, 8 out of 12 students who admitted to have followed bad links talked about the curiosity of following a bad link and 3 out of 12 talked about opportunity to explore the course before learning. It clearly shows that the students were aware of the adaptation, but did not consider the advice to not follow a bad link. In one hand, we could say that students understand the way the adaptive technique recommend links, considering, thus, that this technique would influence in his choice. On the other hand, even the student knowing that is not recommended for him, he prefer exploring the course and do not consider the system's advice.

One remark is that when students navigate through bad links 87% of the clicks came from the welcome page, the menu view links or the TO DO list of concepts. On the one hand this suggests that they are curious or exploring the course before they start learning as confirmed by them in the questionnaire and by answering a specific question about following bad links during their navigation, suggests that our hypothesis that the adaptive technique is an effective model of adaptation and students' guide are valid. On the other hand it also shows that the *link annotation* used on these three pages has less effect on the students' navigation than the *link hiding* that is used for links appearing in the running text of the pages, suggesting that our hypothesis is only valid for the link hiding technique.

Considering that 51.5% (8,917/17,318) of the log entries correspond to the last day of learning and 26.5% (4,588/17,318) of the whole log belongs to the day of the exam (including during the exam), it appears that the students were much more concerned with quickly preparing themselves for the exam and were not too concerned with a comfortable learning experience for which the adaptation is intended. The log entries accumulated until two days before the exam represent 23% (3,976/17,318) of the whole log, and the bad links entries on these days represent only 15% (70/471) of the total number of bad links followed. Students who started learning in the beginning of the course period followed the course's link structure and adaptation better than of the students who started learning a few days (or even hours) before the exam. Clearly, when rushing towards the exam students are not open for advice that tells them they are not yet ready for certain topics and should study something else first. They want to take it all in, as quickly as possible and in any order. This suggests that, independently of the exam (which the adaptive course was supposed to help the student to study for that), the students need to be stimulated to start studying with the system. The absence of classes or other contact with a tutor during the course period leads to students postponing their study activity until right before the exam.

The questionnaire gave us a good insight about the needs of the students, who gave us a few suggestions to improve the quality of the course and its navigation. For example, students suggested to have a navigation menu presented at all times (by default). We made this optional because the course was on the topic of hypermedia and we wanted to stimulate navigation by following links in the pages. But for other course topics it is certainly worthwhile to offer a navigation menu by default. Some students remarked also that the course structure (i.e., the navigation support including a menu) should be more similar to the way they are used to study: it should be more similar to a book. The new paradigm of navigating through links in pages makes the navigation, as a student wrote, "quite difficult and annoying."

Students also want to know how far along they are in the course. Such a "count down" counter exists in the course but it is not permanently displayed. Instead of a menu a progress bar could also be used. As the deadline of the exam was approaching, it became clear that students lose their patience in following the advice to study some concepts before some other concepts. They seem to want to click through the whole course and the whole material quickly.

We have also carried out a structural analysis of the course, comparing the definition of *hubs* to new empirical measures, called *empirical hub coefficient* and *informative pages*. Regarding the fact that the *authoring* and the *openhypermedia* concepts have low *empirical hub coefficient* and, consequently, both concepts have low time reading. Most of the *informative pages* were used as they were intended for: it has many links, but the information can guide the students through the navigation over these links. The correlation coeficient between the EHC and the out-degree of a concept is 0.65, which means a large out-degree implies large EHC. These evidences combined with our analysis of the informative pages in the course is confirmation that the link structure influences the choices made by the students.

5 Continuous-Time Layered Evaluation in an Adaptive Hypermedia System

5.1 Introduction

In this chapter we present the continuous-time layered evaluation technique, as was proposed by us in [61]. We start explaining the layered evaluation technique, which was proposed to break the paradigm of evaluating an adaptive system as a whole [44, 77, 55, 14]. In [14], it is proposed to decompose an AH system in two distinct layers, called *user modelling* (UM) and *adaptation decision making* (ADM). The authors' proposal is based on an adaptive educational system. They define the UM layer as the evaluation of the user model of the students. For example, the UM layer is responsible for analyzing whether a student has not visited a certain important webpage, has not understood a specific material, or has not succeeded in finding the information she needs. They define the ADM layer as the evaluation of specific aspects of interaction. It uses the UM layer to adapt the AH system to the needs of the student. For example, in the hypermedia courses evaluated in this thesis, the logic of the ADM is captured into a set of *adaptation rules*. The adaptation rules determine which adaptation aspects should be selected according to the results of the UM process; i.e., these rules are responsible for the adaptive presentation, including hiding and annotation of links. The UM layer, for example, detects that a student has not succeeded completing a task, a pop-up message can be triggered by the ADM layer with additional information and links that can help the student finish the task. Figure 7 shows the layer decomposition made by the authors.

The other pillar of our layered evaluation technique is the continuous empirical evaluation introduced by Ortigosa and Carro [55]. Their goal is to evaluate adaptive courses in order to identify possible failures in order to improve or, at least, suggest possible actions to be done in these courses. They discussed situations in which an empirical analysis can help detect possible design problems leading the students to low performance (students fail to do the exercises related to the topic being studied), disorientation (students browse the course without any logical order), lack

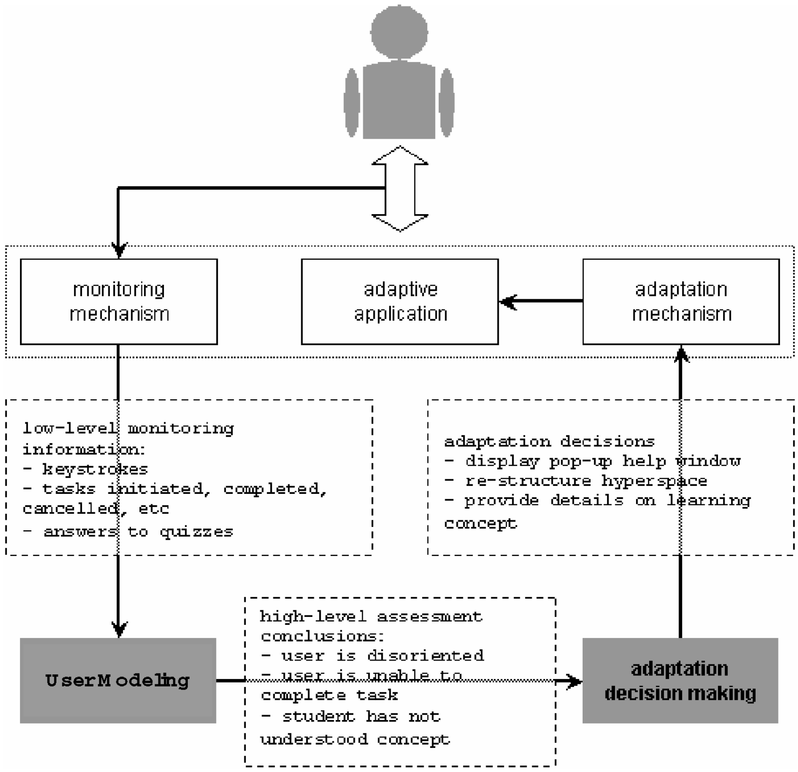


Figure 7: Adaptation decomposed (presented in [14])

of motivation (students do not browse the course frequently), and dissatisfaction (students drop out from the course shortly after the beginning). Ortigosa and Carro also give some insights about which aspects of the course design could be the cause of these problems. For example, they mention that if students obtain a low performance in some activities of the course, then it may be the case that some information is not clearly explained in some webpage or the topics of the course are not well organized. Another point raised by Ortigosa and Carro was that there may exist unidentified dependencies between topics. Usually, this type of problems are very hard to identify, but the authors remark that students may need to know some topics (called prerequisite) before studying others. Then, it would be natural to request that a previous visit to a prerequisite or a minimum score in the tests associated to them should be required in order to access a certain page. Then, if there are unidentified dependencies in a course, this could become evident if students who visited the prerequisite get better results in the current topic than those who did not.

We not only consider the aspects pointed out by Ortigosa and Carro, but we also propose new aspects for the evaluation process. In Chapter 3, we presented our findings about the evaluation of an adaptive course created and served by AHA!. For that course, we observed that students constantly return to the same *concept*, a phenomenon that we call repetitive browsing, and that students visit advanced pages without having visited more fundamental pages or doing required activities, which we call curious browsing. These two aspects represent anomalies in the way one would want a student to visit the pages of the course [60]. We also investigate how much time students spend on each concept. This turns out to be interesting empirical information regarding how effective the content of the pages is in guiding the students through the links of the course, especially on pages that contain a very large number of links, which could in principle lead the student to not know which link to follow [60].

5.2 Methodology

The combination of the layered evaluation approach by Brusilovsky et al. [14] and the continuous empirical evaluation by Ortigosa and Carro [55] forms the basis of our methodology. The methodology consists on applying the proposed techniques described in this section to the adaptive courses served by AHA! (or GALE) in order to detect situations where the course does not satisfy the student's needs or to identify possible problems in the course's design (the structure or the adaptation rules) to improve it, and depending on the faced problem, suggest updates to the authors. The proposed evaluation technique uses the idea of the layered evaluation, focusing on UM and ADM layers, to evaluate a course continuously. For example, the *hypermedia course* evaluated in this thesis and presented in Chapters 3 and 4 is under *continuous* update and it has been offered and revisited every year since 1994 [19]. Different groups of (bachelor and/or master) students navigate through this course every year in different contexts and with different goals. Consequently, we have navigational paths and performed test logs for each group of students with their different goals. For these reason, a continuous evaluation approach should be applied in this course so that students of later courses benefit from the improvement made after previous evaluations. The goal of the continuous evaluation is to reveal possible problems in the course or in the way students are using the course while the course design can still be modified. Therefore, the evaluation can be used to improve the course design in real time and provide a better learning experience to the students.

The proposed layered evaluation will encompass the following aspects: the students navigational paths (i. e., sequence of pages) that students follow during the learning process, the structure of the course, the improvement on the students' knowledge, the students' possible disorientation, and their opinion expressed in response to surveys.

1. **Navigation paths analysis**
   The navigation paths are used to represent the sequence of concepts visited in a course. Each identifier in the sequence represents a concept in the course. The main idea of analyzing the navigation path is to check if students completing a test successfully visited some page before finishing the test that students failing the test did not visit. At this point we want to find patterns among all the students who failed the test. After getting a pattern, we shall contrast it to students who pass the test. For illustration purpose, we refer to Figure 8 (adapted from [55]). In (1), we show the navigation paths for three students who failed the test **S**. In (2), we show the pattern extracted from these three students (i. e., which concepts these students visited in common). Figure 9 presents, in (3), three students who passed the test **S**. In (4), we show the pattern extracted from the students who passed the test. In order to determine why students failed the test **S**, one needs to compare the patterns for the students failing the test (Figure 8 (2)) and for the students passing the test (Figure 9 (4)). In the examples of Figures 8 and 9, one can observe that the path "2,*,5,*,7" appears only in Figure 9(4). This suggests that students who have completed the test successfully visited the concept 5 before performing the test S, while students who failed the test have not visited concept 5. This gives evidence that visiting concept 5 before performing the test S is important to the students. On the one hand, the evaluation mechanism can suggest to the author of the course to add a *prerequisite* so that students can only

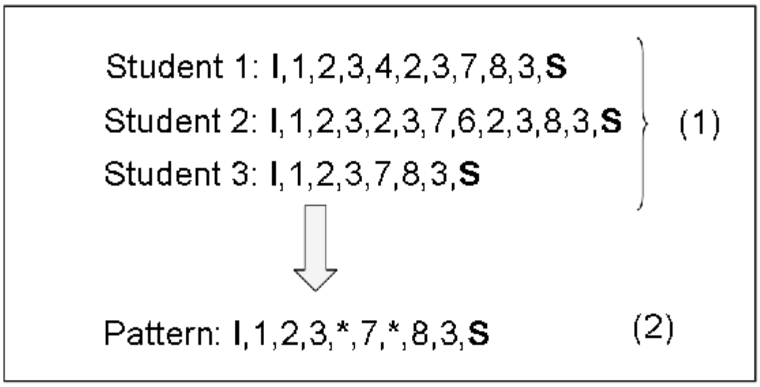


Figure 8: Navigation Path sequences from students who fail in the test S. (1) represents the students' paths. (2) represents the common pattern extracted from (1), where * stands for visits to zero or more concepts.

perform test S after visiting concept 5. Alternatively, in order to corroborate the hypothesis that concept 5 is important for test S, the author of the course may add a review with the content of concept 5 before the test S. On the other hand, the evidence may not be statistically significant, and the system should just send an warning message to the author. We would also suggest a different evaluation approach, such as observational studies or controlled experiments, to verify the evidence.

2. **Average time spent**
   The evaluation of an AH system for distance learning is always concerned about usability, student performance, or students' knowledge improvement. One point that should be evaluated is the average time spent by students on a concept. We can distinguish the three presentation classes in *AHA!*: *bad*, *good* and *neutral* links, which represent the status of a concept. We associate these presentation classes to the concept status. We consider the concept status: non-recommended, recommended but not visited, and recommended and visited. These concept statuses are associated with the link status *bad*, *good* and *neutral*, respectively. For example, a *good* link represents a concept

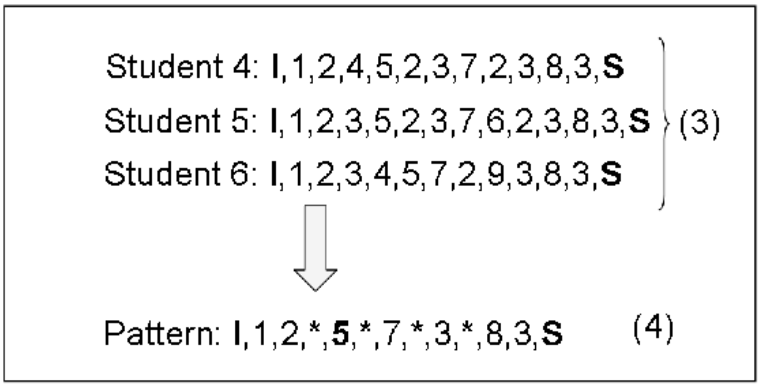


Figure 9: Navigation Paths sequence from students who succeed in the test S. (3) represents the students' paths. (4) represents the common pattern extracted from (3), where * stands for visits to zero or more concepts.

that has not yet been visited (and consequently has not yet been learned by the student). We expect that students spend a large amount of time reading a page of such a concept. On the other hand, a *bad* link represents a non-recommended concept, for which we expect that students are already missing some information or required knowledge to understand the concept. Hence, the average time spent by the students in such concepts may even be higher than for recommended concepts that were not visited. Finally, the *neutral* link represents an already learned concept. We then expect that the average time spent by the students on such concepts is lower than for the other concept statuses. It is interesting to check that the concept is already known, so that the student only needs to skim quickly through the page to find a specific piece of information of a link to be followed. Note that this idea is intrinsically related to the *informative pages* introduced in [60] and presented in details in Chapter 3.

3. **One student following *bad* links**
   For navigation through links in an hypertext course that uses the link hiding technique in the adaptation engine, it is important to evaluate the UM state

because the link hiding depends on the UM. Following *bad* links (for more information about what bad links means, see Section 3.1.1) is not prohibited and does not block the student's navigation through the course. However, a large number of *bad* links being followed by the student could indicate that the UM is not capturing the knowledge of the user correctly. Therefore, the question we would like to answer in this evaluation is: is the user's knowledge being captured correctly by the system, also considering that it is changing all the time?

Our approach to evaluate the UM is simple: once we identify that a student has been following *bad* links constantly, we present her a survey to identify her needs and discover the cause of this access via bad links. Depending on the students' answers, we might detect that the UM may not have been updated by the system in an appropriate way. The student could have clicked on a link that is shown as a *neutral* or *good* while the UM still represents it as a bad link. This would be a bug (either in the software or in the link not being made adaptive). The link could also have been *bad* without the student noticing that the link is in fact a *bad* link (and expecting the link to be there and to be recommended). In the former case, the UM status is not being updated correctly. In the latter case, the adaptation design of the course may be flawed. Both cases require the attention of the designer of the course. This topic is part of our experiments presented in Chapter 4.

4. **Different students following the same *bad* link**

   In the previous item we have discussed the case when a single student follows *bad* links constantly. The current topic follows the same spirit but has a subtle (albeit important) difference. Here we want to discuss the case where *many* students follow (the same) *bad* links. Considering that *bad* links are non-recommended concepts, a large number of students following the same non-recommended concept raises the doubt of why this link is not recommended if many students are clicking on it. Again, this could be a problem of the UM or the design of the course. On the one hand, giving another survey to the students would be sufficient to identify the needs of the students and the cause of this situation. On the other hand, it is important for the author of the course to get a feedback on which concepts students are visiting through bad links.

5. **Student motivation**
   In a distance learning course, regardless of whether it is adaptive or not, it is always hard to identify if the students are motivated. The author of an adaptive course needs to know when students loose interest; more important, the author of the course wants to known the cause of this fade of motivation and if the adaptation engine or the course design is making the student feel confused and disoriented. We want to concentrate in two types of behavior: a student who enters a course and spends a long time navigating through the course pages without ever performing the tests, and a student who visits the test pages but does not complete them. The first case can be caused by many factors. For example, the student may simply not know where the test page is or may have not noticed that there exists a test to be performed. Also, the student may not feel prepared to complete the test or may have decided to give up. In order to investigate whether a student is likely to fall into one of these categories, we propose to analyze the total time spent by the student in the whole course and in each concept of the course. If the student has spent a large amount of time navigating through the course and accessing again and again a large number of concepts in a small period of time, then probably the student is motivated to pursue the course, but got lost and does not know what to do. This suggests that the test needs to be relocated to a better place in the course or needs to be highlighted. Such a behavior is also likely to be observed in the case when the student does not feel prepared to complete the test. However, in this latter case, we expect the student to keep visiting all the concepts referring to that topic over and over again.

There are several challenges to be tackled in this chapter. The most challenging problem is the *navigation paths analysis*, because it is a way of enabling future adaptation engines to change the adaptation autonomously. In Chapter 4 we presented the findings from the evaluation of an adaptive course where students follow bad links. In Chapter 7 we present a tool (we call *Analysis Plug-in*) through which authors can analyze which concepts are accessed most via bad links. The *analysis plug-in* also presents the time spent on a concept for each student.

6 GALE Extensibility Evaluation

This chapter on evaluating the extensibility of GALE is based mostly on our paper [62] presented at the ELearn conference in 2013.

The flexibility of the AHA! system [26] was already shown through a few applications. For example, a master student at Virginia Tech used AHA! to create an adaptive alcohol abuse tutorial [32], with a look and feel that was completely different from the hypermedia course for which AHA! was originally designed. Another good example of this flexibility was used in [24] where AHA! was used to emulate the look and feel and adaptation of Interbook [12].

GALE's flexibility (of configuration in *galeconfig* and the ability to add new Java classes) was a desire of the developers of adaptive applications, because they wished not only to be able to express their own adaptation strategies and adaptation effect but also to add new functionality like adding adaptation to Virtual Reality [71], adding visualization of student progress or extending logging to perform more in-depth analysis of student behavior.

Developers can thus extend GALE functionality by extending the generic core of GALE [29]. It can be done for any specific type of application, as we presented in Appendix A. The main goal of GALE is to become a generic and extensible adaptive system that can be used by many researchers and educators without all being forced to use the same type of presentation and adaptation based on the same type of rules. For this reason, it is important to be able to reuse (or, at least, emulate the behavior of) other adaptive systems developed so far. The service extension is a good way of emulating or extending other adaptive systems. Services have been added to import adaptive applications developed for other adaptive systems. Such a new service is used to retrieve the domain model (conceptual structure) and the adaptation rules for an application, but not the content (pages) of an application. The GALE distribution includes services to accept applications defined in the CAM or Course Editor [39] developed in the GRAPPLE EU project and also applications in the format used by AHA! (version 3). Besides the GAM format, GALE also supports an XML representation of its applications: the GDOM format. There is a service to read GDOM definitions and an export plugin that can generate a GDOM representation for any application (independent of which format was used to import the application). It is possible to develop compilers (to use in a service) for formats like Interbook, Scorm, and others, but this has not yet been done until now.

The main goal of the evaluation is to assess the extensibility and flexibility of GALE in the sense of serving different kinds of adaptive applications. This goal intends to answer our last research question stated in Section 1.2 (and repeated here to better discuss our approach): Can GALE system effectively serve different kinds of adaptive applications to reach its goal as a generic and extensible adaptive system? To achieve our goal we present in this chapter a survey research approach. Easterbrook et al. [33] define a survey research as a method where the data analysis techniques are used to generalize from a representative sample to a well-defined population.

Considering our research question, one can think that a simple measure of the number of different applications served by GALE would be enough. In one hand, for a basic answer of this research question, we can say that a simple measure is sufficient to answer that. In this case, the main objective is to find ways and measures that can be used to distinguish one application from the other. There are numerous ways to check whether an application is different from another one. We are interested in the adaptive techniques used, its context and the rules (and relationships) implemented to adapt the application. For example, an adaptive course that uses the link hiding/annotation technique is different from an adaptive course that uses a view to present the student's progress. They are also both different from an adaptive course that identify the student's cognitive style to present its content. On the other hand, the GALE extensibility is not only a matter of serving different applications, it is also a matter of how difficult is to create different applications and how difficult is to extend GALE. In this case, the survey research also intends to understand these difficulties.

An application in GALE is a complete program served by GALE, while an extension is part of the GALE system and its applications, such as processors, plug-ins, services, modules and views [67, 66, 29]. These extensions are included as part of an application.

Our evaluation approach presented in this chapter is based on a questionnaire (see Appendix C). We first introduce the experimental design in Section 6.1. Then, we present the execution of the case study and data collection in Section 6.2.

In Section 6.3 we analyze the applications and extensions developed and present the results for this evaluation. Section 6.4 discusses the results obtained by the qualitative evaluation of the extensibility and modularity of GALE. A brief overview of this chapter can also be found in [62].

6.1 Experimental Design

The survey design describes essentially the population being studied and clearly identifies the research questions that ask about the nature of that particular population. The methods used to reduce bias and determine the sample size is also an important topic in this section.

We start our design choosing the population and the sample size we will analyze. In this way, the following requisites were analyzed to define the sample size and the population involved in this evaluation:

- Graduate students of computer science and correlated areas.
- Limited time to create and implement applications and extensions.

The requisite of graduate students of computer science and correlated areas were included for a few reasons. The first reason is that the GALE system was developed over the past 20 years within the university (TU/e). In this way we chose to use graduate students. Most of its applications are adaptive learning environments (courses, tutorials, thesis, etc). To develop extensions to GALE requires a basic knowledge of programming languages, specially the Java language. Therefore, the computer science and correlated areas (technological areas) is a requisite. At last, graduate students are the next generation of software professionals (and most of them already work in the area) and, so, are relatively close to the population of interest [46].

The limited time of development is a requisite because most of graduate students already have other tasks/activities to do in both environments the academic and/or their jobs. Indeed, these requirements give the opportunity to the researchers to propose an assignment for the "Adaptive Systems" course offered at TU/e to master students, where the students have to create adaptive applications and implement extensions to GALE.

The course chosen consists of the course called “Adaptive Systems” offered in 2012 to master students. The students were in the master programs of Computer Science and Engineering, Business Information Systems, Embedded Systems and Human-Technology Interaction. The analysis is made trough a questionnaire with multiple-choice and open questions presented in Appendix C. The questionnaire contains questions about the experience of the students using, authoring and developing in/for GALE.

The reliability and validity of qualitative research is a well-known problem in the literature. For instance, LeCompte and Goetz present in [51] these problems in ethnographic research. While reliability is concerned with the replication of the research in order to have the same scientific results, the validity is concerned with the accuracy of these results. Kitchenham et al. [46] present a guideline to avoid (or at least minimize) bias from researcher's opinion over the study: “If you cannot avoid evaluating your own work, then make explicit any vested interests (including your sources of support) and report what you have done to minimize bias.” The main interest in the validity of our hypothesis would rely on one of the supervisors of this thesis, since he is the mentor, but not the developer, of the GALE system. To minimize this bias, beside the author of this thesis, this evaluation is also assessed by the other supervisor of this thesis and a doctor who is scientific researcher and programmer at the TU/e.

The questionnaire was made available online to all students (42 students in total), shortly before their project presentation (and left open for a few days). The students were not required to complete the questionnaire and the system did not log their identity. The anonymous nature of the questionnaire was intended to give students more “freedom” to answer the questions honestly and also to try to minimize the bias, because when students are answering questions, they bend the truth to what they think the researcher want to hear or what is socially acceptable.

6.2 Conducting the Experiment and Data Collection

The data considered for this chapter consists of the questionnaire presented in Appendix C answered by 16 master students enrolled in the course called “Adaptive Systems” offered in 2012 at TU/e. The students were in the master programs of Computer Science and Engineering, Business Information Systems, Embedded Systems and Human-Technology Interaction. The students had 4 weeks of classes, 20 hours in total. The classes intended to prepare the students to a later (group) assignment (as well as for an exam, given in week 5). In the course students learned about adaptive Web-based systems and adaptive technology-enhanced learning in general as well as about the specifics of GALE as a generic platform to create adaptive applications.

All students were required to use GALE and not allowed to use anything else instead. The assignment was composed of the creation of an adaptive application, using any content the students desired, and the development of a new GALE extension in order to see whether the claimed extensibility of GALE is actually achievable and usable in practice. Students would fail in the course if they did not do these two things. Students did have a choice between two authoring schemes for GALE: using a textual language or using a graphical authoring interface. (We recommended the former but did not punish students who preferred the latter.) To give students more confidence and let them separate the content of the course from the research study being carried out during the course for this thesis, the author of this thesis introduced himself to the class and explained the research that was being carried out. Students were also suggested to contact the teachers or the author of this thesis to assist them in any trouble or doubt they could have about the research or the development.

Students had six weeks to develop their application and extension and to then present their work (project). The groups for the assignment were divided by the students themselves. There were only two conditions for the group to assign up to 4 students and to have at least one computer science student.

Beside the classes, all students were given the Milkyway application [59], the GRAPPLE (and GALE) tutorial and the adaptive PhD thesis by David Smits [66] as examples. An application in GALE is a complete program served by GALE, while an extension is part of the GALE system and its applications, such as processors, plug-ins, services, modules and views [67, 66, 29]. These extensions are included (or it is better to say used) as part of an application.

6.3 Analysis and Results: A Questionnaire Approach

We start our analysis and results showing the subject of the projects presented by 11 groups of students. These groups worked on different adaptive applications. The applications contains, obviously, the content they chose freely and the extensions they developed. The subjects the students chose for developing an application (and extensions) are the following:

1. A tutorial/course on C-programming;
2. An introduction into Currency Trading;
3. A tutorial on Do-It-Yourself Tools;
4. A course on Elementary Electro Physics;
5. An adaptive selector for Food Recipes;
6. A Latex tutorial;
7. A tutorial about Linux Commands;
8. An adaptive Radio Player;
9. An introduction into Relativity Theory;
10. An adaptive version of a (commercial) Research Methods Book
11. An introduction into the facilities of a Sports Center.

All of these applications (except for the radio player) used adaptive navigation support through *prerequisite* relationships. Other features used by the students in some projects were *conditional inclusion of fragments* (where pieces of content are included when certain condition is met. For details refer to [20]), *stretchtext* (where the reader can expand or contracts the content in that place. For details refer to [20]), template pages (to ensure consistent layout. For details refer to [67]), content from wiki sites, and specific handling of form input (e.g. radio player). Contrary to what one might expect, students preferred to use the GAM language over the graphical authoring tools that were developed in the GRAPPLE project

[39]. This may be explained by the type of students (each group contained at least one computer science student) who value full control over ease of use of the authoring interface. The applications presented by the students were of a high quality, which means that in a few weeks they were able to have an adaptive application based on existing content, with fully functional adaptation. By the term *high quality* we mean the students developed an application with concepts, concepts relationships, different adaptive techniques (sometimes the same application have different adaptive techniques), original content and fully functional adaptation. Based on this assertion, we present in Section 6.3.1 our questionnaire analysis and results about the authoring experience in GALE, and in Section 6.3.2 we show the analysis and results about the development experience in GALE.

6.3.1 Authoring in GALE

We start our questionnaire analysis by asking about the authoring experience in GALE.

The questionnaire was answered by 16 students out of 42. The youngest one was 22 years old and the oldest one was 28 years old. The average age was roughly 24 years. 14 out of 16 already had experience in Java language programming (required to make extensions to GALE). Because of the small number of respondents our findings are somewhat anecdotal. No statistical analysis was performed because the small numbers are unlikely to yield strong statistical evidence of any finding.

We start our quantitative analysis with the students answers for the question "how difficult was the authoring as a whole, on a scale of 1 to 5?". (So 1 means very easy and 5 means very difficult.) 14 students answered, with an average score of 3.5. Only 20% of the students considered authoring to be easy (and none very easy), which corresponds to earlier findings in the GRAPPLE project (that were worse even though in the project a supposedly more user-friendly graphical authoring tool was used). Before analyzing the open comments (qualitative approach) about the whole authoring process, we analyze some aspects of authoring in GALE more in depth to understand what is easy and what makes authoring difficult.

Students were also asked about how difficult the authoring process was in terms of specific aspects of the system. Table 10 shows the answers of the students, where level 1 is the lowest level (very easy) and 5 is the highest level (very difficult). The numbers presented in the table indicate the number of students who chose that level, with percentage in brackets.

Aspects	**Levels**					
	1	**2**	**3**	**4**	**5**	**No Ans.**
Concepts	3 (18%)	5 (31%)	3 (18%)	3 (18%)	0 (0%)	2 (12%)
Concepts and Relationships	6 (37%)	1 (6%)	5 (31%)	2 (12%)	0 (0%)	2 (12%)
Adaptation Rules	0 (0%)	4 (25%)	7 (43%)	2 (12%)	1 (6%)	2 (12%)
Installation and Deployment	0 (0%)	5 (31%)	1 (6%)	4 (25%)	4 (25%)	2 (12%)
Test (quiz)	2 (12%)	2 (12%)	7 (43%)	2 (12%)	0 (0%)	2 (12%)

Table 10: Number of answers about the aspects of authoring regarding the ease of creation in a level from 1 (very easy) to 5 (very difficult).

It is interesting to note that 50% of the students considered the *Installation and Deployment* aspects the most difficult part of the authoring process (marked as level 4 or 5) and none of them considered the aspect of creation (of *concepts* and *relationships*) to be very difficult. The aspect of authoring (and development) is one topic that needs to be taken care of in future work. One important remark is that the teachers of the course received, during the six weeks of the project, only one email about the *installation and deployment* topic, and they were visited by 2 students, on different days, for personal help on installation and deployment of GALE system. Whereas all groups presented their project, it seems that the students found the installation difficult but did manage it without help from the teachers. Indeed, the process of installation and deployment needs to be better documented and explained.

Regarding ease of use, we highlight two interesting comments by the students. One student said that "We spent more time trying to understand how to implement things and debugging than actually doing relevant work." Another one said "The setup and usage documentation should be more clearly written... Another thing could be the compiling and setup process of the entire project." These comments reinforce the needs of a better explanation in the process of installation and deployment. Combining these comments, the other three where assistance was given by the teachers and the answers presented in Table 10, we can infer that the process of creating concepts and concept relationships are quite easy and the authors of adaptive applications can be greatly helped by having pre-installed GALE servers.

The *adaptation rules* aspect shows that 10 out of 16 students consider this aspect not very easy. 10 students marked level 3 or higher for their answers. It is another aspect where we have to make some effort to make it easier and well documented. Considering that all students used a text editor to write their own configuration file (the GAM file) instead of the graphical tool, the students commented on some of their difficulties regarding the GAM language in the questionnaire. One said "I think it is not a good idea to let such a person (who does not have experience in the GAM language) edit GAM files in order to perform authoring." Another student complemented "Sometimes it felt a little bit inefficient, but the possibilities were available...". They were worried about the structure of the GAM file and about the language they had to write in, but they agreed in one thing: the level of experience in programming makes a difference. The first student went further on and said that "...such user should need a high level of programming experience before he/she will feel comfortable authoring in such way (using GAM language instead of graphical tool)." While the second said "The syntax of the GAM file was a little bit weird at the start of the development, but I got used to it quite fast."

Table 10 shows the aspect *concepts* and, also, *concepts and relationships* with low level of easiness (very easy) on authoring them. As we can see in Table 10, more than 80% of the answers are marked in level 3 or lower, with 50% of the answers marked in level 1 or 2 for the *concept* aspect, while the *concepts and relationships* aspect got roughly 44% of the answers in level 1 or 2. These aspects are the basis of the adaptation, and it is important to have almost half of the students creating the concepts and its relationships in an easy way in a short period of time.

As part of their applications some groups included a multiple-choice test (especially the ones developing a course or tutorial). The results for the test (quiz) aspect, presented in Table 10, suggest that the students did not need much effort to use the MC plugin and module. A multiple-choice test is represented as an HTML form (with some additional tags to identify questions and answers), is associated to a concept and is then presented as a regular course page. The "form" elements for questions and answers are generated by a GALE module and the MC test is evaluated through a GALE plug-in. Since the students all saw examples of MC tests (in source code) they could very easily learn to create their own tests.

6.3.2 Developing Extensions to GALE

The students also developed extensions to GALE, included in the applications they created (we present the subjects of these applications in the beginning of this chapter.). Some noteworthy extensions that were made are:

1. One group developed an extension to the multiple-choice test module and plug-in contained in the GALE distribution. In GALE an MC test is associated with a concept and the result of the test is a "knowledge" score for that concept. The students changed the code in order to link individual questions to concepts.

2. Another group also tackled the multiple-choice functionality to extract explanations of answers from pages of the application (instead of copy/pasting them into the quiz).

3. Some groups added new views, as used for an automatically generated navigation menu. One group added a progress bar to the top-level items in the navigation menu, and one group even created a completely new graphical (graph-based) navigation menu.

4. The adaptive radio used a new plug-in to randomly select a radio station with a bias towards previously liked stations.

Items 1, 2 and 3 are remarkable extensions since they deliver more options to the authors while requiring minor additional effort. By the analysis of the code (Java) by GALE developer experts, these extensions are regarded as of a moderate level of implementation code. Item 4 is innovative both in its conception and in the technique used. The implemented algorithm was never used before in an adaptive application created in and delivered by GALE.

In the questionnaire the first question about the development of extensions was "How difficult was the development of GALE extensions?". 13 students answered this question, with an average close to 3.5. (4 students marked easy or very easy and 7 difficult or very difficult.) To better understand the students point of view about the difficulty in implementing GALE extensions we asked about the ease or difficulty of specific aspects of designing and implementing extensions: installation, Java package explorer, code comments, developing clean code, deployment and test. Table 11 presents the students' answers for these six aspects on a scale of 1 to 5, where 1 is the lowest level (very easy) and 5 is the highest level (very difficult).

Aspects	**Levels**					
	1	**2**	**3**	**4**	**5**	**No Ans.**
Clean Code	1 (6%)	4 (25%)	4 (25%)	3 (18%)	1 (6%)	3 (18%)
Java Package Explore	2 (12%)	0 (0%)	4 (25%)	1 (6%)	5 (31%)	4 (25%)
Code Comments	1 (6%)	2 (12%)	3 (18%)	3 (18%)	4 (25%)	3 (18%)
Installation	0 (0%)	1 (6%)	2 (12%)	5 (31%)	4 (25%)	4 (25%)
Deployment	1 (6%)	1 (6%)	2 (12%)	4 (25%)	5 (31%)	3 (18%)
Test	0 (0%)	3 (18%)	3 (18%)	4 (25%)	4 (25%)	2 (12%)

Table 11: Number of answers about the aspects of development regarding the ease of implementation in a level from 1 (very easy) to 5 (very difficult).

The first aspect we would like to point out is whether the GALE source code could be considered to be clean code, which means whether the code was written in an understandable way to allow for easy maintenance afterward. Understanding the existing GALE code is important for creating extensions as the GALE code is a good "example" of the type of code the students had to develop. On average the students claimed the GALE code was written in a way that is not too difficult nor too easy to understand. (Note, they only commented on the existing GALE code, not the code of their extension.)

The *java package explore* aspect was not approved by the students, since 6 out of 12 of the valid answers (students who gave an answer) considered this aspect to be difficult or very difficult, and only 2 students considered that an easy aspect. This aspect was a big surprise for the GALE developer experts, since they had considered the java packages "self-explanatory." For example, the GALE processor classes are implemented in the *nl.tue.gale.ae.processors* package, or the GALE configuration classes are implemented in the *nl.tue.gale.ae.config* package. Each functional part of GALE is located in a different directory and has a corresponding name. One point that the students might have considered as part of this aspect is the place where the GAM file should be, as commented by a student: "The setup should be more clearly written. For example, the use and path placement of the default gam files (concept.gam)." This aspect was not discussed in detail with the students, but it is something that need to be clarified before the next experiments.

The *code comments* aspect is a topic in which the GALE developer experts really agree with the students, since the experts believe that the code relies a bit much on being self-explanatory and could indeed benefit from more comments. There are some classes that do not have explanatory comments about what the class does or what some lines of code do. That is also what the students marked in their answers. Most of them considered the code comments not that good, but a few of them mentioned that the comments are good or really good. One student pointed out that "There was hardly any code documentation, save for in-line comments."

It is clear from the students' comments and answers that the three aspects of the ease of implementation installation, deployment and test are the worst part for the development in GALE. Table 11 presents the results about these three aspects of implementation. Our research question is related to how the GALE system can effectively reach its goal as a generic and extensible adaptive system. In this case, we intend GALE to be usable as a basis for many adaptive system research projects, with other developers implementing and extending GALE for their specific research area. Thus, the GALE system's source code should really be easier to understand. The main complaint of the students was the installation and setup process. One said "The installation of the various tools is quite elaborate. A series of instruction movies for different platforms (Windows, Mac,...) would be helpful to stimulate a jump start." Another student complained "Getting GALE system to work in Eclipse is an absolute nightmare." The setup of GALE was quite often a source of complaint as we can see in another student's comment "I had a very hard time setting up GALE on a Macbook."

The deployment and test aspects of implementation were also a reason for complaining. When a student was asked about suggestions, comments or complaints about the development of GALE extensions he said “Provide clear instructions of how the whole complex editing trajectory should be followed if nothing changes.” Another student complained about the tests he/she was doing “What is also weird is that sometimes the very same code doesn’t work while it worked a day before...”. Both students did not give more details about what happened to let us improve the test aspect of the implementation, but we understand that with the current level of debugging skills of our students problems encountered during the development of GALE extensions can be hard to find.

6.4 Discussion and Conclusions

It is important to remark that, despite the problems faced in the use of GALE, the students were able to develop applications and extensions in GALE.At this point, it is important to say that the neither researchers nor students compared GALE to other systems, they were considering only GALE. The students used different adaptive techniques and methods; for example, some of them added *conditional inclusion of fragments*, some added *stretchtext*, some used template pages, some used content from a wiki, and some required specific handling of forms input. We consider an adaptive application different from another one regarding the adaptive techniques used, its context and the rules (and relationships) implemented to adapt the application. Thus, GALE can serve a lot of different types of adaptive applications.

The analysis of the questionnaire answers provided us useful insight about the authoring and development process in GALE as experienced by master students in the main field of computer science (and closely related fields). The main criticism made by the students is related to the *installation* and *setup* of both GALE and the development environment. More than 50% of the students considered the process of installing and deploying an application or an extension in GALE a very difficult process. Summarizing the comments of students, one said “We spent more time trying to understand how to implement things and debugging than actually doing relevant work.” By relevant work the student means the implementation (Java code) of GALE extensions. These criticism came somewhat as a surprise to us, especially coming from computer science students, because we have installed GALE and compiled and run it (with or without Eclipse) on different Windows, Linux and Mac systems before giving it to the students. Fortunately, this is something that can be remedied by improving the installation procedure and documentation. Also, authors of adaptive applications do not need to perform any of these step, only developers and system managers need to do that, since authors need to create the concepts, relationships and adaptation rules which it is not part of the installation process of GALE. We also remark that as GALE allows many adaptive applications (with different presentation style and different adaptation) to co-exist on a single server, authors in general need not be concerned with installation. The students only needed to set up their own server because they were asked to program extensions to GALE (a more technical task than authoring an adaptive course).

Another aspect of authoring in GALE with high criticism was the creation of the *adaptation rules*. 10 out of 16 students consider this aspect not very easy. 10 students marked level 3 or higher for their answers. Considering that all students used a text editor to write their own configuration file (the GAM file) instead of the graphical tool, it is very understandable that they had to do some effort in order to start writing and authoring with the GAM language, since they were not used to that language. The language can be partially written using Java code, but it is not mandatory as explained in the previous section. An additional problem is that the syntax sometimes differs from what they were used to in Java. The difficulties in using GAM emphasize the importance of providing templates of adaptation rules for common situations (like e.g. dealing with prerequisites). The "inheritance" feature of GALE allows authors to concentrate on defining concepts and relationships (which was found to be easy by the students) and to simply inherit adaptation rules (as writing them was found difficult). The GALE distribution thus needs to be expanded with more examples of adaptation rules that can be reused. In any case, we are aware that this is an aspect where we have to make some effort to make it easier and well documented.

It is clear from the students' comments and answers to the questionnaires that the three aspects of the ease of implementation installation, deployment and test are the worst part for the development in GALE. Besides, the group assignments had received good grades, the students' evaluation of the authoring and development process indicate that this was mainly due to a lot of effort by the students, not by the GALE environment making it easy. The students' feedback through the questionnaire provides good directions to improve the quality of GALE, both as a platform for creating adaptive applications or courses and as a platform that can be extended to suit different needs and desires by adaptive systems researchers.

Although students complained that the use of GALE (as author and as developer) was difficult, more difficult than we anticipated, our experiments with student-generated applications have shown that defining and implementing adaptation in a variety of adaptive applications is not really so hard: the students successfully completed their projects within the 6 week period that was allocated. The extensibility of GALE was an explicit research and design goal because we intend GALE to be usable as a basis for many adaptive (hypermedia) system research projects. Each project can create additions to GALE for their specific research purpose but does not have to design an adaptation platform from scratch, wasting valuable development time. Since students were able to start and complete extensions to GALE in just a few weeks GALE is fulfilling its promise of being usable as a basis for adaptive systems researchers, potentially saving months or even years of effort typically spent on developing an adaptive system from scratch, only to demonstrate the effect of some particular new adaptive feature or type of application.

The use of master students as the developers and authors of adaptive applications and extensions do not invalidate our experiment. Masters students are the next generation of computer professionals [46]. All students were required to use GALE and not allowed to use anything else instead. The authoring and development of GALE applications and extensions presented in this chapter was part of the group assignment of the "Adaptive Systems" course offered to master students at TU/e in the year of 2012. The student would fail the course if they did not do these two things. That is of course a limitation for the study but the limitation was exactly the same for all students.

7 New GALE Tools for Analyzing Navigation, Test and Quiz Logs

This chapter presents in detail the newly developed tools for analyzing navigation logs and test and quiz results. These tools were developed as a GALE plug-in, from scratch, instead of using or integrating on-line analytical processing (OLAP) tools or any known Web analytic tools. We chose this option in order to have a new GALE extension that can be used as an example for future developments, and to keep GALE as a self-contained tool, with no external tool with external support needed. The importance of such a tool for the authors is best explained using Chapters 3 and 4 as examples. These chapters present the evaluation of two versions of an adaptive course, and the main goal is to analyze the structure of the course from the point of view of the student's navigation. In general, the authors would like to check whether the students follow the structure of the course, meaning for instance whether the students follow *good* or *bad* links, and whether the students performed the tests (at the right moment, and how well they fared).

Chapters 3 and 4 present statistical measurements generated by an external tool. Indeed, the *empirical hub coefficient* (EHC), the *informative pages* and the correlation coefficient (between the out-degree of a concept and its EHC) were generated by an external tool. It was possible for us to use an external tool because we had access to the AHA! and GALE logs. (If you have GALE installed locally, you will also have access to the GALE logs, since its configuration is stored in the GALE config file.) However, in general, it is difficult for authors to make use of the logs due to the following problems and restrictions:

1. You need access to the application logs;

2. You need an external tool to analyze the logs;

3. The analysis is made "offline", which is not ideal since GALE serves on-line courses and applications.

4. Security and privacy. The AHA! and GALE log files contain all log data for that server since the log is not split into different files for different

applications, so authors analyzing the log have access to the log of other authors' applications.

An analysis tool built into the server could tackle these problems.

The main goal of the analysis tools (log, test and quiz tools) is to summarize the log data and present it in an "understandable" way to the authors of an adaptive application. In GALE these tools are implemented as plug-ins. The log analysis tool gives to the author an important overview of the students' navigation over the structure of the course. For instance, it can be used to answer our first two research questions: 1. does the annotation/hiding adaptive technique influence the choices of the students? 2. how does the interplay between the link structure of an adaptive course and the rules employed by the adaptation influence the choices of the students? One can say that the first question is not entirely answered since we do not know why the student is, for example, following a *bad* link. In this case, the author should ask the student why they are following these links using the *quiz* plug-in (we did that in the evaluation presented in Chapter 4). The analysis of such answers can be done by the quiz analysis tool. The evaluation presented in Chapter 4 was made partially with the support of the tool described in this chapter.

We have written Sections 7.1 and 7.2 in such a way that you can read them in any order. Section 7.1 is a more technical explanation and Section 7.2 describes the statistical measurements presented by these tools and present some examples. If this thesis were adaptive this navigation freedom would have been obvious, and redundancy between the sections could have been avoided to a larger extent.

7.1 Technical Support

The GALE system allows programmers to extend its functionality by implementing their own *Java Classes* for modules, processors, plug-ins, services, etc., as described in Appendix A. For the analysis tools we implemented plug-ins. A plug-in in GALE is an implementation that has the *pluginProcessor* associated with it. A plug-in is capable of "bypassing" the (remainder of the) processor pipeline to perform completely different processing and generate a result "page" that is not processed or adapted further by any processor that comes "later" in the pipeline. Appendix A presents an insight on how to implement a plug-in in GALE. For further references about creating a GALE plug-in, read David Smits' thesis [66]. A plug-in in GALE generates a "complete presentation". In more technical terms, a plug-in allows the programmers to have direct control over the client HTTP request and response. Good examples of how it works are the *logout plug-in*, where the user is logged out of the application, or the *MCPlugin* that evaluates a multiple test and presents a score to the user. In both cases they generate a response to the client and skip all the subsequent processors (more details in Appendix A) that would for instance generate a layout with header and navigation menu. When a request is handled by a plug-in the presentation will not show such header or menu (unless the plug-in itself generates them).

The GALE log is stored in a file on the GALE server. It is important to mention that the navigation, test and quiz logs are written in different files. The log files are logged "on-demand" (we mean: for every request to a concept, test or quiz the *LogProcessor* is called which generates a log entry and flushes its output buffer to guarantee completeness of the log even in the case of a web server crash). GALE uses the Spring[6] "inversion of control" (IOC) to configure and instantiate all the components. With Spring IOC the programmers can add the new *LogManager* to retrieve and analyze the log file by means of a newly implemented *Java class* in two steps: 1. Configure the Java class (in Spring it is called *bean*) in the config file (galeconfig.xml) to use the *LogManager bean property*; and 2. Inject the *bean property* in the implemented Java class.

The navigation log file contains the following data: the *date and time*, the *user*, the *concept*, the *resource* (URL of page), the *view*, the *link class* and the *referrer*. These data are better explained in Section 3.1.1. The *view*, the *link class* and the *referrer* are data that we missed in the evaluation presented in Chapter 3. The *link class* presents the annotation state of links followed by the user. We would like to mention that to log this state of links it was necessary to implement a new *processor*. The reason to implement this processor is that the annotation state is computed from UM, and the *UpdateProcessor* changes the basis for this computation before the *LogProcessor* runs. In that case, the new processor have to come before the *UpdateProcessor* in the pipeline. The new processor stores the link class in an UM attribute that is not touched later by the *UpdateProcessor*. The *LogProcessor* can

[6]See www.springsource.org for more details about Spring.

then log that state for our analysis. Note that simply reversing the order of execution of the *UpdateProcessor* and *LogProcessor* (to log the annotation state before the UM is changed) is not possible because the *resource* or page to be presented to the user is adaptively determined based on the UM state, and it is determined based on the *new* UM state after running the *UpdateProcessor*. Running the *LogProcessor* first would cause the wrong or perhaps even an undefined *resource* value being logged.

The implementation and use of the analysis tools as plug-ins in GALE can be considered an easy task for a Java programmer, since the plug-in is written as a Java class, and it extends the abstract class called *AbstractPlugin* which implements the *Plugin* interface. GALE contains a number of plug-ins already and these can also serve as example code for programmers creating new plug-ins. It is important to remark that there are three different plug-ins for the log, test and quiz analysis.

To make a plug-in available in GALE it is necessary to configure that in the galeconfig.xml file. You must insert the new plug-in Java class in the list of plug-ins known by the *PluginProcessor*. When creating a new plugin it is important to be aware that after the *PluginProcessor* has run all subsequent processors in the pipeline are skipped. The example below shows the lines inserted in galeconfig.xml to make the log analysis tool available in GALE.

```
<entry>
   <key>
      <value>analysis</value>
   </key>
   <bean class="nl.tue.gale.ae.processor.plugin.AnalysisPlugin">
      <property name="logManager" ref="logManager" />
   </bean>
</entry>
```

When a GALE request (URI) ends with `?plugin=analysis` the *AnalysisPlugin* will be called. The *AnalysisPlugin* is the implementation of the log analysis tool.

The mechanism that reads the navigation, test and quiz logs works in the same way. The GALE LogProcessor stores log information in files. This is fine for adding entries to log files but the file I/O mechanism is not really suitable for the analysis plug-ins to read the log files each time the plug-in is called. For better performance the analysis tool reads the log file the very first time it is called and then saves the data in an in-memory HyperSQL database[7]. The next time the analysis tool is called, it verifies whether there is an update (new lines are included) in the log file. If this is true, the tool only reads the new lines and inserts them in the database. An in-memory database could reduce the performance of the server, but in our experience the biggest log file we read was not longer than 1GB and did not pose a problem for the running server.

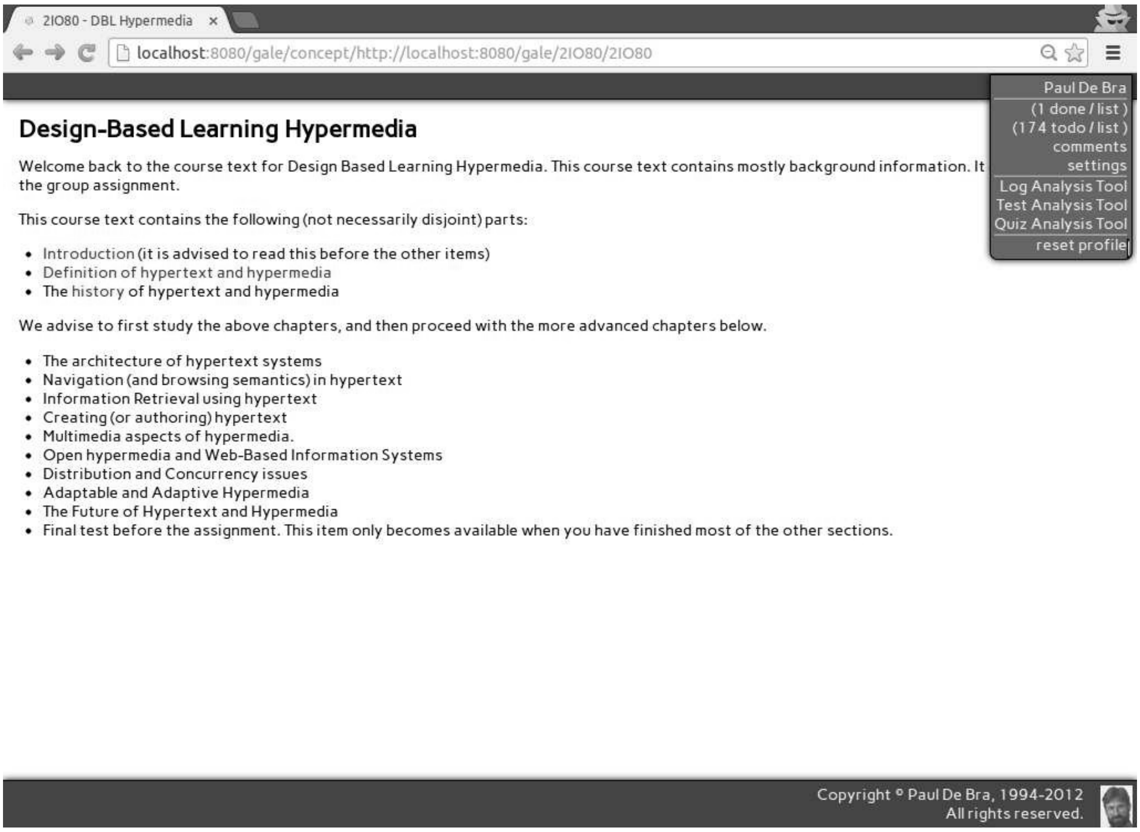


Figure 10: Screenshot of a concept with links to the analysis tools within the menu in the top right position.

For security reasons and to keep the confidentiality of the users, only authors of an application have access to the analysis tool and the tool also restricts that access to the data of the authors' own applications. Figure 10 shows a page with a menu in the header (top right position) used to access the analysis tools. To analyze the log

[7]See www.hsqldb.org for more details about HyperSQL Database.

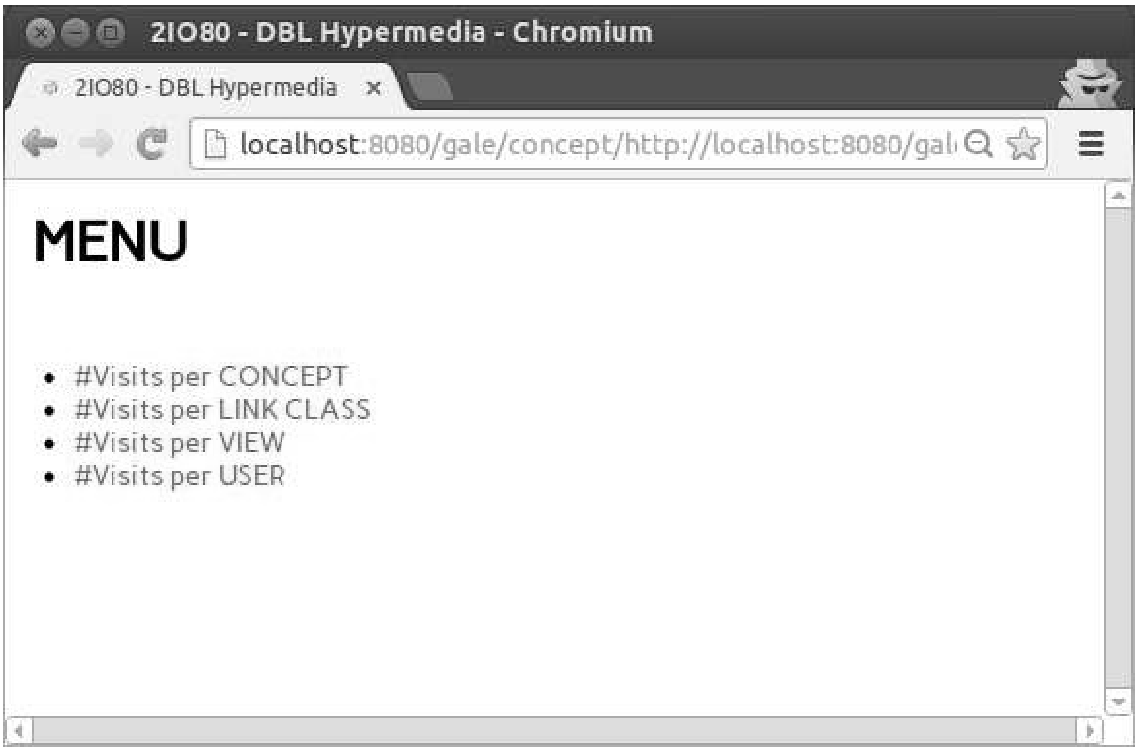
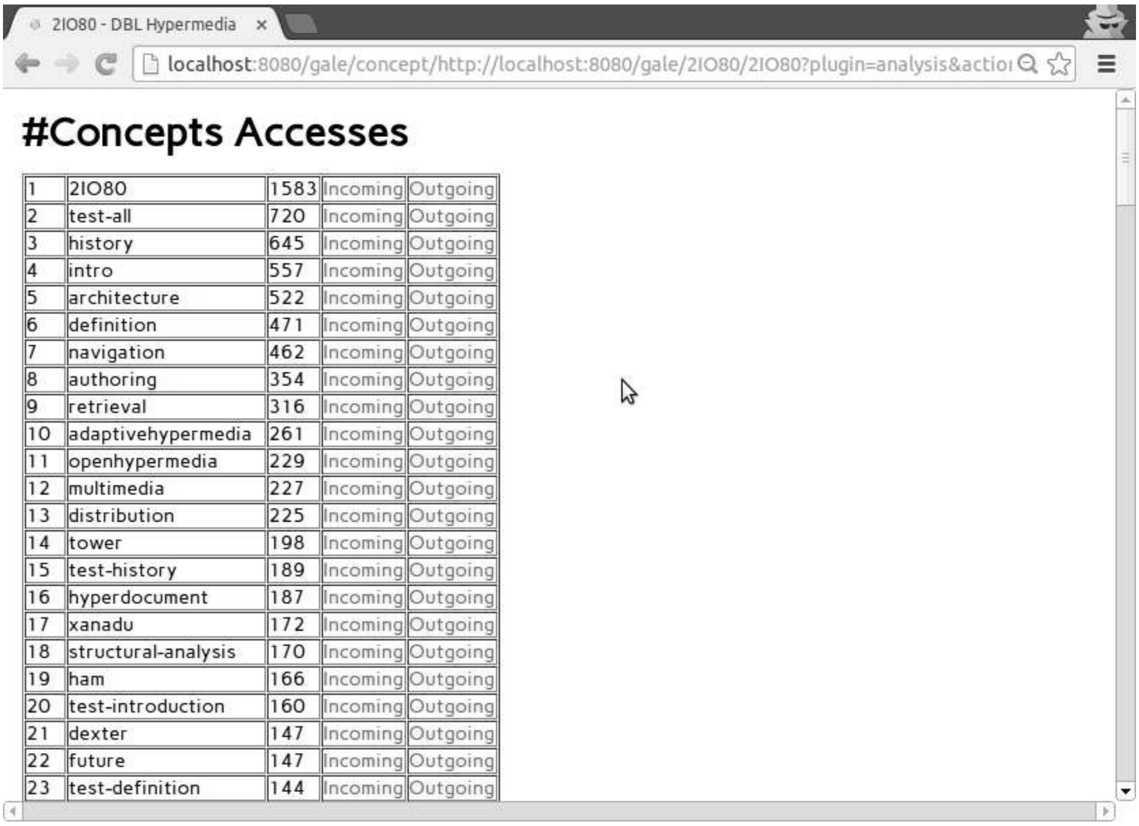


Figure 11: Screenshot of the menu options for the Log Analysis Tool.

of an application, such as the 2I080 course presented in the Chapter 4, the author has to access the 2I080 course and then (s)he can call the tools.

The *AnalysisPlugin* is called when an author selects the *Log Analysis tool* from the menu presented in Figure 10. Figure 11 shows a list (*hardcoded*) of possible statistical overviews the *AnalysisPlugin* can generate. The four topics shown in Figure 11 are HTML links which call the *AnalysisPlugin*. These links pass the URL as a parameter to the plug-in. The parameter is called *action* and it identifies the topic in the list. For example, when the author clicks on the first link topic *#Visits per CONCEPT*, the GALE request (URI) ends with `?plugin=analysis&action=concept`. In this example the *action* parameter has the value *concept*.

The analysis plug-in gets the HTTP request (using the GET method) and parameters. A query in the database is generated depending on the parameters. The query returns a *result set*. As a plug-in, the analysis tool generates a result "page", which means that the response is a HTML page. The HTML page is created based on the *result set*. Figure 12 shows the example above, where the



1	2IO80	1583	Incoming	Outgoing
2	test-all	720	Incoming	Outgoing
3	history	645	Incoming	Outgoing
4	intro	557	Incoming	Outgoing
5	architecture	522	Incoming	Outgoing
6	definition	471	Incoming	Outgoing
7	navigation	462	Incoming	Outgoing
8	authoring	354	Incoming	Outgoing
9	retrieval	316	Incoming	Outgoing
10	adaptivehypermedia	261	Incoming	Outgoing
11	openhypermedia	229	Incoming	Outgoing
12	multimedia	227	Incoming	Outgoing
13	distribution	225	Incoming	Outgoing
14	tower	198	Incoming	Outgoing
15	test-history	189	Incoming	Outgoing
16	hyperdocument	187	Incoming	Outgoing
17	xanadu	172	Incoming	Outgoing
18	structural-analysis	170	Incoming	Outgoing
19	ham	166	Incoming	Outgoing
20	test-introduction	160	Incoming	Outgoing
21	dexter	147	Incoming	Outgoing
22	future	147	Incoming	Outgoing
23	test-definition	144	Incoming	Outgoing

Figure 12: Visits per Concepts: the number of visits for a concept and links to drill-down the measure per *Incoming* link or per *Outgoing* link.

action parameter has the value *concept*. In the example, the *result set* returns the number of times a concept was visited by the users. The test and quiz plug-ins also use such parameters to retrieve data from the corresponding logs.

The last two columns of the table presented in Figure 12 contains links to drill-down the measure per incoming link or per outgoing link, respectively. We present more detail about these measures in the next section. At this point we would like to remark that these links also pass parameters through the GALE request, which is `?plugin=analysis&action=concepts_incoming&conceptParam=XXX` and `?plugin=analysis&action=concepts_outgoing&conceptParam=XXX`, for the incoming link and outgoing link, respectively. The XXX value depends on the line of the table, for example, the value of XXX in the first line is *2I080*, and in the second line is *test-all*.

The next section presents the possible statistical overviews that can be generated for GALE applications.

7.2 Statistical Measures

In this Section we present the statistical measurement used in the evaluation of an adaptive course and how it would help authors improve their own courses. Earlier in this chapter we said that the new tools were developed to keep GALE as a self-contained tool and to give users an example on how to implement new extensions to GALE (more precisely a GALE plug-in). We could have developed new tools presenting *basic* statistical measurement, e.g., the number of users who accessed an application or the number of tests performed in a course), but the power of these tools is similar to the OLAP tools: to roll-up and/or drill-down the data. To differ from OLAP tools we implemented the quiz and test analysis tools, and what we call *the EHC and informative pages table* (for *EHC* and *informative pages* refer to Section 3.3.1, Section 4.3.1 or summarized in [60]).

The analysis we have done in Chapter 4 was performed partially using the tools presented in this section. To better explain how such tools could, for instance, assist the author of the hypermedia course of Chapter 4, we recall the research questions we have made in this thesis: 1. does the annotation/hiding adaptive technique influence the choices of the students? 2. how does the interplay between the link structure of an adaptive course and the rules employed by the adaptation influence the choices of the students? These questions will lead us to a good example on how these tools will help the authors. Before we answer these questions, we present the statistics implemented in the log analysis tool. In general, the statistics present the number of visits of a specific measure grouped by other(s) measure(s). The tool presents the following statistics:

1. Visits per Users: it presents the number of visits (or clicks) per user; Clicking on the statistical number presented, the author is given a list with all entries for that specific user;

2. Visits per Concepts: it presents the number of visits for a concept. Here, the author can check the most visited concept. Authors can access the visits

per concepts per incoming link or the visits per concepts per outgoing link. Figure 12 shows an example of this measure for an *adaptive course* served by GALE.

3. Visits per Concepts per Incoming Link: it presents the number of visits for a concept grouped by incoming concept (the concept from which the user came). This measure allows the author to check from which concept the user came from to visit the current concept.

4. Visits per Concepts per Outgoing Links: it presents the number of visits for a concept grouped by outgoing concept (the concept accessed after visiting the current concept). In Chapter 3 we discuss the idea of *informative pages*. Recall that an *informative page* is a page with many links and information about the links, where the information intends to help the user choose the correct link to follow.

5. Visits per Link Class: it presents the number of visits (or clicks) to a concept grouped by link class (for more detail on link class see Chapter 3). Chapters 3 and 4 show the importance to the author to know which concepts are accessed through bad links. For this reason, the statistic *visits per link class* shows the number of visits (or clicks) grouped by link class.

6. Visits per Link Class per User: it presents a list of visits grouped by user and link class, i. e., the number of clicks in a specific link class grouped by user. This measure is discussed in Chapter 5, where the author can find the students who follow bad links.

7. Visits per Link Class per Concept: it presents a list of visits grouped by concept followed by the user using a specific link class, i. e., the number of clicks on a (link to a) concept using a specific link class. This measure is discussed in Chapter 5, where the author can find the different students who followed the same bad link.

8. Visits per View: it presents the number of clicks in a link grouped by view (for more detail on system views see Appendix A). An application can have different views. For example, the *Adaptive Course* presented in Chapter 4 has

three views: content, where the text of the concept is presented, menu, where the structure of the course is shown as a tree of concepts and sub-concepts, and, header, the where configuration menu and user information are presented. This measure shows in details how the views were used by the users in the application. The measure counts the number of times a user followed a link from within each view and also which (links to) concepts were followed from within each view. This is an important measure for the authors since they can understand how the users are navigating through the application. For this measure the author can see the visits per views per user and visits per view per link class.

9. Visits per View per User: it presents the number of clicks in a specific view grouped by user. For example, David Smits presents in his *thesis application* served by GALE a view called *next topic view* [66]. This view presents a link to the next concept in the concept tree order (this order was created by Smits), which means that if the thesis were a book the next topic would be the "next page" of the book. Considering that it is an adaptive application, and experts navigated through that, it would be interesting to check whether users used the *next topic view* or not.

10. Visits per View per Link Class: it presents the number of clicks in a specific view grouped by link class. Chapter 4 shows how students follow the advice given through link annotation and link hiding in the different views of the *adaptive course*. In this manner, authors can distinguish between bad links followed through annotated links versus bad links followed through hidden links. It is also possible to check whether students look for good links when they are navigating through the course pages or if they consider all the links that appear in menu-like structures. From this measure, authors access the visits per view per link class per user.

11. Visits per View per Link Class per User: it presents the number of clicks on links in a specific view using a specific link class grouped by user.

The first level is composed by: Visits per User; Visits per Link Class; Visits per View; and Visits per Concept. The tool present a link and the author can click and drill-down in the statistics. For example, when the author visits the statistic *visits per link class*, he can drill-down to check which users visited a specific link class or he can also drill-down to check which concepts were visited using a specific link class.

Another important remark is that from the lowest level (or the deepest level) the author can drill-down to the log entry list. For example, if the tool is showing the visits for *conceptA* where users accessed that using a *bad* link, the author can drill-down and check the complete information (student ID, date, activity, session, score, view, link class and referrer, presented in Chapter 4) for all log entries where the concept is *conceptA* and the link class is *bad*.

We show in this section how an author could use the analysis tools to answer our first two research questions (or, at least, try to answer these questions). We start with the question "how many *bad* links did the students follow and from which view?" To answer this specific question the author needs to access the log analysis tool and click on the statistic called *Visits per View*. At this moment, the tool will show the possible views of the course (the *hypermedia* course in Chapter 4 has the *content*, *header* and *static-tree-view (menu)* views), the number of visits for each of them and two links to drill-down the data: *Visits per View per User* and *Visits per View per Link Class*. Thus, to find out how many bad links were followed from each view it is necessary to follow the link: Visits per View per Link Class for each of the views. The *hypermedia* course would show 471 visits via bad links in total, where 262 from the menu view and 209 from the content view. Going further and clicking on the content view via bad links we find that 105 out of 209 clicks came from the *welcome* page. Roughly 50% of the bad link accesses from the content view came from the *welcome* page. Considering the link hiding/annotation and going deeper in this analysis, we find out that 87% of the bad link visits are via link annotation, as shown in Chapter 4. This is a simple example to show the importance of such statistics to the authors of a course. To get a complete answer to our research question we need more than only a simple question, we could have done a few more questions to try to infer whether the link hiding/annotation influence the student's choice. For example, we asked: a. which concept is more visited via bad links? b. how many users accessed a concept via bad links? c. how many users accessed a concept via bad links more than 5 times? These questions would start the analysis of the influence of the adaptive technique on the student's choice, but we are aware that we need more than only a few statistics to understand the student's behavior.

Our second research question is related to the link structure of the course and its influence over the student's choice. In our analysis in Chapter 3 we define the *empirical hub coefficient* (EHC) and *informative pages* (for more details refer to Section 3.3.1 or [60]). To confirm our definition of EHC and informative pages and our hypothesis that the link structure influence the student's choice, we implemented the EHC and informative pages tables, which shows, exactly, Tables 7 and 8 and the correlation coefficient between the EHC and the out-degree of the concepts in Chapter 4. Section 4.3.1 presents the analysis we have made for these tables.

We also developed a tool for the analysis of the tests, we call that *test analysis tool*. Before presenting the tool, we would like to remark that a test in GALE is a multiple choice form. The author inserts as many questions and possible answers (for each question) as he wants. There is no limit to the possible number of answers for each question, and different questions may have a different number of (correct and wrong) answers. The questions and answers are presented to the user randomly. The author can indicate the number of questions and (correct and wrong) answers that will be presented for the user.

First the author chooses a test, then the tool presents a list of questions and answers for that test. Figure 13 shows an example of the tool where the questions of a test called *test-history* are presented. The tool shows the number of times a question was answered by the users. In Figure 13 the words "*Appearence times: 4*" shows that the first answer (and also the second, in this case) was answered 4 times by the users. For each question we show the number of times the answers were shown for that question and, also, the number of times the users marked that answer. For example, in Figure 13 the first question and second answer shows *Intermedia 3/2*, where 3 means the number of time this answer appeared for that question and 2 means the number of times a student chose that answer. The correct answers are shown in bold.

The test analysis tool is an opportunity for the authors to check the users' answers and mistakes and it can be used by the author, depending on the questions he created, to review the concepts related to that question. For example, an author can see that in *question1* 85% of the users marked the wrong answer *answerA*. If

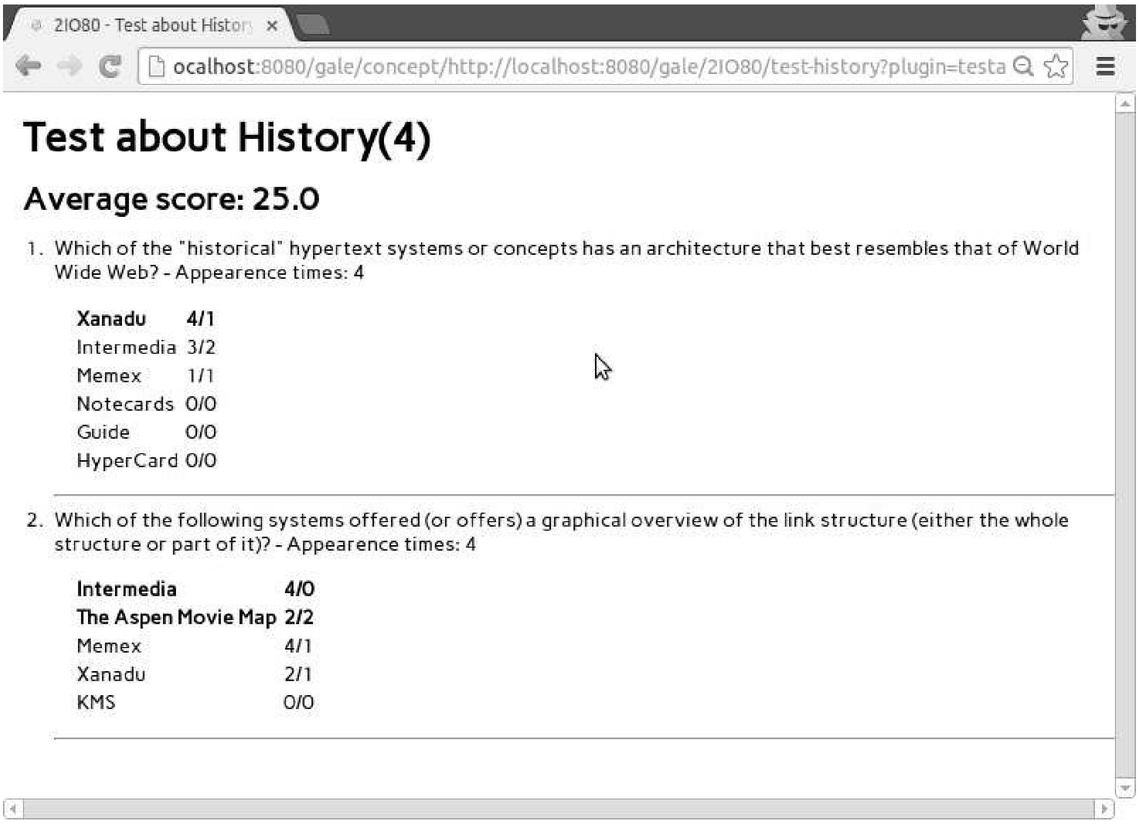


Figure 13: A screenshot of the test analysis tool. The test called *test-history* is shown as an example of the tool.

question1 is related to *conceptA*, it indicates that, probably, the approach used to explain this concept needs a review, since 85% of the users marked the wrong answer.

The quiz log analysis is another tool that we developed. A “quiz” in GALE is like a test, with no randomly selected question and answers, no right answer and, consequently, no score. An example on how and why to use a quiz is presented in Chapter 4. There we used a quiz to ask the students why they were following bad links. The quiz log analysis tool presents a simple table with the quiz questions and answers with the number of times each answer was marked by the users.

In the next chapter we discuss, in more detail, the work we intend to design and implement to make these tools more powerful.

8 Conclusion and Future Work

In this thesis we discussed the evaluation of an AH system and AH applications. The field of AH is already an important field in the academic research area with applications in e-learning, e-culture, entertainment, recommender systems, etc. We have concentrated on e-learning (or technology-enhanced learning to use a more modern term). The users, authors and developers are constantly bringing new ideas and needs to the AH field. In most of the cases these needs and ideas are subject to an evaluation.

This thesis covers the evaluation of adaptive courses regarding the influence of the adaptive technique of link hiding/annotation and the structure of links in the choices of the students. We present our contributions from this evaluation in Section 8.1. In the sense that the main goal of GALE is to become a generic and extensible adaptive system that can be used by many researchers and educators without all being forced to use the same type of presentation and adaptation based on the same type of rules. This thesis also covers the evaluation of the extensibility of GALE. We present our contributions from the GALE extensibility evaluation in Section 8.2.

8.1 Contributions on Adaptive Course Evaluation

Briefly, the main goal of all the evaluations presented in this thesis is to analyze the students navigation and students' feedback to assist authors of an adaptive course to improve the quality of the course and, also, to the developers of the GALE system to improve the quality and usability of the system. An important note for this work is that the analysis of the students navigation is not so simple as in a non-adaptive system, since the structure of links is adapted for different users of the system, based on their knowledge, navigation, characteristics, etc.

We have performed an analysis of the presentation classes of the links in the "Hypermedia Structures and Systems" course (to simplify we call it as the *Hypermedia* course) offered by the Eindhoven University of Technology since 1994, focusing on detecting whether students follow the so-called *bad* links and if their navigation behavior is affected by the presence of *bad* links. We found that pages that can be classified as authorities (i. e. that have many incoming links) make the students in some sense curious and cause them to investigate the pages even when they are not yet recommended by the system. The overall conclusion of our study is that when an author wishes for students to follow the adaptive guidance, *authorities* that are not recommended from the start should be avoided. We also found out that students who start the course late in the agenda tend to ignore the link hiding/annotation technique and follow more *bad* links than other students.

In a later analysis of an updated version of the *Hypermedia* course (which was carried out six years later with a different set of students) we also gave a questionnaire which the students who performed the course answered. As in the previous analysis, we found that students who start studying late in the course period tend to follow more *bad* links than students who use the entire period. We also noticed in the log and questionnaire analysis that students often followed *bad* links because they were curious about the linked non-recommended concept or because they are exploring to know more about what will be offered in the course before they start studying the material in more depth. We have observed that students navigating via *bad* links have a tendency of continuing to use *bad* links in the subsequent pages. It indicates that a student with a certain learning style needs to have an overview of what the course is about and the authors needs to take this into account providing different overview to the students to avoid a student following bad links.

In the analysis of the updated *Hypermedia* course, we also noticed that there are more log entries during the exam (4,495 entries) than during the whole period leading up to the exam (3,976 entries). An important work to be done in later courses is to stimulate the students right from the beginning of the course term[8]. The absence of classes or other contact with a tutor during the course period leads students to postpone their study activity until right before the exam.

One of our research questions says that the adaptive link hiding technique is an effective model of adaptation and students' guidance. Our findings in the first evaluation lead us to an educated guess about the influence of the adaptive hiding technique to guide the students through the course: most of the students did not

[8]This has been tried in the 2014 edition by making an attempt at the final test of the course a prerequisite for participating in the exam. Alas, most students still postponed most of the work to the last two days before the exam, so in 2015 we will need to try yet something else.

always follow the system's suggestion because of their curiosity. Regarding the second evaluation, we observed that the *link hiding* technique is an effective way to guide the students through the course. However, our analysis also revealed that the *link annotation* technique is not as effective as the hiding technique in guiding the students. We say effective in the sense that students follow the structure of the course as planned by the authors.

8.1.1 Hubs and Informative Pages

We have also carried out a structural analysis of the course, comparing the definition of *hubs* to new empirical measures, called *empirical hub coefficient* (EHC) and *informative pages*. Intuitively, *hubs* are pages that contain a large number of links to others pages. We cannot apply this concept directly in our setting, since in an adaptive application the number of links of a web page may change over time. The EHC of a page (or, more precisely, a concept) X is the ratio between the number of times students clicked on a link of X to go to a different page (or concept) and the number of times that students accessed page X. We also introduced the notion of *informative pages*, which are pages that not only have a large number of links, thereby being good candidates for *hubs*, but also contain enough information to *guide* the student through the links of the page.

We have observed that *empirical hubs* appear in the course we studied, and most of these *hubs* turned out to also be *informative pages* that guided the navigation of the students through the large number of links that *empirical hubs* contain. We also identified the presence of non-informative *hubs* that give the student many choices but no guidance. An author should ensure that all *hubs* have information to guide the students through the links in the page. In other words, an author should turn the hubs into *informative pages*.

8.2 Contributions on GALE Extensibility Evaluation

Another important goal of this thesis was the evaluation of the extensibility and modularity of GALE, since it was created to be a generic open source extensible adaptive engine. Our hypothesis is that GALE reaches its goal as a generic and extensible adaptive system serving different kinds of adaptive applications. In this case, we got feedback from master students through a questionnaire and by analyzing the applications and extensions developed by them. The students had a few problems developing their own applications and extensions. Despite the problems faced in the use of GALE, the students were able to develop applications and extensions of a high level of quality. The students used different adaptive techniques and methods; for example, some of them added *conditional inclusion of fragments*, some added *stretchtext*, some used template pages, some used content from a wiki, and some required specific handling of forms input. These show that GALE can serve a lot of different types of adaptive applications.

The questionnaire analysis about the creation of adaptive application and the development of GALE extensions by master students gave us a very deep insight on that task. The main criticism made by the students is related to the installation and setup of both GALE and the development environment. More than 50% of the students considered the process of installing and deploying an application or an extension in GALE a very difficult process. If we were to summarize the comments of all students in one sentence we would choose this one: "*we spent more time trying to understand how to implement things and debugging than actually doing relevant work.*" We are aware that this comment does not represent the opinion of all students, and we observed that most students shared this criticism. Fortunately this criticism can be remedied by improving the installation procedure and documentation. Also, authors of adaptive applications do not need to perform these steps, only developers and system managers need to do that. Another important note is that the installation and deployment are tasks that need to be performed only once by the developers. Thus, the problems reported about these tasks are not directly connected to the extensibility of GALE.

Although students complained that the use of GALE (as an author or developer) was difficult, more difficult than we anticipated, our experiments with student-generated applications and extensions have shown that defining and implementing adaptation in a variety of adaptive applications is not really so hard: the students successfully completed their projects within the 6 week period that was allocated. In fact, within the 6 weeks of the project, students not only created their own adaptive applications, but also implemented different kinds of GALE extensions, as shown in Chapter 6. Indeed, a application created (also the extensions) differ from another application considering the adaptive techniques used, its context and the rules (and relationships) implemented to adapt the application.

The extensibility of GALE was an explicit research and design goal because we intend GALE to be usable as a basis for many adaptive (hypermedia) system research projects. Each project can create additions to GALE for their specific research purpose but does not have to design an adaptation platform from scratch, wasting valuable development time.

8.3 Limitations

In this thesis we investigated a few aspects of adaptation that influence the choice of the students for an adaptive course created and served by AHA! and GALE. A limitation we would like to point out regards the group of students used in this evaluation. The experiments were made with three different groups: two groups of bachelor students and one group of master students. The experiments with the group of bachelor students were made in the evaluations in Chapters 3 and 4, with 76 and 46 students, respectively. Despite not being the ideal scenario, it gave good insights on the influence of the adaptation on the students choice. The group of master students appeared in the evaluation presented in Chapter 6, and they were only 16. This evaluation was also part of the group assignment of the "Adaptive Systems" course offered to master students at TU/e. This is a limitation of this experiment, and we are aware that a generalization of our findings is too weak, such as the validation of our hypothesis. Nonetheless, this study gave us a good starting point to prepare quality materials to improve the user experience in developing new applications and extensions.

The small number of courses we analyzed, and the small number of students present in each course, prevent us from drawing general conclusions from all results regarding the case studies and survey method of evaluation suggested in this thesis. In one hand, if the evaluation turns out to produce poor results, this does not imply that AHA! and GALE are bad platforms: the bad results could have been caused because the author of the course did not do a good job in designing the course. On the other hand, if the outcome of the evaluation is good, it could still have been the case that AHA! and GALE are bad authoring platforms: the course could have been evaluated well simply because the authors of those specific courses were particularly effective in designing and implementing the course. Consequently, our results in this thesis consists of a proof of concept that the adaptive technique of link hiding/annotation and the structure of links of a course influence in the choices of the students, with some lessons to be learned from our case studies and surveys. For example, an *informative page* is a good way to offer some guidance to the student, and the process of installing and setup GALE is a difficulty process that needs improvements.

8.4 Future Work

In Chapter 7 we discussed the analysis tools that help authors of adaptive applications in GALE investigate the log navigation, test answers and quiz answers of their users. The analysis tools were partially used in this thesis to evaluate the adaptive course in Chapter 4. For future work we intend to improve the number of statistics presented in these tools. Here we list some of them:

1. The students who visited the related concepts after failing the test.
2. The students who visited the related concepts before answering the test.
3. The percentage of students that failed the test and visited at least one of the related concepts.
4. The percentage of students that failed the test and did not visit any of the related concepts.
5. The percentage of students that passed the test and visited at least one of the related concepts.
6. The date and time the quiz or test was answered.

In GALE the logs are stored in text files (including navigation logs, test and quiz results). It consumes much time and effort to read and write in these files. An improvement for GALE is to implement new Java classes to log using a database. This implementation will reduce the time needed to develop tools for analyzing the logs, and also speed up the time to process the logs.

At this moment, only the authors are authorized to access the analysis tools for their own applications. We are aware that this is not the ideal. For future work we intend to create a mechanism where the author can allow other users to access the analysis.

A structure that will make the navigation tool more powerful is to implement a filter and a sort mechanism. With these two mechanisms the user can create his own statistical measures. For example, one can wish to filter the data to see only the students that visited a given concept, or that visited a certain concept via a bad link, or that visited that concept on a given date.

We stated in Section 8.3 our limitations, one of which being that the results observed in this thesis are restricted to the evaluation of only two adaptive courses. For future work we are interested in carrying out the evaluation of other courses, using the techniques presented in Chapter 5 and 4. Link structure analysis of others courses would confirm the definitions of *empirical hubs* and *informative pages* we propose in this thesis.

For the AH system field there are many challenges to be tackled, including the evaluation topic. The privacy of the users is one thing that needs to be improved, such as having a client-side user model, created and maintained through the browser, without the need to install any software on the client. Another challenge is to have part of the computation for the adaptation in the client side. Even a "lowly" tablet or phone has enough processing power to perform the adaptation for a single user. The last thing, and most challenging, the adaptation should in the future become able to correct itself. Automated analysis of the behavior of the entire user population should reveal that certain "optimization" of the adaptation is possible. Realizing such *adaptation of the adaptation* or *meta-adaptation* requires interesting new research, especially to ensure that the adaptation can only change "for the better".

References

[1] F. Abel, N. Henze, E. Herder, and D. Krause. Interweaving public user profiles on the web. In *Proceedings of the 18th international conference on User Modeling, Adaptation, and Personalization*, UMAP'10, pages 16–27, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-13469-6, 978-3-642-13469-2. doi: 10.1007/978-3-642-13470-8_4.

[2] M. Balík and I. Jelínek. Towards semantic web-based adaptive hypermedia model. In *PhD Symposium at the European Semantic Web Conference (ESWC)*, Tenerife, ES, 2008.

[3] T. Bates. *Technology, e-learning and distance education*. Studies in Distance Education. Routledge, EUA, 2005. ISBN 9780415284363.

[4] R. A. Botafogo, E. Rivlin, and B. Shneiderman. Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Trans. Inf. Syst.*, 10(2):142–180, Apr. 1992. ISSN 1046-8188. doi: 10.1145/146802.146826.

[5] C. Boyle and A. O. Encarnacion. An adaptive hypertext reading system. *User Modeling and User-Adapted Interaction*, 4:1–19, 1994. ISSN 0924-1868. 10.1007/BF01142355.

[6] P. Brusilovsky. Intelligent tutoring systems for World Wide Web. In *Proceedings of the Third International WWW Conference*, pages 42–45, Darmstadt, 1995.

[7] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2–3):87–129, 1996.

[8] P. Brusilovsky. Adaptive hypermedia. *User Modeling and User Adapted Interaction*, 11(1–2):87–110, 2001.

[9] P. Brusilovsky and J. Eklund. A study of user model based link annotation in educational hypermedia. *Journal of Universal Computer Science*, 4(4):429–448, 1998.

[10] P. Brusilovsky and L. Pesin. Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. *Journal of Computing and Information Technology*, 6:27–38, 1998.

[11] P. Brusilovsky, E. W. Schwarz, and G. Weber. ELM-ART: An intelligent tutoring system on World Wide Web. In *Proceedings of the Third International Conference on Intelligent Tutoring Systems*, pages 261–269, London, UK, 1996. Springer-Verlag. ISBN 3-540-61327-7.

[12] P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: a tool for development adaptive courseware. *Computer Networks and ISDN Systems*, 30(1–7):291–300, 1998. ISSN 0169-7552. doi: 10.1016/S0169-7552(98)00082-8. Proceedings of the Seventh International World Wide Web Conference.

[13] P. Brusilovsky, C. Karagiannidis, and D. Sampson. The benefits of layered evaluation of adaptive applications and services. In S. Weibelzahl, D. Chin, and G. Weber, editors, *Empirical Evaluation of Adaptive Systems: Proceedings of Workshop at the Eighth International Conference on User Modeling, UM2001*, pages 1–8, 2001.

[14] P. Brusilovsky, C. Karagiannidis, and D. Sampson. Layered evaluation of adaptive learning systems. *International Journal of Continuing Engineering Education and Lifelong Learning*, 14:2004, 2004.

[15] V. Bush. As we may think. *The Atlantic Monthly*, 1945.

[16] L. Calvi. Formative evaluation of adaptive CALLware: a case study. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 276–279, London, UK, 2000. Springer-Verlag. ISBN 3-540-67910-3.

[17] D. N. Chin. Empirical evaluation of user models and user-adapted systems. *User Modeling and User-Adapted Interaction*, 11(1-2):181–194, 2001. ISSN 0924-1868. doi: 10.1023/A:1011127315884.

[18] O. Conlan, C. Hockemeyer, V. Wade, and D. Albert. Metadata driven approaches to facilitate adaptivity in personalized elearning systems. *The Journal of Information and Systems in Education*, 1:38–44, 2002.

[19] P. De Bra. Teaching hypertext and hypermedia through the Web. *Journal of Universal Computer Science*, 2(12):797–804, 1996.

[20] P. De Bra and L. Calvi. AHA! an open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia*, pages 115–140, 1998.

[21] P. De Bra, G.-J. Houben, and H. Wu. AHAM: a Dexter-based reference model for adaptive hypermedia. In *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia*, HYPERTEXT '99, pages 147–156, New York, NY, USA, 1999. ACM. ISBN 1-58113-064-3. doi: 10.1145/294469.294508.

[22] P. De Bra, A. Aerts, D. Smits, and N. Stash. AHA! meets AHAM. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, AH '02, pages 388–391, London, UK, 2002. Springer-Verlag. ISBN 3-540-43737-1.

[23] P. De Bra, A. Aerts, B. Berden, B. de Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash. AHA! the adaptive hypermedia architecture. In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, HYPERTEXT '03, pages 81–84, New York, NY, USA, 2003. ACM. ISBN 1-58113-704-4. doi: 10.1145/900051.900068.

[24] P. De Bra, T. Santic, and P. Brusilovsky. AHA! meets Interbook and more... In *Proceedings of the AACE ELearn 2003 Conference*, pages 57–64, 2003.

[25] P. De Bra, N. Stash, and D. Smits. Creating adaptive applications with AHA!, tutorial for AHA version 3.0. In *Tutorial at the AH2004 Conference, Eindhoven, Computing Science Report*, 2004.

[26] P. De Bra, D. Smits, and N. Stash. The design of AHA! In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, HYPERTEXT '06, pages 171–195, New York, NY, USA, 2006. ACM. ISBN 1-59593-417-0. doi: 10.1145/1149941.1149942.

[27] P. De Bra, D. Smits, and N. Stash. Creating and delivering adaptive courses with AHA! In *Proceedings of the First European Conference on Technology Enhanced Learning - EC-TEL'06*, volume 4227, pages 21–33, Crete, 2006. Springer.

[28] P. De Bra, M. Pechenizkiy, K. van der Sluijs, and D. Smits. GRAPPLE: integrating adaptive learning into learning management systems. In J. Luca and E. Weippl, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008*, pages 5183–5188, Chesapeake, VA, USA, 2008.

[29] P. De Bra, E. Knutov, D. Smits, N. Stash, and V. F. C. Ramos. Gale: a generic open source extensible adaptation engine. *New Review of Hypermedia and Multimedia*, 19(2):182–212, 2013. doi: 10.1080/13614568.2013.806961.

[30] M. De Jong and P. J. Schellens. Reader-focused text evaluation. *Journal of Business and Technical Communication*, 11(4):402–432, 1997. doi: 10.1177/1050651997011004003.

[31] P. De Vrieze. *Fundaments of Adaptive Personalization*. PhD thesis, Radboud University Nijmegen, The Netherlands, 2006.

[32] B. Devdutta. AlcoZone: An adaptive hypermedia based personalized alcohol education. Master's thesis, Electrical and Computer Engineering Department, Virginia Tech, USA, 2006.

[33] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*, pages 285–311. Springer, 2008.

[34] C. Frasson, G. Gauthier, and G. McCalla. Intelligent tutoring systems. In *Proceedings of the Second International Conference on Intelligent Tutoring Systems*, Berlin, 1992. Springer-Verlag. ISBN 0-387-55606-0.

[35] C. Gena. Methods and techniques for the evaluation of user-adaptive systems. *Knowledge Engineering Review*, 20(1):1–37, 2005. ISSN 0269-8889. doi: 10.1017/S0269888905000299.

[36] C. Gena and S. Weibelzahl. The adaptive Web. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *Lecture Notes in Computer Science*, chapter Usability engineering for the adaptive Web, pages 720–762. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 978-3-540-72078-2.

[37] M. Gonschorek and C. Herzog. Using hypertext for an adaptive help system in an intelligent tutoring system. In *AI-ED, 7th World Conference on Artificial Intelligence in Education*, pages 274–281, 1995.

[38] G. Grimmett and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, August 1992.

[39] M. Hendrix and A. I. Cristea. Design of the CAM model and authoring tool. In *4th European Conference on Technology-Enhanced Learning*, EC-Tel 2009, 2009.

[40] N. Henze and W. Nejdl. Adaptivity in the KBS Hyperbook system. In *2nd Workshop on Adaptive Systems and User Modeling on the WWW, workshop held in conjunction with the WWW Conference (WWW8) and the International Conference on User Modeling*, Toronto, CA, 1999.

[41] K. Höök. Evaluating the utility and usability of an adaptive hypermedia system. *Knowledge-Based Systems*, 10(5):311–319, 1998. ISSN 0950-7051. doi: 10.1016/S0950-7051(97)00034-8. Intelligent User Interfaces.

[42] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Guiding students to the right questions: adaptive navigation support in an e-learning system for java programming. *Journal of Computer Assisted Learning*, 26(4):270–283, 2010. ISSN 1365-2729. doi: 10.1111/j.1365-2729.2010.00365.x.

[43] C. Kaplan, J. Fenwick, and J. Chen. Adaptive hypertext navigation based on user goals and context. *User Modeling and User-Adapted Interaction*, 3(3): 193–220, 1993. doi: 10.1007/BF01257889.

[44] C. Karagiannidis and D. G. Sampson. Layered evaluation of adaptive applications and services. In *AH '00: Proceedings of the International*

Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pages 343–346, London, UK, 2000. Springer-Verlag. ISBN 3-540-67910-3.

[45] J. Kay and R. J. Kummerfeld. An individualised course for the C programming language. In *Proceedings of the Second International WWW Conference*, Chicago, 1994.

[46] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. E. Emam, and J. Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Transaction Software Engineering*, 28:721–734, August 2002. ISSN 0098-5589. doi: 10.1109/TSE.2002.1027796.

[47] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999. ISSN 0004-5411. doi: 10.1145/324133.324140.

[48] E. Knutov. *Generic adaptation frameform for unifying adaptive web-based systems*. PhD thesis, Eindhoven University of Technology, The Netherlands, 2012.

[49] E. Knutov, P. De Bra, and M. Pechenizkiy. AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, 15(1):5–38, 2009.

[50] E. Knutov, P. De Bra, and M. Pechenizkiy. Generic adaptation framework: a process-oriented perspective. *Digital Information*, 12(1), 2011.

[51] M. D. LeCompte and J. P. Goetz. Problems of reliability and validity in ethnographic research. *Review of educational research*, 52(1):31–60, 1982.

[52] S. McGuigan. The use of statistics in the British journal of psychiatry. *The British Journal of Psychiatry*, 167(5):683–688, 1995. doi: 10.1192/bjp.167.5.683.

[53] B. Meyer. Retail user assistant: Evaluation of a user-adapted performance support system. In B. Blumenthal, J. Gornostaev, and C. Unger, editors, *Human-Computer Interaction*, volume 876 of *Lecture Notes in Computer Science*, pages 45–55. Springer Berlin / Heidelberg, 1994. 10.1007/3-540-58648-2_25.

[54] J. Oberlander, M. O'Donnell, C. Mellish, and A. Knott. Conversation in the museum: experiments in dynamic hypermedia with the intelligent labelling explorer. *The New Review of Hypermedia and Multimedia*, 4:11–32, 1998. ISSN 1361-4568.

[55] A. Ortigosa and R. M. Carro. The continuous empirical evaluation approach: evaluating adaptive web-based courses. In *UM'03: Proceedings of the 9th international conference on User modeling*, pages 163–167, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-40381-7.

[56] A. Paramythis, A. Totter, and C. Stephanidis. A modular approach to the evaluation of adaptive user interfaces. In S. Weibelzahl, D. Chin, and G. Weber, editors, *Proceedings of the Workshop on Empirical Evaluations of Adaptive Systems, held in the context of the 8th International Conference on User Modeling (UM'2001)*, pages 9–24, Sonthofen, Germany, 2001.

[57] A. Paramythis, S. Weibelzahl, and J. Masthoff. Layered evaluation of interactive adaptive systems: framework and formative methods. *User Modeling and User-Adapted Interaction*, 20:383–453, December 2010. ISSN 0924-1868. doi: 10.1007/s11257-010-9082-4.

[58] R. Paramythis and S. Weibelzahl. A decomposition model for the layered evaluation of interactive adaptive systems. In *Proceedings of the 10th Intenational Conference on User Modelling, Lecture Notes in Artificial Intelligence 3538*, pages 438–442. Springer, 2005.

[59] E. Ploum. Authoring of adaptation in the GRAPPLE project. Master's thesis, Eindhoven University of Technology, The Netherlands, 2009.

[60] V. F. C. Ramos and P. De Bra. The influence of adaptation on hypertext structures and navigation. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT '10, pages 77–82, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0041-4. doi: 10.1145/1810617.1810632.

[61] V. F. C. Ramos, P. De Bra, and G. B. Xexéo. A continuous layered evaluation of adaptive hypermedia systems (chapter 8). In Y. Psaromiligkos,

A. Spyridakos, and S. Retalis, editors, *Evaluation in e-Learning*, pages 145–158. Nova Publishers, 2012.

[62] V. F. C. Ramos, P. De Bra, and D. Smits. GALE extensibility evaluation: a qualitative approach. In T. Bastiaens and G. Marks, editors, *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN) 2013*, pages 296–305, Las Vegas, NV, USA, 2013. AACE. ISBN 978-1-939797-05-6.

[63] V. F. C. Ramos, P. De Bra, and G. B. Xexéo. Qualitative and quantitative evaluation of an adaptive course in GALE. In D. Hernandez-Leo, T. Ley, R. Klamma, and A. Harrer, editors, *Scaling up Learning for Sustained Impact*, volume 8095 of *Lecture Notes in Computer Science*, pages 301–313. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40813-7. doi: 10.1007/978-3-642-40814-4_24.

[64] C. Romero, S. Ventura, D. Jose Antonio, and P. De Bra. Personalized links recommendation based on data mining in adaptive educational hypermedia systems. In *Creating New Learning Experiences on a Global Scale. Second European Conference on Technology Enhanced Learning, EC-TEL 2007*, pages 293–305. Springer, 2007. doi: 10.1007/978-3-540-75195-3.

[65] P. Seefelder de Assis and D. Schwabe. A general meta-model for adaptive hypermedia systems. In P. De Bra and W. Nejdl, editors, *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Bases Systems*, AH '04, pages 433–436, 2004.

[66] D. Smits. *Towards a Generic Distributed Adaptive Hypermedia Environment.* PhD thesis, Eindhoven University of Technology, The Netherlands, 2012.

[67] D. Smits and P. De Bra. GALE: a highly extensible adaptive hypermedia engine. In *Proceedings of the 22nd ACM conference on Hypertext and Hypermedia*, HT '11, pages 63–72, New York, NY, USA, 2011. ACM. doi: 10.1145/1995966.1995978.

[68] N. Stash and P. De Bra. Incorporating cognitive styles in AHA! (the adaptive hypermedia architecture). In *Proceedings of the IASTED International Conference Web-Based Education*, pages 378–383, 2004.

[69] O. Stock, M. Zancanaro, P. Busetta, C. Callaway, A. Krüger, M. Kruppa, T. Kuflik, E. Not, and C. Rocchi. Adaptive, intelligent presentation of information for the museum visitor in PEACH. *User Modeling and User-Adapted Interaction*, 17:257–304, 2007. ISSN 0924-1868. 10.1007/s11257-007-9029-6.

[70] P. Totterdell and E. Boyle. The evaluation of adaptive systems. In D. Browne, P. Totterdell, and M. Norman, editors, *Adaptive User Interfaces*, pages 161–194. Academic Press, London, 1990.

[71] O. Troyer, F. Kleinermann, B. Pellens, and A. Ewais. Supporting virtual reality in an adaptive web-based learning environment. In U. Cress, V. Dimitrova, and M. Specht, editors, *Learning in the Synergy of Multiple Disciplines*, volume 5794 of *Lecture Notes in Computer Science*, pages 627–632. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-04635-3. doi: 10.1007/978-3-642-04636-0_59.

[72] K. van der Sluijs, J. Hidders, E. Leonardi, and G.-J. Houben. GAL: A generic adaptation language for describing adaptive hypermedia. In *Proceeding of the International Workshop on Dynamic and Adaptive Hypertext: Generic Frameworks, Approaches and Techniques (DAH'09) in conjunction with ACM Hypertext 2009*, DAH'09, pages 13–24, 2009.

[73] L. Van Velsen, T. Van Der Geest, R. Klaassen, and M. Steehouder. User-centered evaluation of adaptive and adaptable systems: a literature review. *The Knowledge Engineering Review*, 23(03):261–281, 2008. doi: 10.1017/S0269888908001379.

[74] Y. Wang, N. Stash, L. Aroyo, P. Gorgels, L. Rutledge, and G. Schreiber. Recommendations based on semantically enriched museum collections. *Web Semant.*, 6(4):283–290, Nov. 2008. ISSN 1570-8268. doi: 10.1016/j.websem.2008.09.002.

[75] G. Weber and M. Specht. User modeling and adaptive navigation support in WWW-based tutoring systems. In *International Conference on User Modeling*, pages 289–300. Springer, 1997.

[76] S. Weibelzahl. Evaluation of adaptive systems. In *8th International Conference on User Modeling*, volume 2109, pages 292–294, Berlin, 2001. Springer.

[77] S. Weibelzahl and G. Weber. Advantages, opportunities, and limits of empirical evaluations: Evaluating adaptive systems. *Künstliche Intelligenz*, 3:17–20, 2002.

[78] S. Weibelzahl and G. Weber. Evaluating the inference mechanism of adaptive learning systems. In *Proceedings of the 9th international conference on User modeling*, UM'03, pages 154–162, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-40381-7.

[79] G. E. Welche and S. G. Gabbe. Review of statistics usage in the American journal of obstetrics and gynecology. *American Journal of Obstetrics and Gynecology*, 175(5):1138–1141, 1996.

[80] H. Wu. *A Reference Architecture for Adaptive Hypermedia Applications*. PhD thesis, Eindhoven University of Technology, The Netherlands, 2002.

[81] J. M. Yancey. Ten rules for reading clinical research reports. *American Journal of Orthodontics & Dentofacial Orthopedics*, 109(5):558–564, 1996.

A GALE: Open Generic Extensible Adaptive System

This appendix describes GALE system [66, 67], the Generic Adaptation Language and Engine that came out of the GRAPPLE EU FP7 project. The main focus of this appendix is not only describe GALE, but also present the extensible nature of GALE. Most of this appendix is taken from the PhD thesis of David Smits [66] and associated main publication [67] and is only included here because a good basic understanding of the GALE architecture is needed to understand our extensions (see chapter 7) and the evaluation of extensions made by students (see chapter 6). We also have a brief and comprehensive description of this appendix in [29].

The GALE system was recently introduced as an evolution of the AHA! system [26]. AHA! was originally designed to serve a hypertext course taught through the Web [19]. AHA! aimed at being generic and general purpose and was perhaps best described in the world's first adaptive paper [26] presented at the ACM Hypertext 2006 conference (through an adaptive talk). Knutov et al. [49] describe many new adaptation techniques developed to date and provide a list of challenges for creating a new generic adaptive hypermedia system. That research has led to the GAF model (Generic Adaptation Framework) [50, 48], capable of dealing with several types of adaptive systems and applications, from "traditional" adaptive hypermedia to personalized search and recommender systems. GALE tackles this challenge through a very modular and extensible approach described in this appendix. We describe the core functionality of GALE that does not make any assumption of the application area and can best be described as an (empty) adaptation shell. We also describe some standard built-in elements that make GALE directly suitable for popular applications, where there is a hierarchical domain model, content based on some form of XML, and where there is some type of linking for which we want to have adaptation. We then demonstrate how certain types of extensions can and have already been made and used to make GALE more suitable for different types of applications, thus bringing GALE closer to what the GAF model envisions.

This appendix presents in Section A.1 the global architecture of GALE. To show the generic aspects of GALE we concentrate on three aspects: the processor pipeline for selecting and adapting resources and links, the configuration through which you can completely alter GALE's behavior, and the event bus, services and extensions through which you can make GALE compatible with different adaptation languages. Section A.2 presents an application developed by a student to show the complementarity of the conceptual structure and the content of an adaptive application. We also list some other adaptive sites that were created by groups of students, with or without the graphical CAM authoring tool [39] developed in the GRAPPLE project and later renamed to the Course authoring tool. We also show some features of the adaptive PhD thesis written about, for and served by GALE [66].

A.1 The Architecture of GALE

Figure 14 shows the GALE architecture. We will concentrate on the purple part (bottom left) which deals with the adaptation to a resource and some of the blue part (top part of the right side) which deals with internal and external services to handle adaptation formats and user modeling services. If you are reading this appendix sequentially so far and you prefer an example-based description you should read Section A.2 first. Our description focuses on the aspects of GALE that illustrate its flexibility and its extensibility.

The green part of Figure 14 handles the user logging in and GALE setting up a session and loading some information into a cache for the Domain Model (DM) and User Model (UM). After logging in HTTP requests for concepts can be processed.

Let us first consider the overall process of GALE handling a request from a user (which typically comes in as a request for a URL, sent from the browser through HTTP). We can visually represent this process as shown in Figure 15. The process involves interaction with the Domain Model (DM) of an application (describing the topic domain and adaptation) and the User Model (UM) which holds all the information GALE knows about each individual user. (Note that the use of colors is just for clarity. There is no correspondence between colors in figures 14 and 15.) Throughout this work we will often refer to the configuration of GALE which is

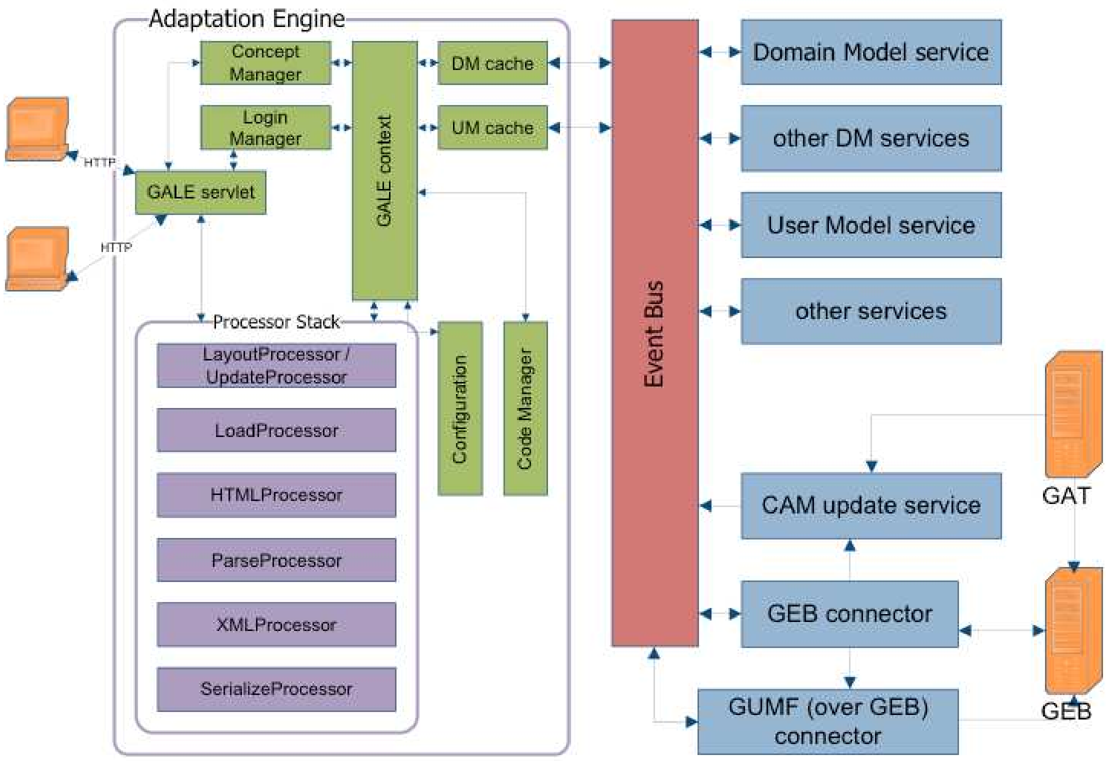


Figure 14: The architecture of GALE [67]

done through a file we call galeconfig. It allows almost every aspect of the behavior of GALE to be changed to suit the needs of its users.

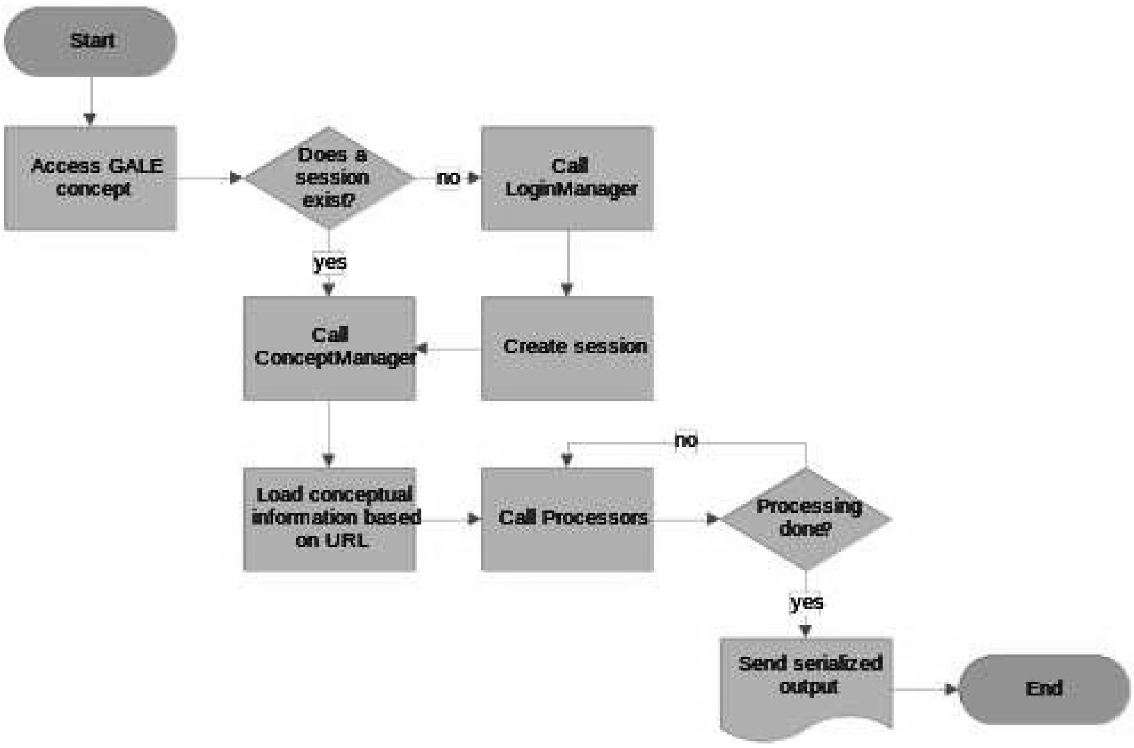


Figure 15: Handling a request for a concept [67]

1. If this is the first request the user sends since starting the browser no session will be associated with that request, so a session is initiated and a login procedure started. This is standard behavior for Web-based applications. GALE can work with several login managers, defined and enabled or disabled in *galeconfig* (see Section A.1.2) but for stand-alone GALE use the login follows the following multi-step procedure:

    (a) For a first request (without session information as there is no session yet) the user is still unknown. The login manager redirects to a servlet/page that prompts the user for a user id and password.

    (b) The user id is passed on to the UM cache, to request the application-independent part of UM for this user. Internally GALE refers to this as the user entity.

(c) Since the UM cache will not have cached the user model yet, it will communicate with the user model service through the event bus. The event bus makes it possible to distribute the processing load for adaptation and user modeling between different machines.

(d) UM is needed by the login manager to verify that the user has provided the correct password. Next the login manager (servlet) returns a redirect to the original URL. As a result the user's browser will request the same concept again, this time with session information. (This repetition of the request is invisible to the end-user.)

2. GaleServlet now calls the concept manager in order to find out how to handle the request. If the request is for a concept, the concept manager will determine the identity of the requested concept and retrieve the domain and adaptation data for the concept from the DM cache which may need to load it from a Domain Model service. (Here again it is possible to offload the DM service to a different machine.) If the URL does not refer to a concept it is handled differently (e.g. a file can be simply retrieved and served, as in the case of an image).

3. Handling the concept is a multi-step sub-process that uses processors (more or less in the top to bottom order as shown in the purple part of Figure 14). GALE can be extended with new processors that can be used anywhere in the processing pipeline. (We describe these extensions in Section A.1.3.) Section A.1.1 describes the default processor pipeline (the default configuration in galeconfig). The processor pipeline will typically load a resource, process it in memory, and at the end serialize it again and send the result to the user's browser as an HTTP response.

A.1.1 The GALE Processor Pipeline

The processing of a concept (request) is done through a series of processors (each one handling the output of the previous one). Galeconfig defines the list of processors and thus also the order in which processors are invoked. One can easily insert a processor into the processor chain (as shown in Section A.1.3) to add some functionality to GALE.

GALE uses its own language GAM to define an application's domain and adaptation model (DM). Section A shows some of that code. Here we use a subset of GAM which we refer to as GALE EL code (EL stands for Expression Language). This is code to either evaluate an expression over information from DM and UM or to update values in UM. UM is the main information source for the personalization (or adaptation) offered by GALE. We sometimes use a property of a concept, which is a DM element, and sometimes an attribute, which is a UM element. The GALE EL code is in fact Java code in which some shorthand notation is used to refer to concepts, properties and attributes. The code manager is configured in *galeconfig*. In principle it is possible to write your own code manager to enable GALE to work with a different expression language, perhaps not based on Java but on another programming language. Section A gives several examples of GALE EL code. Here we just give a bare minimum introduction:

- ${#suitability} refers to the value of the suitability attribute of the current concept (# always refers to a attribute).
- ${#image?title} refers to the title property of the image attribute of the current concept (? always refers to a property, whether a property of a concept or of an attribute). While using GALE the value of an attribute may change but properties of an attribute cannot change. (The properties are defined in the DM.)
- ${->(parent)?type} refers to the type property of the parent of the current concept (following the parent relation). Again, this is defined in the DM: properties and relations cannot be changed on the fly.
- ${#visited} = ${#visited}+1; is a statement that assigns the result of ${#visited}+1 to the visited attribute of the current concept. It thus simply increments the visited attribute by 1.

Note that the properties, attributes and relations used to illustrate the syntax are not predefined in GALE. The "beauty" of GALE is that it only defines that there can be concepts with properties, attributes (possibly also with properties) and relations and does not say which concepts, properties, attributes or relations should exist. However, to make GALE more easily usable from the start there are some "additions" that are included in the GALE distribution that do make such assumptions. For instance there is a navigation accordion menu available (static-tree-view) that presents part of a concept hierarchy, assuming that parent relations exist to define that hierarchy. Also, the *LoadProcessor* we describe below expects the existence of a resource attribute in order to decide which file to load.

We now explain the processors of the pipeline, mostly in the order they are called (which is the top to bottom order in Figure 14).

1. The first processor that is called is the *UpdateProcessor*. It signals an *EventManager* that the *access concept* event has occurred. The default *EventAccessHandler* executes the event code of the concept as defined in DM. Typically this event code will signal some UM update to the UM cache. For instance, the number of visits to the concept may be incremented. UM cache communicates over the event bus with the UM service. Event code associated with UM attributes may prompt the UM service to generate more UM updates. The visit to a concept may signal increasing interest in that topic and related topics or may indicate an increase in knowledge of the concept. The resulting changes to UM are posted on the event bus, and as a result are integrated in the UM cache. Important to note here is that 1) UM updates are calculated before generating adaptation (a design choice that corresponds to previous behavior of AHA! and motivated and explained in [26]), and 2) UM updates come from both the *UpdateProcessor* (in the adaptation engine) and from (event code) rules executed within the UM service. This distributed execution of UM updates is essential in GALE as GALE can also be used together with external UM services, including the GRAPPLE user Model Framework GUMF [1]. UM updates resulting from a concept access are handled synchronously (the adaptation engine waits for a response from the UM service). However, should an event access result in a message being sent to an external UM service such as GUMF it is handled asynchronously (so the adaptation engine will not wait for a response but will handle the UM update whenever it comes in). Figure 16 shows the process of UM updates in a more graphical way.

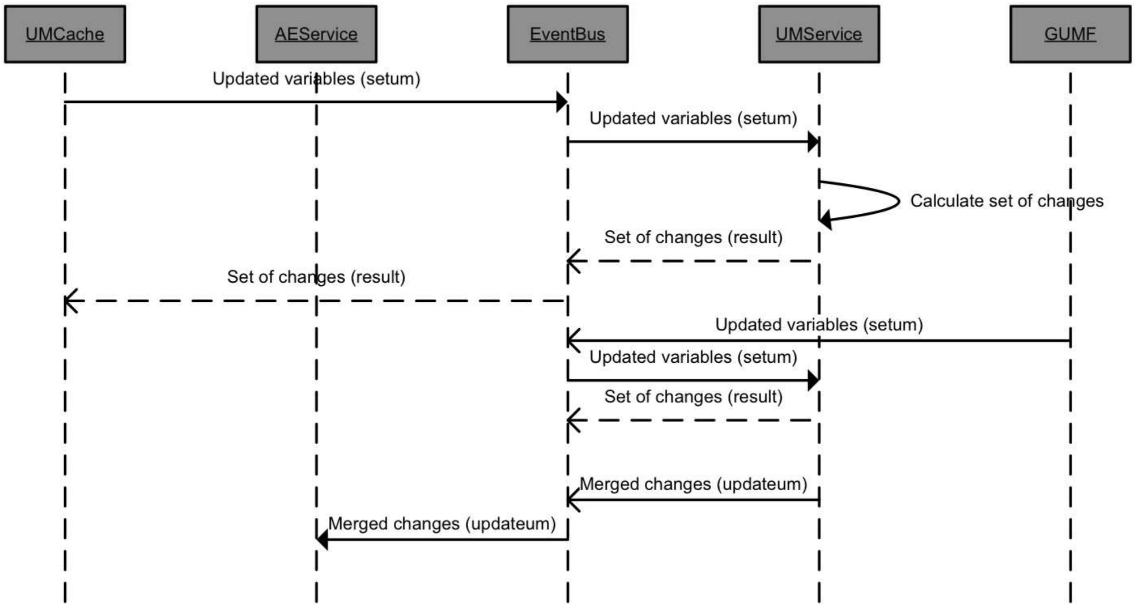


Figure 16: User model updates being handled by the adaptation engine and UM service(s) [29]

2. After the UM updates have been performed the *LoadProcessor* will retrieve the actual resource (file) associated with the concept. The name of the resource is found in the resource attribute of the concept. This "name" may refer to a local file or a file to be retrieved from some server using http. It may also be a (GALE EL code) expression over DM and UM to "compute" the name of the actual resource. An *InputStream* is opened so that a subsequent processor can load and process the data. File name extensions are used to determine the mime type of the resource.

3. Optionally the *LogProcessor* then adds an entry to a global log file (access.log by default). The id of the user, date, request, referrer (that may be present in the HTTP request), the name of the requested concept and the resulting resource are logged, for possible later analysis. Note that in order to log the resource that is retrieved and adapted the *UpdateProcessor* must run before the *LogProcessor*. An implication of the chosen order is that the *LogProcessor* cannot log any user model state information that existed before the *UpdateProcessor* performed its updates. In Section A.1.3 we show how to extend GALE to perform such additional logging if desired.

4. If the mime type of the resource is HTML (not XHTML) the *HTMLProcessor* uses the (open source) *Tagsoup*[9] converter to convert the file to XHTML. The new *InputStream* now contains valid XHTML.

5. If the input (after step 4) is XML (also XHTML) the *ParseProcessor* converts it into an in-memory DOM tree, using the open source *dom4j*[10] parser. We will not describe the processing of non-XML input further, but in *galeconfig* it is possible to insert processors for other data types if desired. (Such processors of course need to be developed as the GALE distribution only contains a processor for XML.)

6. The *XMLProcessor* walks through the DOM tree in order to perform adaptation where needed. The modules that may be used to perform adaptation to certain tags are loaded by the *XMLProcessor*. Which modules exist and to which XML tags they are applied is configured in galeconfig. Modules are provided to handle "if" tags, "object" tags, links, variables, and more. We explain a few modules in next section. Adding new modules to handle different tags, possibly in different XML formats, is relatively easy. Within the GRAPPLE project the addition of modules has been investigated for device adaptation and for adaptation to virtual reality [71].

7. Optionally, the *LayoutProcessor* generates a frame-like structure using tables, by creating an (in-memory) XML document that contains the views (any class that implements the *LayoutView* interface) embedded in a table that defines the layout. This XML document has a placeholder element where the actual content should be. A *CSSLayoutProcessor* is also available that uses css and div sections to lay out the browser screen. A *FrameLayoutProcessor* can also be used that uses an iframe element to display the actual content. The different processor choices make it possible to combine GALE with many other presentation structures (e.g. with portals or learning management systems). For simplicity we do not describe the differences between these processors in detail, nor their actual position in the pipeline (which also differs).

[9]See ccil.org/ cowan/XML/tagsoup/ for more information on tagsoup.
[10]See dom4j.sourceforge.net for more information on dom4j.

8. When the DOM tree is adapted the *SerializeProcessor* generates the textual XML representation and presents that to *GaleServlet* as an *InputStream*. *GaleServlet* uses this to generate the HTTP response to be sent back to the user's browser.

For resource types that do not have specific processors associated with them, *GaleServlet* will create the final *InputStream* itself in order to send the content back to the browser as an HTTP response. This happens for instance with images embedded in HTML pages. It is also possible to "bypass" the processor pipeline to perform completely different processing and generate a result "page". This is done by means of a *Plug-In*. Examples of predefined plug-ins are the *Form* plug-in that allows input from forms to be used to execute UM updates, the *Password* and the *Logout* plug-in with obvious functionality. A typical "extension" of basic GALE functionality is a plug-in to evaluate multiple-choice tests for use in educational applications. (This is not a real extension as it is already included in the GALE distribution.)

A.1.2 GALE Flexibility through Configuration

GALE was developed as part of the GRAPPLE EU project. GRAPPLE aimed at integrating open source or commercial Learning Management Systems (LMS) with Adaptive Learning Environments (ALE). The project aimed at making adaptivity available to a broad audience (mainly in technology-enhanced learning). To this end the adaptation engine needed to be made very generic and extensible, because developers of adaptive applications are expected to have special desires for adaptation functionality we might not foresee. Within the project itself adaptation in simulation and adaptation in virtual reality were already considered, as well as device adaptation. Different LMS may require a different way of embedding GALE output in their presentation. The different layout processors make this possible. To make GRAPPLE usable with many different LMS and to support life-long learning a repository of user-specific information was needed (the GRAPPLE User Modeling Framework, GUMF [1]). Different LMS store information in GUMF, and Section A.1 already explained how input from such an external UM service is captured by GALE.

The integration process between LMS and GALE is described more in detail for instance in [72].

GALE uses the Spring[11] "inversion of control" container to configure and instantiate all components. Without going too much in detail we describe the most important configuration elements here (omitting small things like where the access log file or some other files are stored or at which address GEB and GUMF are located). The GALE configuration is stored in the *galeconfig* file galeconfig.xml (in Tomcat?s webapps/gale/WEB-INF directory). What the configuration mostly does is associate names that have meaning as functional parts of GALE with a Java class name or Bean (implementing the functionality) and it also defines which properties configure the behavior of that functional part. We look at some parts in detail below.

1. The *processorList* defines processors that handle a request and adapt resources. The main processors have been mentioned in Section A.1 already. Here we look specifically at two processors: *XMLProcessor* and *PluginProcessor*.

    (a) The *XMLProcessor* performs transformations (for adaptation) to the DOM tree of an XML resource. *Galeconfig* contains a list of Modules that handle specific XML tags. Adding adaptation to a new tag can be done by creating a new module and associating it with the tag (and its namespace) in this list. We only present a small selection from the modules that are included in the GALE distribution:

    - The *IfModule* handles the <if> tag. It expects <if> to have an argument "expr" that is a Boolean expression in GALE code. It expects one or two child elements: a <then> and optionally an <else> element. The module replaces the <if> subtree by either the content of the <then> subtree or the <else> subtree. The *IfModule* thus realizes what is known as the *conditional inclusion of fragments* technique [49].
    - The *AdaptLinkModule* handles the <a> tag which is used just like the HTML <a> tag, but referring to a concept, not a page or

[11]See www.springsource.org for more details about Spring.

resource. GALE (actually the *LoadProcessor*) decides which page to retrieve and return based on the resource attribute of the concept. An optional exec argument can be used to associate a (UM) action with following the link. For instance: exec="${tour#start}=true;" could be used to say that following this link indicates the start of a tour, whereas accessing the same concept (tour) through other links does not. GALE supports the adaptive link annotation (and hiding) techniques through link anchor colors and through icons in front of and behind the link anchors. While this can be defined in *galeconfig* (for all adaptive applications running on the same server at once) one can define for each concept an attribute #link.classexpr to conditionally define a link class (corresponding to a link color in the application's stylesheet) and an attribute #link.iconlist that determines under which conditions (over DM and UM) which icons are placed. Any GALE EL expression can be used to conditionally generate the filename (URL) of an icon (or the name of the link class). Typical use of icons can be found in e.g. ELM-ART [7] and Interbook [12], using colored balls to indicate the suitability of a link and checkmarks to indicate the knowledge level of the link target concept. Typical use of link colors uses blue for recommended links, purple for suitable but already visited links and black for non-recommended links that appear hidden amongs black running text. But any combination of colors is possible and e.g. the adaptive PhD thesis [66] also uses orange and dark purple. The choice is completely up to the application designer.

- The *ObjectModule* inserts either a file specified through the data argument or a concept specified through the name argument. The <object> tag is preferred over the <if> tag when the same fragment needs to be conditionally included in many different pages. With data="header.xhtml" we can insert a header file in a page, whereas with name="programming" we insert whichever resource is associated with the concept programming (possibly involving evaluating expressions to select that resource).

- The *VariableModule* inserts either the value of a UM attribute or the result of a GALE expression in the page. <variable name="#visited"> for instance can show the number of visits to the current concept; <variable expr=${#visited}> does the same, but now as expression.
- The *AttrVariableModule* is similar but inserts its result in the surrounding element's tag. <img><attr-variable name="src" expr="${?image}"><img> uses the value of the image property of the current concept as the source (url) of an image to insert. Note that because of syntax restrictions of the XML language we could not use ¡variable¿ inside the <img> tag itself. (Arguments of an XML element cannot contain XML elements.)
- The *ForModule* repeats a fragment of XML for elements in a list.

```
<for var=‘‘concept’’ expr=‘‘${<-(parent)}’’>
<variable expr=‘‘${%concept?title}’’/><br/>
</for>
```

  This example inserts a list of the titles of the children of the current concept.
- The *PluginModule* generates a link to a plug-in. (It does not perform the plug-in code itself as this is done by the *PluginProcessor* described below. <plugin name="logout">Logout<plugin> results in a link through which the user logs out. Note that because GALE transforms <plugin> into a link anchor the description of the plugin (the word "Logout" in the example) must not contain an <a> tag but can contain other HTML tags if desired.
- The *MCModule* is a nice example of how the "core" functionality of GALE can be extended to serve specific types of applications. This module (which we associated with a <test> tag in our applications) uses subelements that are called <question> and <answer> (with different parameters and content) to generate a multiple-choice test. We used this in a setup where we wrote more questions and answers

than are needed and the module selects a random subset of questions and answers (making sure to include correct answers of course). As the multiple-choice test is generated by a module it is embedded in an (XHTML) page that can contain any other content (and links) as desired. Also, the content of each question and answer can contain arbitrary tags as well.

(b) The *PluginProcessor* handles plug-ins that perform some function and then generate “complete” output. Examples of plug-ins are the password and logout plug-ins (to change the user’s password and to log out) and exec to execute some GALE code and show the result (used mostly for debugging purposes). To facilitate educational applications we added the mc plugin to evaluate a multiple-choice test generated by the *MCModule*.

2. The *hibernateDataSource* bean controls how GALE stores data (including DM and UM). As we use Hibernate[12] GALE is independent of the database backend used. We have used two storage methods so far: *hsqldb* which stores data in a simple text format and *mysqldb* to store data in a MySQL database. MySQL (or any other real database like Postgres or Oracle) is recommended for a real server installation. In that case it is possible to use a separate machine for the database to offload the GALE server.

3. Figure 14 shows that GALE uses an event bus through which the core GALE engine communicates with DM and UM and other services. Two implementations of this bus exist: one (*LocalFactory*) that uses method calls and thus requires all services to reside on the same server and one (*SOAPFactory*) that uses SOAP and can handle DM and UM services that run on different machines. The event bus is configured to communicate with a number of services. (This bus uses the Publish/Subscribe method.) The DM and UM services are most obviously needed, but several other services exist to implement compatibility of GALE with different authoring formats and tools for adaptive application, as described in the next section.

[12]See www.hibernate.org for more details on this Java persistence framework.

4. (13) The *loginManager* bean defines how GALE users can identify themselves. The *DefaultLoginManager* presents a simple form for username and password (explained in Section A.1). The *LinkLoginManager* allows LMS users to automatically log in on GALE using the id they have on the LMS. The LMS uses a "secret" key to identify itself to GALE and essentially "promises" that the user is who (s)he says (s)he is. The *IdPLoginManager* makes use of Shibboleth[13] to make GALE usable in a federation of institutes/companies that share a single sign-on facility. The *OpenIdLoginHandler* allows the reuse of an already existing id in any service using Open ID[14].

5. The *codeManager* defines which language of adaptation code is used. All examples of GALE code used in this paper use GAM: Java with some shorthand to refer to DM and UM values, but it is possible to use very different code provided that a new code manager is developed. Note that in GALE there are actually two code managers (slightly different): one in the adaptation engine (dealing with concept event code and GALE expressions used in GALE XML tags) and one in the UM service (dealing with UM attribute event code).

6. (15) The *configManager* is a wrapper for handlers of different types of configuration: for processors, for link adaptation (with icons) and for presentation. The latter (*Presentation-Config*) defines which automatically generated parts can exist in the presentation of adapted pages. These parts are called views. In order to create a view it may be necessary to make some assumptions about the presence of specific properties or attributes. The GALE distribution comes with some predefined views: the *static-tree-view* which displays a menu over the hierarchy of concepts of the current application, based on the assumption that there are *parent* relations to define that hierarchy (see Figure 21), the *next-view* which generates a link to the next concept in a guided tour (see Figure 20), and the *file-view* which inserts (and adapts) a file with a fixed name. The file-view can used to include a header and/or footer for instance (see Figures 20 and 21) instead of including an <object> tag.

[13]Consult the Shibboleth Consortium site at shibboleth.net for more details about Shibboleth.
[14]Consult the Open ID Foundation site at openid.net for more details about Open ID.

A.1.3 Extending GALE: from generic towards general-purpose adaptation engine

As we already showed in the previous (sub)section the generic core of GALE needs to be extended in order to obtain the desired functionality for a specific type of application. GALE can be extended in extreme ways, like by adding a new code manager. GALE was created to be a general-purpose adaptation engine in the sense that its functionalities allow authors to use (or at least emulate) a great variety of adaptive techniques without the need of extensions. But by having a generic core and modular design, GALE can be extended to be used for other purposes, i.e., for different types of adaptive applications that require functionality not yet present in the core, e.g. for a recommender system. The most common (and easiest) ways to extend GALE include the following types of extensions:

1. Adding modules: The standard GALE distribution comes with modules for conditional inclusion of fragments (the *IfModule* and *ObjectModule*), for link adaptation (the *AdaptLinkModule*), for presenting information calculated from the DM and UM (the *VariableModule* and *AttrVariableModule*), for generating lists (the *ForModule*) and counting DM or UM elements (the *CountModule*) and for including more complex content parts generated from DM and UM by means of views (the *ViewModule*). Specific applications may require adapting different XML tags in specific ways. We already showed the existence of the *MCModule* for multiple choice tests. GALE also has a *TextModule* for handling input in the Creole[15] syntax used in Wikis. In the GRAPPLE project modules were added to handle virtual reality input described in the X3D language (see [71] for details).

2. Adding views: Views generate content from DM and UM or any other source and are included in the presentation through the <view> tag (or any other tag associated with the *ViewModule*). Figure 21 shows the *static-tree-view* that produces an accordion menu over the hierarchy formed by parent relations. The link colors and colored balls are chosen based on the value of a suitability attribute in combination with a visited attribute. Students have experimented

[15]See wikicreole.org for more details on the WikiCreole language.

with different variations of the *static-tree-view* including one that shows a progress bar for the top n (configurable) levels in the concept hierarchy.

3. Adding plugins: Instead of just "adding" or "altering" something in a given resource you sometimes may wish to generate a complete presentation, e.g. as a response to completing a form (for which GALE offers the *FormPlugin*). A special case is the *MCPlugin* for multiple-choice quizzes which returns a score, and the *ExecPlugin* which executes arbitrary GALE EL code. When creating a new plug-in you should be aware that after the *PluginProcessor* has run all subsequent processors are skipped. (This guarantees that GALE cannot mess up what the plug-in has generated.)

4. Adding processors: A nice example of a reason to add a new processor was found in the research described in [60] and presented in Chapter 3. In this work the effect of link annotation is researched by checking whether users follow the advice given through link annotation or not. To study this in the hypermedia course taught at the TU/e the user sessions needed to be replayed (from the available log) because the annotation state of links (when followed) was not logged. To avoid this we wished to add this state to the information being logged by GALE. However, this state is computed form the user model, and the *UpdateProcessor* changes the basis for this computation before the *LogProcessor* runs. The solution was to add a processor that evaluates the annotation state and stores it in a user model attribute not touched later by the *UpdateProcessor*. The *LogProcessor* can then log that state for later analysis by researchers. Indeed, we used this data in the evaluation presented in Chapter 4.

5. Adding services: Before GALE numerous other adaptive systems have been developed. It would be a shame not to be able to reuse them in GALE, given that GALE is very generic and should be able to emulate (or be extended in order to emulate) the behavior of these other systems. Such emulation was already performed before in [24] where the AHA! system [26] was used to emulate Interbook [12]. To import applications in other formats one can develop new DM services to be placed on the event bus. The main approach here is that the new service should only be used to retrieve the domain model

(conceptual structure) and the adaptation rules for an application, not the content (pages) of an application. The GALE distribution includes services to accept applications defined in the CAM or Course Editor [39] developed in the GRAPPLE EU project and also applications in the format used by AHA! (version 3). Besides the GAM format GALE also supports an XML representation of its applications: the GDOM format. There is a service to read GDOM definitions and an export plugin that can generate a GDOM representation for any application (independent of which format was used to import the application). It is possible to develop compilers (to use in a service) for formats like Interbook, Scorm, and others, but this has not yet been done until now.

A.2 Examples of GALE Applications

This section presents an example-based introduction to (and motivation for) GALE. It reports on applications that were realized in GALE and on student projects to create applications and GALE extensions. In Chapter 6 we evaluate part of the applications developed by the students and presented in this appendix. An important remark is that this section can be read without first reading the previous section.

We will not describe how to define an overall adaptation strategy or how to design the adaptation based on pedagogical relationship types such as prerequisites. This is the subject of GRAPPLE's (CAM or Course) authoring tool described in [39].

GALE supports two approaches towards creating the content of an application, represented in Figures 17 and 18 (taken from [59]).

Figure 17 shows an author writing complete pages. This is a viable option for authors who wish to create adaptive applications without writing (or even seeing) any GALE code. This approach is well suited for an application in which pages do not have a common structure. The hypermedia course 2ID65 at the TU/e was originally created in this way and the GRAPPLE tutorial (at http://gale.win.tue.nl/) as well. This is also a good approach for making existing

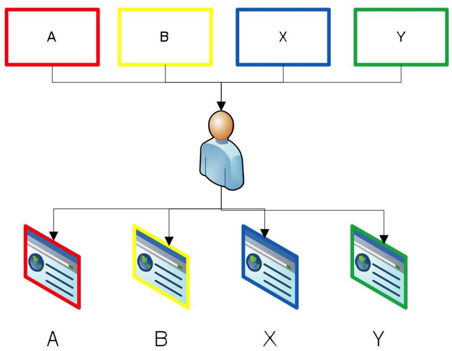


Figure 17: Creating pages separately [67]

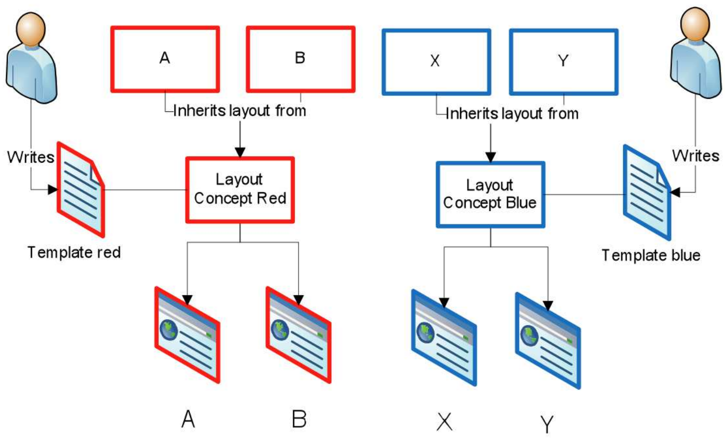


Figure 18: Authoring through template pages [67]

applications adaptive, for instance applications generated from databases (possibly from a content management system or a wiki).

Figure 18 shows an example where a cluster of pages share the same structure. (A and B are alike, X and Y are also alike.) When pages are created separately one needs to be careful to replicate the style on all pages that should be similar, and changing the style involves changing all these pages individually. Therefore Figure 18 shows the use of one template page for each set of pages that should look alike. The templates are used to (virtually) create individual pages by indicating which bits to use where in the template. Changing the presentation of all pages that use the same template requires a single file to be edited. Figure 19 shows the presentation of a concept from the application Milkyway that was created by a master student at the TU/e [59] and that has frequently been used to illustrate GALE. All pages of Milkyway have the same look and feel, not only with the header and footer (and a navigation menu that we cut off) but also within the main content part of the page. Milkyway deals with stars, planets and moons, and has slightly different templates for each of these types of celestial bodies. Below we explain the template used for presenting planets.

In a GALE application every *concept* can have an associated *layout*. In Figure 19 the header and footer are defined as part of that layout. The "main" part of the presentation is the page. The layout can be shared between concepts but can also be different for each concept. It can also be adaptively changed but we do not expect this to be common. There are different ways to realize this layout (within HTML): using frames, tables, iframes and simple "divisions" that are positioned based on a CSS style sheet. All possibilities are supported by GALE through different *layout processors*. You can see that the presentation in Figure 19 consists of multiple parts because the scroll bar does not cover the header and footer. In the figure you can see that the example page contains a *title* (Jupiter), a *typology* (Is Planet of: Sun), an *image* with *title*, an *information fragment* and a *list of links to children* (The following Moon(s) rotate around Jupiter.) Below we show how each part is defined, to give you a feel for creating pages containing GALE tags and GALE code. This shows that pages can be partly "constructed" out of DM and UM elements rather than being purely "authored". All DM properties that are used (like "Jupiter",

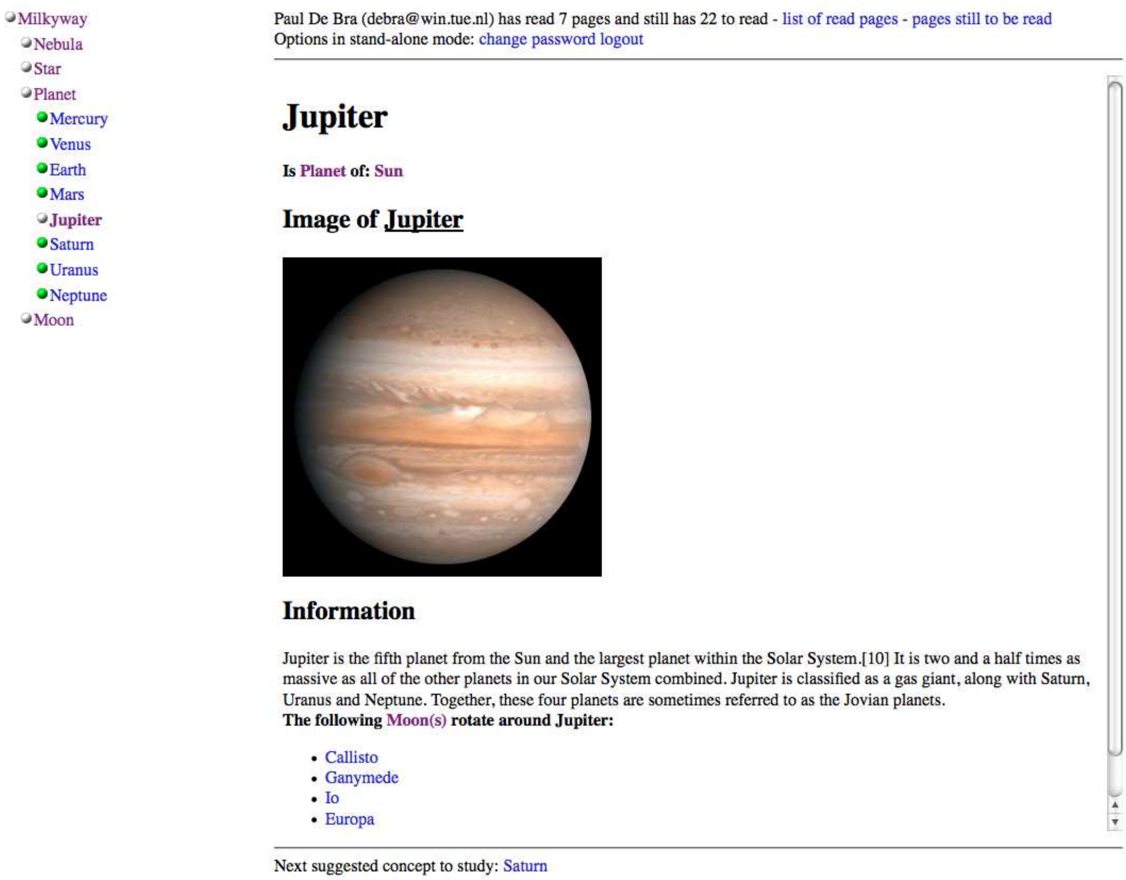


Figure 19: Example of a page based on a template.

"Sun") and relations (like "*parent*" and "*isPlanetOf*") have been created using the graphical GRAPPLE authoring tools (the Domain editor to be exact).

- In the header we see the name and email of the user. The code <variable name="gale://gale.tue.nl/personal#name"> and <variable name="gale://gale.tue.nl/personal#email"/> extracts that information from UM and inserts it in the page. (The "gale:" pseudo-protocol written in what looks like a URL tells gale to retrieve information from the "GALE Context" and is easier to understand than the complete Java method call to retrieve the name or email. That method call could of course also have been written here as GALE code syntax is Java with shorthand to retrieve DM and UM information.)

- The title "Jupiter" is generated by means of the code <variable expr="${?title}"/> which inserts the title property of the current concept in the page. As shown in Section A.1 a # sign refers to a user model attribute and the ? sign refers to a domain model property.

- The "Is Planet of: Sun" part is a bit more complex. "Planet" is actually the title of the parent concept and "Sun" is the title of the concept to which Jupiter has an *isPlanetOf* relation. <a><attr-variable name="href" expr="${->(parent)?title}"/> generates the link anchor tag. Calculating the value for the "href" argument cannot be done inside the <a> tag because XML tags are not allowed to appear inside another XML tag so it is done using <attr-variable>. This is similar to <variable> but inserts its output into the parent xml tag, in this case the <a> tag. The anchor text is <variable expr="${->(parent)?title}"/> (which generates the word "Planet"). The link to "Sun" uses <a><attr-variable name="href" expr="${->(isPlanetOf)?title}"/> <variable expr="${->(isPlanetOf)?title}"/>. Like with "Planet" the same expression appears twice because "Sun" is the anchor text as well as the link destination.

- To insert the image (we ignore the title here) we use the code <img><attr-variable name="src" expr="${?image}"/></img> where the name (URL) of the image is part of the domain model (created with the graphical Domain editor). As you see the attr-variable tag can be used in combination with any other tag to specify an argument used in that other tag (the parent tag in the XML document).

- The information fragment is stored in a separate file and included in almost the same way as the image:

```
<object><attr-variable name="data" expr="${?info}"/></object>.
```

  The name of the file is part of the domain model.

- The list of moons is generated using the <for> tag. (*Milkyway* additionally checks whether a planet has moons to avoid generating an empty list if it doesn't.) We only show the code for the list (not for "The following moons rotate?")

```
<ul><for var="concept" expr="${<-(isMoonOf)}">
   <li><a>
      <variable expr="${%concept?title}"/>
      <attr-variable name="href" expr= "&quot;%concept&quot;"/>
   </a></li>
</for></ul>
```

  All concepts with an *isMoonOf* relation to the current concept are associated, one by one, with the variable "concept". A bullet list is generated with for each such concept a list item with a link to the concept and with the title of the concept as link anchor text.

The Milkyway application has a "Course model" in which *prerequisites* are defined. For instance, a planet should be studied before studying its moons becomes recommended. Figure 20 shows the top part of the page about the concept "Moon". This time we have not cut off the navigation menu. In this menu we see link annotation (with link colors and with colored balls) to indicate that only the four moons of Jupiter are recommended (as we only studied the planet Jupiter and no other planet that has moons).

The link annotation shown in Figure 20 uses the same "good", "neutral" and "bad" link classes (blue, purple, black) that were previously used in some AHA!

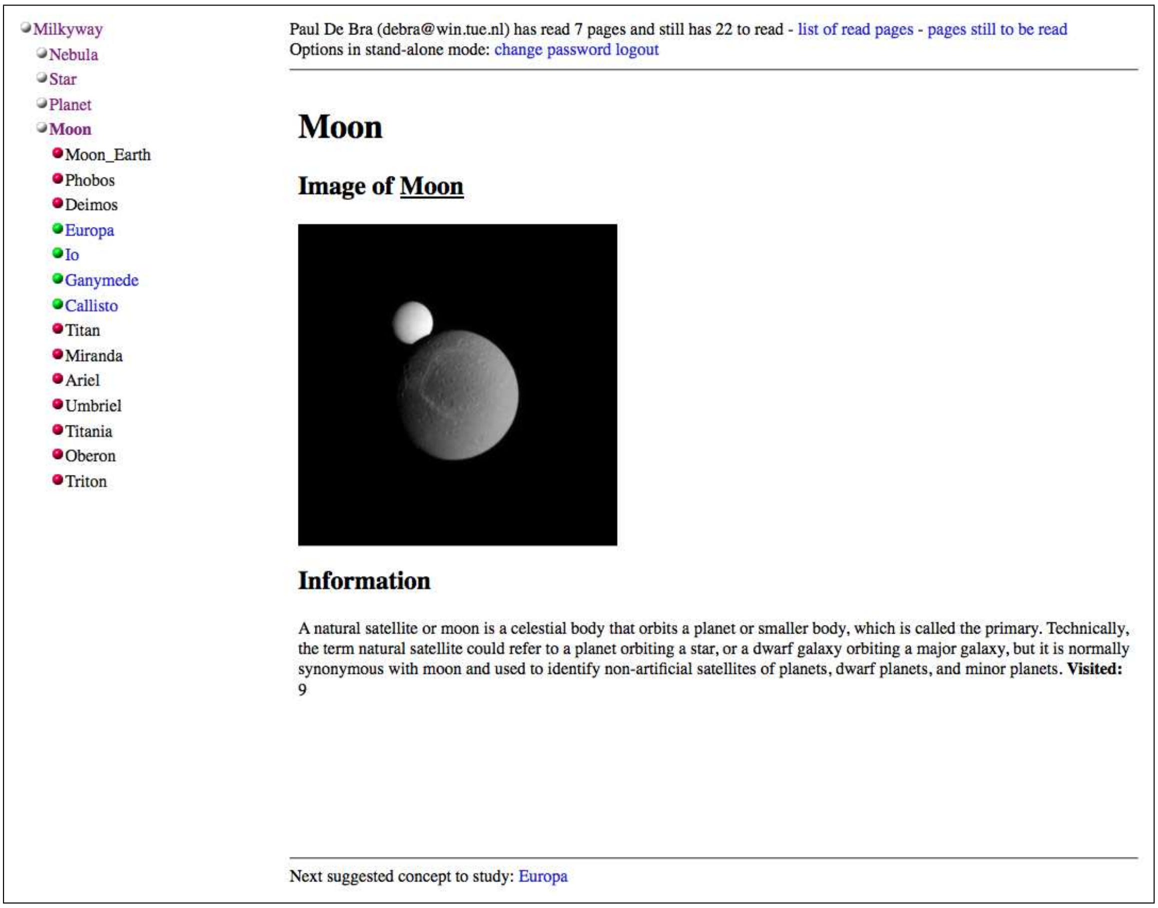


Figure 20: Link adaptation based on prerequisites.

applications, as we can see in Chapter 3 and 4. In GALE (and in fact already in AHA! as well) you can define arbitrarily many link classes and arbitrary conditions for deciding which class a link should have. The default class and color scheme are suitable for the default adaptation (or pedagogical) models created using the GRAPPLE authoring tools, described in [39] for instance. The graphical authoring tools hide the complexity of the GALE adaptation rules from authors, but they do enable advanced authors to create their own arbitrarily complex rules, specified in the Generic Adaptation Model language (GAM).

In April 2012 David Smits defended the world's first adaptive PhD thesis, about GALE, and served by GALE [66]. The domain model and adaptation rules for this thesis were written directly in GAM and the layout was defined using CSS (cascading style sheets). Figure 21 shows a screen shot from the thesis. It illustrates some interesting style differences between the thesis and the *Milkyway* application and shows some extensions to the core GALE functionality.

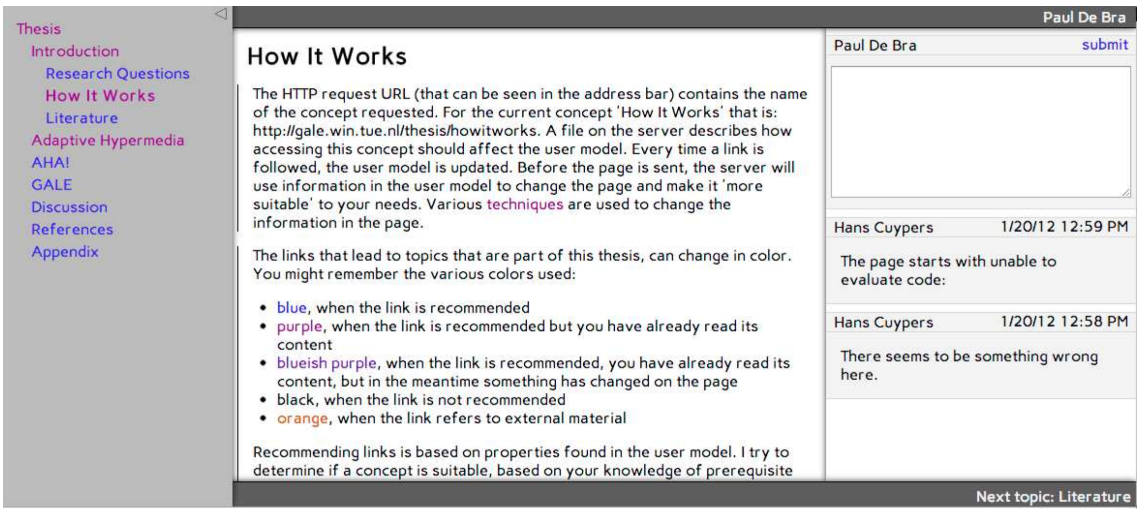


Figure 21: Screenshot of the adaptive PhD thesis about and served by GALE.

This "How it works" page from the thesis explains how the link adaptation makes use of five different link colors with a different meaning. It shows vertical lines to indicate the parts of the current page that are different from what was shown on this page during the previous visit. (So there not only is content adaptation but also some help to the end user to see which parts have been adapted.) The screenshot also shows an optional comments field where the reader can enter comments and see all users' comments on this page. Only the author of the application can delete comments. There is some adaptable functionality as well: the menu on the left can be closed (by clicking on ◁), the comments part can be hidden, some stereotype user profile can be selected ranging from layperson (e.g. family reading the thesis) to expert (e.g. review committee member), and the profile can also be reset to start over and try different adaptation. The thesis also has layout that automatically adapts to the device used by the reader, for instance moving the menu to the top when a narrow-screen smartphone is used.

The above description shows that it is possible to create very different types of application and presentation in GALE, making it truly "general purpose" as an adaptive Web-based hypermedia platform. At the TU/e some groups of students developed their own GALE applications, using any content they desired, to see whether the flexibility and extensibility of GALE can really be used by sufficiently technically skilled authors. All students were given the *Milkyway* application [59] and the *adaptive PhD thesis* [66] as examples. In 2011 most groups used the graphical GRAPPLE authoring tools [39] for defining the adaptation. In 2012 most groups used the GAM language directly, after receiving feedback from the 2011 students about the different authoring approaches. Chapter 6 presents an evaluation of these GALE applications developed by the group of students from the year of 2012.

Below we list the subjects of the applications the students developed, and also whether the graphical GAT authoring tool was used or the GAM language, whether page templates were used (instead of writing each page individually and completely) and whether a GALE layout manager was used or layout was done using CSS. In the table below the first 10 projects were from 2011, the latter from 2012.

All of these examples used adaptive navigation support through *prerequisite* relationships, some added *conditional inclusion of fragments*, some *stretchtext*, some used template pages, some used content from a wiki, some required specific handling of forms input (e. g. the restaurant menu and radio player). Contrary to what one might expect students preferred to use the GAM language over the graphical authoring tools. This may be explained by the type of students (computer science) who valued full control over ease of use of the authoring interface.

In 2012 master students created their own GALE extensions in order to see whether the claimed extensibility of GALE is actually achievable and usable in practice. We present an evaluation of these GALE extensions in Chapter 6.

App. Subject	Authoring Tool (GAT vs GAM)	Template pages used? (yes/no)	Layout (GALE/CSS)
Photography	GAT	yes	GALE Layout
Logic	GAT	yes	GALE Layout
Biomes of the world	GAM	no	CSS
Disney Virtual Tour	GAM	yes	CSS
Restaurant Menu	GAT	yes	CSS
Web Technology intro	GAT	yes	GALE Layout
Movies	GAM	yes	CSS
The “happy” Butcher	GAT	yes	CSS
TU/e Information System	GAT	yes	GALE Layout
HTML/CSS intro	GAT	yes	GALE Layout
Research Methods book	GAM	no	CSS
Relativity Theory	GAM	no	CSS
Elementary Electro Physics	GAM	no	CSS
C-programming	GAM	no	CSS
Radio Player	GAM	no	CSS
Food Recipes	GAM	no	CSS
Latex	GAM	no	frames
Currency Trading	GAM	yes	CSS
Linux Commands	GAM	no	CSS
Do-It-Yourself Tools	GAM	no	CSS
Sports Center	GAM	no	CSS

Table 12: Adaptive applications developed by (groups of students)

B GALE Users' Experiences: using

This questionnaire is part of Vinicius Ramos thesis research. If you have any question, information or complain, you can contact him directly by email: vfcr@cos.ufrj.br.

You are free to leave any question about you and your experience in blank, but we kindly ask you to fill in the questionnaire regarding your experience.

As it is part of a Ph.D. research, it will **NEVER** be used as an evaluation or an assignment of the 2IO80 - DBL Hypermedia Course.
All the information provided by you will be used in the research as an **anonymous person**, and nobody, except for the researcher, will have access to it.

About you

This section is related to your personal information and previous experience in computer.

1. **Gender?** ☐ Male ☐ Female
2. **How old are you?** I am ______ years old.
3. **How many hours per DAY do you spend working with computers?**
   - ☐ from 0 to 1 hour
   - ☐ from 1 to 3 hours
   - ☐ from 3 to 5 hours
   - ☐ more than 5 hours
4. **How many hours or minutes per WEEK do you spend (estimate) on the activities below? (please indicate if it is hours or minutes)**
   - ☐ Checking email: ____________________
   - ☐ Social Networks (Facebook, twitter...): ____________________
   - ☐ Reading news: ____________________

☐ Development (software): ____________________

☐ Other (please specify): ______________________________

5. **Please, mark with an X each programming language (framework / platform) you are used to:**

☐ PHP newbie ☐—☐—☐—☐—☐ expert

☐ JAVA newbie ☐—☐—☐—☐—☐ expert

☐ .Net newbie ☐—☐—☐—☐—☐ expert

☐ C++ newbie ☐—☐—☐—☐—☐ expert

☐ C# newbie ☐—☐—☐—☐—☐ expert

☐ Pascal newbie ☐—☐—☐—☐—☐ expert

☐ Python newbie ☐—☐—☐—☐—☐ expert

☐ Ruby newbie ☐—☐—☐—☐—☐ expert

☐ Other: _______________ newbie ☐—☐—☐—☐—☐ expert

Student Experience

This section is related to the "2IO80 - DBL Hypermedia" adaptive learning course offered by GALE.

6. **Is the "2IO80 - DBL Hypermedia" course your first experience with an adaptive course? If it is NOT, please, tell us more about your previous experiences.**

____

____

7. **How rich was the course interface? For Rich interface we mean: navigational help, specific social media, forum, question and answers, search, etc.**

very poor ☐—☐—☐—☐—☐ very rich

Comments:

____

____

8. **How difficult was the navigation?**
   very easy □—□—□—□—□ very difficult
   Comments:

   ____

   ____

9. **Regarding LINK ANNOTATION, where the system presents links in different colors, perhaps with additional icons, which of the following statements applies:**
   - □ I was not aware of link annotation.
   - □ I only notice that visited links became purple.
   - □ The link annotation was clearly intended to offer guidance through the course.
   - □ The link annotation was manly intended to avoid to going to some pages.
   - □ Other (please specify): ____________

Regarding LINK ANNOTATION, please, answer/evaluate the next few sentences.

10a. The link colors are responsible for guiding you throw the contents of the course.

□ For sure. □ Maybe. □ I don't remember. □ I am not sure. □ No, they are not.

10b. Regarding LINK HIDDING, where the main text is black, the black link is hidden in the main text but you can still click on it. Have you tried clicking on these links?

- □ Yes. Because I was curious to see what happens if I click on that.

- ☐ Yes. Because I would like to explore the course a little bit before I learn.
- ☐ Yes. Because the system presented a lot of black links and I would like to know what it means.
- ☐ Yes. But I am not sure why I clicked on that, probably to navigate throw black links and just that.
- ☐ Sorry. I don't remember.
- ☐ No, I have never.

Regarding MENU VIEW, where the system creates a tree view with links to the course's concepts. Please, answer the next sentence.

11. Did you follow most of the links from the menu view or the main (content) view?

☐ Mostly Menu View. ☐ I used both. ☐ I don't remember. ☐ Mostly Content View.

12. Regarding CONTENT ADAPTATION, which of the following statements applies:

- ☐ I was not aware of any content adaptation.
- ☐ I occasionally notice some minor changes when revisiting a page.
- ☐ The content of a page was radically different depending on prerequisites.
- ☐ The presentation seems to depend on learning or media preferences.
- ☐ Other (please specify): ______________________________

13. Regarding ADAPTATION, what was your user experience?

- ☐ I found it helpful.
- ☐ I hardly noticed it.
- ☐ I found it patronizing.
- ☐ Other (please specify): ________________

Regarding TESTS, where the system present randomized multiple choice questions.

14a. How were the questions presented

- ☐ All questions and answers in one page.
- ☐ Separate page for each question.
- ☐ Combination

15. Did you experience any error or problem during the course(s)?

____

____

16. Did you miss any feature, plugin or tool that you would suggest to be implemented in GALE system?

____

____

17. Regarding GALE system and its adaptation for end-users, do you have suggestions, comments or complains about it?

____

____

General

Use this space to write whatever you want: write about adaptive system, write about the courses you have participated, ask, complains or give suggestions for this questionnaire, or anything else you want.

____

____

____

C GALE Users' Experiences: using, authoring and developing

This questionnaire is part of Vinicius Ramos thesis research. If you have any question, information or complain, you can contact him directly by email: vfcr@cos.ufrj.br.

You are free to leave any question about you and your experience in blank, but we kindly ask you to fill in the questionnaire regarding your experience.

As it is part of a Ph.D. research, it will **NEVER** be used as an evaluation or an assignment of the 2ID55 - Adaptive Systems Course.

All the information provided by you will be used in the research as an **anonymous person**, and nobody, except for the researcher, will have access to it.

This questionnaire is divided in 4 parts: 1. about you; 2. student experience; 3. author experience; and 4. developer experience.

About you

This section is related to your personal information and previous experience in computer.

18. **Gender?** □ Male □ Female
19. **How old are you?** I am ______ years old.
20. **How many hours per DAY do you spend working with computers?**

    □ from 0 to 1 hour

    □ from 1 to 3 hours

    □ from 3 to 5 hours

    □ more than 5 hours

21. How many hours or minutes per WEEK do you spend (estimate) on the activities below? (please indicate if it is hours or minutes)

- ☐ Checking email: ____________________
- ☐ Social Networks (Facebook, twitter...): ____________________
- ☐ Reading news: ____________________
- ☐ Development (software): ____________________
- ☐ Other (please specify): ______________________________

22. Please, mark with an X each programming language (framework / platform) you are used to:

- ☐ PHP newbie ☐—☐—☐—☐—☐ expert
- ☐ JAVA newbie ☐—☐—☐—☐—☐ expert
- ☐ .Net newbie ☐—☐—☐—☐—☐ expert
- ☐ C++ newbie ☐—☐—☐—☐—☐ expert
- ☐ C# newbie ☐—☐—☐—☐—☐ expert
- ☐ Pascal newbie ☐—☐—☐—☐—☐ expert
- ☐ Python newbie ☐—☐—☐—☐—☐ expert
- ☐ Ruby newbie ☐—☐—☐—☐—☐ expert
- ☐ Other: _______________ newbie ☐—☐—☐—☐—☐ expert

23. Have you ever taken a distance learning or blended learning (online material and classes) courses on the Web?

- ☐ Yes.
- ☐ No.

If your answer is NO for the last question, please go to question 40.

24. Please, write the distance learning course name, the year and the provider (for example: 2ID65 - Hypermedia Course, 2002, TUe by Prof. Dr. P. De Bra). *If you do not remember any of this information just write what you have.*

____

25. What were your goals in most of the courses?

- ☐ **Learn a new programming language.**
- ☐ **Fulfill a required undergraduate course.**
- ☐ **Learn new software.**
- ☐ **Fulfill a required company (employer) course.**
- ☐ **Other:** ______________________________

Student Experience

This section is related to adaptive learning courses offered by GALE (or AHA!). If you never used GALE (or AHA!) as student, please, go to the next section.

26. How many courses (in GALE or AHA!) have you taken? ___________

27. How many courses (in GALE or AHA!) did you finish? ___________

28. In which way did you experience GALE (or AHA!) as student?

- ☐ Online only (no classes). How many: ________
- ☐ Blended (online and classes). How many: ________

29. How rich was the courses interface? For Rich interface we mean: navigational help, specific social media, forum, question and answers, search, etc. very poor ☐—☐—☐—☐—☐ very rich

Comments:

____

____

30. How difficulty was the navigation?

very easy ☐—☐—☐—☐—☐ very difficult

Comments:

____

____

31. Regarding LINK ANNOTATION, which of the following statements applies:

- □ I was not aware of link annotation.
- □ I only notice that visited links became purple.
- □ The link annotation was clearly intended to offer guidance through the course.
- □ The link annotation was manly intended to avoid to going to some pages.
- □ Other (please specify): ________________

Regarding LINK ANNOTATION, where the system presents links in different colors, perhaps with additional icons. Please, answer/evaluate the next few sentences.

32a. The link colors are responsible for guiding you throw the contents of the course.

□ For sure. □ Maybe. □ I don't remember. □ No, they are not.

32b. Regarding LINK HIDDING, where the main text is black, the black link is hidden in the main text but you can still click on it. Have you tried clicking on these links?

- □ Yes. Because I was curious to see what happens if I click on that.
- □ Yes. Because I would like to explore the course a little bit before I learn.
- □ Yes. Because the system presented a lot of black links and I would like to know what it means.
- □ Yes. But I am not sure why I clicked on that, probably to navigate throw black links and just that.
- □ Sorry. I don't remember.
- □ No, I have never.

Regarding MENU VIEW, where the system creates a tree view with links to the course's concepts. Please, answer/evaluate the next few sentences.

33. If there was a navigation menu, did you follow most of the links from the menu view or the main (content) view?

☐ Mostly Menu View ☐ I uses both or don't remember ☐ Mostly Content View

34. Regarding CONTENT ADAPTATION, which of the following statements applies:

☐ I was not aware of any content adaptation.

☐ I ocasionaly notice some minor changes when revisiting a page.

☐ The content of a page was radically different depending on prerequisites.

☐ The presentation seems to depend on learning or media preferences.

☐ Other (please specify): ____________________

35. Regarding ADAPTATION, what was your user experience?

☐ I found it helpful.

☐ I hardly noticed it.

☐ I found it patronizing.

☐ Other (please specify): ____________

Regarding TESTS, where the system present randomized multiple choice questions. Please, answer/evaluate the next few sentences.

36a. How were the questions presented

☐ All questions and answers in one page.

☐ Separate page for each question.

☐ Combination

37. Did you experience any error or problem during the course(s)?

____

____

38. Did you miss any feature, plugin or tool that you would suggest to be implemented in GALE system?

____

____

39. Regarding GALE system and its adaptation for end-users, do you have suggestions, comments or complains about it?

____

____

Author Experience

This section is related to authoring in GALE (or AHA!).

40. How did you create the course(s)?

- □ Graphical interface (with GRAPPLE Authoring Tools (GAT)).
- □ Text interface to create a GAM file (with notepad, for example).
- □ Both.

41. Why did you create the course?

- □ To fulfill the requirement of a(n) (under)graduate course.
- □ To understand how it works.
- □ To offer an adaptive course.
- □ Other: ________________________

42. How difficult was the authoring as whole?

very easy □—□—□—□—□ very difficult

Comments:

____

____

Please, evaluate the following aspects regarding the ease of creation:

43a. Concept very easy □—□—□—□—□ very difficult

43b. Concepts relationship very easy □—□—□—□—□ very difficult

43c. Adaptation rules very easy □—□—□—□—□ very difficult

43d. Installation and deployment very easy □—□—□—□—□ very difficult

43e. Test (quiz) very easy □—□—□—□—□ difficult

Please, let your comments about any aspect of the course's creation you want to tell us:

____

____

44. Which materials helped you and how much it helped you?

- □ Classes (explanation and lectures) nothing □—□—□—□—□ a lot
- □ Slides with annotations nothing □—□—□—□—□ a lot
- □ David Smits' thesis nothing □—□—□—□—□ a lot
- □ Experts help nothing □—□—□—□—□ a lot
- □ Other sources: ____________ nothing □—□—□—□—□ a lot

45. Did you miss any feature, plugin or tool that you would suggest to be implemented in GALE?

____

____

Developer Experience

46. How difficult was the development of GALE extensions?

very easy □—□—□—□—□ very difficult

Comments:

____

____

47a. What kind of functionality/feature did you implement?

- □ A NEW view.
- □ A NEW processor.
- □ A NEW module.
- □ A NEW plugin.

- ☐ Tests features.
- ☐ Authoring facilities/tool.
- ☐ Navigation improvement.
- ☐ Layout improvement.
- ☐ Other: ____________________

47b. Please, tell us what your functionality/feature does.

____

____

47c. Why did you implement this functionality/feature?

- ☐ You missed this implementation while you were using the system as a student.
- ☐ I saw something similar in another system and I would like to have the same thing in an adaptive course.
- ☐ I have just got a suggestion of a friend, colleague or teacher of the course I have applied.
- ☐ Other: ______________________________

Comments:

____

____

Please, evaluate the following aspects regarding the ease of implementation:

48a. Installation very easy ☐—☐—☐—☐—☐ very difficult

48b. Java package explore very easy ☐—☐—☐—☐—☐ very difficult

48c. Code comments very easy ☐—☐—☐—☐—☐ very difficult

48d. Clean code really bad ☐—☐—☐—☐—☐ really good

48e. Deployment very easy ☐—☐—☐—☐—☐ very difficult

48f. Test very easy ☐—☐—☐—☐—☐ very difficult

Please, let your comments about any aspect of the course's creation you want to tell us:

____

____

49. **Which materials helped you and how much it helped you?**
    - ☐ Classes (explanation and lectures) nothing ☐—☐—☐—☐—☐ a lot
    - ☐ Slides with annotations nothing ☐—☐—☐—☐—☐ a lot
    - ☐ David Smits' thesis nothing ☐—☐—☐—☐—☐ a lot
    - ☐ Experts help nothing ☐—☐—☐—☐—☐ a lot
    - ☐ Other sources: ____________ nothing ☐—☐—☐—☐—☐ a lot
50. **Regarding the development of functionality in the GALE system, do you have suggestions, comments or complains?**

    ____

    ____

General

Use this space to write whatever you want: write about adaptive system, write about the courses you have participated, ask, complains or give suggestions for this questionnaire, or anything else you want.

____

____

Curriculum Vitae

Vinicius Faria Culmant Ramos was born on 19-01-1983 in Porto Velho, Brazil. He obtained his bachelor degree in Computer Science in 2005 from the Federal University of Rio de Janeiro (UFRJ), Brazil. In 2008 he got a master's degree in System Engineering and Computer Science from UFRJ, under the supervision of Prof. Luis Alfredo V. de Carvalho, and then, at the end of the year of 2008, started his Ph.D. at the same institution under the supervision of Prof. Geraldo B. Xexeo. During his Ph.D., Vinicius did a 6-month internship in 2009 at TU/e working with Prof. dr. Paul De Bra in the area of Evaluation of Adaptive Systems. Vinicius returned for another internship at TU/e in the years of 2012 and 2013. This internship gave rise to further collaboration, whose results are being presented in this dissertation. Vinicius is currently a Ph.D. candidate for a joint-degree between UFRJ and TU/e. Nowadays, Vinicius is interested in the research and development of Adaptive Systems and in teaching programming languages to children.

SIKS Dissertatiereeks

1998
====

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations
within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

1999
====

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems

1999-7 David Spelt (UT)
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation

2000
====

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

2001
====

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,
Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a multiagent modeling and simulation language for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management: The Role of Mental Models in Business Systems Design

2002
====

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology;
Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and

Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

2003
====

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses

2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
Learning Search Decisions

2004
====

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining

2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)
Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

2005
====

2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM))
AI techniques for the game of Go

2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM)
Adaptive Game AI

2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web

Information Systems

2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments

2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry

2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU)
Usability of XML Query Languages

2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

2006
====

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU)
Building Web Service Ontologies

2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines

2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling

2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion

2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment, and the arts

2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change

2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN)
Aptness on the Web
2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlovic (UT)
Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

2007
====

2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU)
Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT)
Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM)
Context-Based Image Analysis

2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations

2007-19 David Levy (UM)
Intimate relationships with artificial partners

2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems

2007-24 Georgina Ramírez Camps (CWI)
Structural Features in XML Retrieval

2007-25 Joost Schalken (VU)
Empirical Investigations in Software Process Improvement

2008
====

2008-01 Katalin Boer-Sorbán (EUR)
Agent-Based Simulation of Financial Markets: A modular,continuous-time approach

2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations

2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach

2008-04 Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration

2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective

2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective

2008-07 Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning

2008-08 Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference

2008-09 Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective

2008-10 Wauter Bosma (UT)
Discourse oriented summarization

2008-11 Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach

2008-12 Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation

2008-13 Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks

2008-14 Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort

2008-15 Martijn van Otterlo (UT)
The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.

2008-16 Henriette van Vugt (VU)
Embodied agents from a user's perspective

2008-17 Martin Op 't Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises

2008-18 Guido de Croon (UM)
Adaptive Active Vision

2008-19 Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search

2008-20 Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven

2008-21 Krisztian Balog (UVA)
People Search in the Enterprise

2008-22 Henk Koning (UU)
Communication of IT-Architecture

2008-23 Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia

2008-24 Zharko Aleksovski (VU)
Using background knowledge in ontology matching

2008-25 Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency

2008-26 Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled

2008-27 Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design

2008-28 Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks

2008-29 Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans

2008-30 Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content

2008-31 Loes Braun (UM)
Pro-Active Medical Information Retrieval

2008-32 Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes

2008-33 Frank Terpstra (UVA)
Scientific Workflow Design; theoretical and practical issues

2008-34 Jeroen de Knijf (UU)
Studies in Frequent Tree Mining

2008-35 Ben Torben Nielsen (UvT)
Dendritic morphologies: function shapes structure

2009
====

2009-01 Rasa Jurgelenaite (RUN)
Symmetric Causal Independence Models

2009-02 Willem Robert van Hage (VU)
Evaluating Ontology-Alignment Techniques

2009-03 Hans Stol (UvT)
A Framework for Evidence-based Policy Making Using IT

2009-04 Josephine Nabukenya (RUN)
Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-05 Sietse Overbeek (RUN)
Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-06 Muhammad Subianto (UU)
Understanding Classification
2009-07 Ronald Poppe (UT)
Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-08 Volker Nannen (VU)
Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-09 Benjamin Kanagwa (RUN)
Design, Discovery and Construction of Service-oriented Systems

2009-10 Jan Wielemaker (UVA)
Logic programming for knowledge-intensive interactive applications

2009-11 Alexander Boer (UVA)
Legal Theory, Sources of Law & the Semantic Web

2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)
Operating Guidelines for Services

2009-13 Steven de Jong (UM)
Fairness in Multi-Agent Systems

2009-14 Maksym Korotkiy (VU)
From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)

2009-15 Rinke Hoekstra (UVA)
Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16 Fritz Reul (UvT)
New Architectures in Computer Chess

2009-17 Laurens van der Maaten (UvT)
Feature Extraction from Visual Data

2009-18 Fabian Groffen (CWI)
Armada, An Evolving Database System

2009-19 Valentin Robu (CWI)
Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets

2009-20 Bob van der Vecht (UU)
Adjustable Autonomy: Controling Influences on Decision Making

2009-21 Stijn Vanderlooy (UM)
Ranking and Reliable Classification

2009-22 Pavel Serdyukov (UT)
Search For Expertise: Going beyond direct evidence

2009-23 Peter Hofgesang (VU)
Modelling Web Usage in a Changing Environment

2009-24 Annerieke Heuvelink (VUA)
Cognitive Models for Training Simulations

2009-25 Alex van Ballegooij (CWI)
RAM: Array Database Management through Relational Mapping

2009-26 Fernando Koch (UU)
An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27 Christian Glahn (OU)
Contextual Support of social Engagement and Reflection on the Web

2009-28 Sander Evers (UT)
Sensor Data Management with Probabilistic Models

2009-29 Stanislav Pokraev (UT)
Model-Driven Semantic Integration of Service-Oriented Applications

2009-30 Marcin Zukowski (CWI)
Balancing vectorized query execution with bandwidth-optimized storage

2009-31 Sofiya Katrenko (UVA)
A Closer Look at Learning Relations from Text

2009-32 Rik Farenhorst (VU) and Remco de Boer (VU)
Architectural Knowledge Management: Supporting Architects and Auditors

2009-33 Khiet Truong (UT)
How Does Real Affect Affect Affect Recognition In Speech?

2009-34 Inge van de Weerd (UU)
Advancing in Software Product Management: An Incremental Method Engineering Approach

2009-35 Wouter Koelewijn (UL)
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling

2009-36 Marco Kalz (OUN)
Placement Support for Learners in Learning Networks

2009-37 Hendrik Drachsler (OUN)
Navigation Support for Learners in Informal Learning Networks

2009-38 Riina Vuorikari (OU)
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context

2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)
Service Substitution – A Behavioral Approach Based on Petri Nets

2009-40 Stephan Raaijmakers (UvT)
Multinomial Language Learning: Investigations into the Geometry of Language

2009-41 Igor Berezhnyy (UvT)
Digital Analysis of Paintings

2009-42 Toine Bogers
Recommender Systems for Social Bookmarking

2009-43 Virginia Nunes Leal Franqueira (UT)
Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients

2009-44 Roberto Santana Tapia (UT)
Assessing Business-IT Alignment in Networked Organizations

2009-45 Jilles Vreeken (UU)
Making Pattern Mining Useful

2009-46 Loredana Afanasiev (UvA)
Querying XML: Benchmarks and Recursion

2010
====

2010-01 Matthijs van Leeuwen (UU)
Patterns that Matter

2010-02 Ingo Wassink (UT)
Work flows in Life Science

2010-03 Joost Geurts (CWI)
A Document Engineering Model and Processing Framework for Multimedia documents

2010-04 Olga Kulyk (UT)
Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments

2010-05 Claudia Hauff (UT)
Predicting the Effectiveness of Queries and Retrieval Systems

2010-06 Sander Bakkes (UvT)
Rapid Adaptation of Video Game AI

2010-07 Wim Fikkert (UT)
Gesture interaction at a Distance

2010-08 Krzysztof Siewicz (UL)
Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments

2010-09 Hugo Kielman (UL)
A Politiele gegevensverwerking en Privacy, Naar een

effectieve waarborging

2010-10 Rebecca Ong (UL)
Mobile Communication and Protection of Children

2010-11 Adriaan Ter Mors (TUD)
The world according to MARP: Multi-Agent Route Planning

2010-12 Susan van den Braak (UU)
Sensemaking software for crime analysis

2010-13 Gianluigi Folino (RUN)
High Performance Data Mining using Bio-inspired techniques

2010-14 Sander van Splunter (VU)
Automated Web Service Reconfiguration

2010-15 Lianne Bodenstaff (UT)
Managing Dependency Relations in Inter-Organizational Models

2010-16 Sicco Verwer (TUD)
Efficient Identification of Timed Automata, theory and practice

2010-17 Spyros Kotoulas (VU)
Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications

2010-18 Charlotte Gerritsen (VU)
Caught in the Act: Investigating Crime by Agent-Based Simulation

2010-19 Henriette Cramer (UvA)
People's Responses to Autonomous and Adaptive Systems

2010-20 Ivo Swartjes (UT)
Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative

2010-21 Harold van Heerde (UT)
Privacy-aware data management by means of data degradation

2010-22 Michiel Hildebrand (CWI)
End-user Support for Access to Heterogeneous Linked Data

2010-23 Bas Steunebrink (UU)
The Logical Structure of Emotions

2010-24 Dmytro Tykhonov
Designing Generic and Efficient Negotiation Strategies

2010-25 Zulfiqar Ali Memon (VU)
Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective

2010-26 Ying Zhang (CWI)
XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines

2010-27 Marten Voulon (UL)
Automatisch contracteren

2010-28 Arne Koopman (UU)
Characteristic Relational Patterns

2010-29 Stratos Idreos(CWI)
Database Cracking: Towards Auto-tuning Database Kernels

2010-30 Marieke van Erp (UvT)
Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval

2010-31 Victor de Boer (UVA)
Ontology Enrichment from Heterogeneous Sources on the Web

2010-32 Marcel Hiel (UvT)
An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems

2010-33 Robin Aly (UT)
Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

2010-34 Teduh Dirgahayu (UT)
Interaction Design in Service Compositions

2010-35 Dolf Trieschnigg (UT)
Proof of Concept: Concept-based Biomedical Information Retrieval

2010-36 Jose Janssen (OU)
Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification

2010-37 Niels Lohmann (TUE)
Correctness of services and their composition

2010-38 Dirk Fahland (TUE)
From Scenarios to components

2010-39 Ghazanfar Farooq Siddiqui (VU)
Integrative modeling of emotions in virtual agents

2010-40 Mark van Assem (VU)
Converting and Integrating Vocabularies for the Semantic Web

2010-41 Guillaume Chaslot (UM)
Monte-Carlo Tree Search

2010-42 Sybren de Kinderen (VU)
Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach

2010-43 Peter van Kranenburg (UU)
A Computational Approach to Content-Based Retrieval of Folk Song Melodies

2010-44 Pieter Bellekens (TUE)
An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain

2010-45 Vasilios Andrikopoulos (UvT)
A theory and model for the evolution of software services

2010-46 Vincent Pijpers (VU)
e3alignment: Exploring Inter-Organizational Business-ICT Alignment

2010-47 Chen Li (UT)
Mining Process Model Variants: Challenges, Techniques, Examples

2010-48 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2010-49 Jahn-Takeshi Saito (UM)
Solving difficult game positions

2010-50 Bouke Huurnink (UVA)
Search in Audiovisual Broadcast Archives

2010-51 Alia Khairia Amin (CWI)
Understanding and supporting information seeking tasks in multiple sources

2010-52 Peter-Paul van Maanen (VU)
Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention

2010-53 Edgar Meij (UVA)
Combining Concepts and Language Models for Information Access

2011
====

2011-01 Botond Cseke (RUN)
Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02 Nick Tinnemeier(UU)
Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

2011-03
Jan Martijn van der Werf (TUE)
Compositional Design and Verification of Component-Based Information Systems

2011-04 Hado van Hasselt (UU)
Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms

2011-05 Base van der Raadt (VU)
Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

2011-06 Yiwen Wang (TUE)
Semantically-Enhanced Recommendations in Cultural Heritage

2011-07 Yujia Cao (UT)
Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU)
BDI-based Generation of Robust Task-Oriented Dialogues

2011-09 Tim de Jong (OU)
Contextualised Mobile Media for Learning

2011-10 Bart Bogaert (UvT)
Cloud Content Contention

2011-11 Dhaval Vyas (UT)
Designing for Awareness: An Experience-focused HCI Perspective

2011-12 Carmen Bratosin (TUE)
Grid Architecture for Distributed Process Mining

2011-13 Xiaoyu Mao (UvT)
Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA)
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM)
Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA)
Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM)
Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU)
The Mind's Eye on Personal Profiles

2011-20 Qing Gu (VU)
Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD)
Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA)
System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA)
Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT)
Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)
Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU)
Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU)
Dynamic website optimization through autonomous management of design patterns

2011-28 Rianne Kaptein(UVA)
Effective Focused Retrieval by Exploiting Query Context and Document Structure

2011-29 Faisal Kamiran (TUE)
Discrimination-aware Classification

2011-30 Egon van den Broek (UT)
Affective Signal Processing (ASP): Unraveling the mystery of emotions

2011-31 Ludo Waltman (EUR)
Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

2011-32 Nees-Jan van Eck (EUR)
Methodological Advances in Bibliometric Mapping of Science

2011-33 Tom van der Weide (UU)
Arguing to Motivate Decisions

2011-34 Paolo Turrini (UU)
Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

2011-35 Maaike Harbers (UU)
Explaining Agent Behavior in Virtual Training

2011-36 Erik van der Spek (UU)
Experiments in serious game design: a cognitive approach

2011-37 Adriana Burlutiu (RUN)
Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

2011-38 Nyree Lemmens (UM)
Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU)
Organizing Adaptation using Agents in Serious Games

2011-40 Viktor Clerc (VU)
Architectural Knowledge Management in Global Software Development

2011-41 Luan Ibraimi (UT)
Cryptographically Enforced Distributed Data Access Control

2011-42 Michal Sindlar (UU)
Explaining Behavior through Mental State Attribution

2011-43 Henk van der Schuur (UU)
Process Improvement through Software Operation Knowledge

2011-44 Boris Reuderink (UT)
Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT)
Statistical Language Models for Alternative Sequence Selection

2011-46 Beibei Hu (TUD)
Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

2011-47 Azizi Bin Ab Aziz(VU)
Exploring Computational Models for Intelligent Support of Persons with Depression

2011-48 Mark Ter Maat (UT)
Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

2011-49 Andreea Niculescu (UT)
Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

2012
====

2012-01 Terry Kakeeto (UvT)
Relationship Marketing for SMEs in Uganda

2012-02 Muhammad Umair(VU)
Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

2012-03 Adam Vanya (VU)
Supporting Architecture Evolution by Mining Software Repositories

2012-04 Jurriaan Souer (UU)
Development of Content Management System-based Web Applications

2012-05 Marijn Plomp (UU)
Maturing Interorganisational Information Systems

2012-06 Wolfgang Reinhardt (OU)
Awareness Support for Knowledge Workers in Research Networks

2012-07 Rianne van Lambalgen (VU)
When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

2012-08 Gerben de Vries (UVA)
Kernel Methods for Vessel Trajectories

2012-09 Ricardo Neisse (UT)
Trust and Privacy Management Support for Context-Aware Service Platforms

2012-10 David Smits (TUE)
Towards a Generic Distributed Adaptive Hypermedia

Environment

2012-11 J.C.B. Rantham Prabhakara (TUE)
Process Mining in the Large: Preprocessing, Discovery, and Diagnostics

2012-12 Kees van der Sluijs (TUE)
Model Driven Design and Data Integration in Semantic Web Information Systems

2012-13 Suleman Shahid (UvT)
Fun and Face: Exploring non-verbal expressions of emotion during playful interactions

2012-14
Evgeny Knutov(TUE)
Generic Adaptation Framework for Unifying Adaptive Web-based Systems

2012-15 Natalie van der Wal (VU)
Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes

2012-16 Fiemke Both (VU)
Helping people by understanding them - Ambient Agents supporting task execution and depression treatment

2012-17 Amal Elgammal (UvT)
Towards a Comprehensive Framework for Business Process Compliance

2012-18 Eltjo Poort (VU)
Improving Solution Architecting Practices

2012-19 Helen Schonenberg (TUE)
What's Next? Operational Support for Business Process Execution

2012-20 Ali Bahramisharif (RUN)
Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing

2012-21 Roberto Cornacchia (TUD)
Querying Sparse Matrices for Information Retrieval

2012-22 Thijs Vis (UvT)
Intelligence, politie en veiligheidsdienst: verenigbare grootheden?

2012-23 Christian Muehl (UT)
Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction

2012-24 Laurens van der Werff (UT)
Evaluation of Noisy Transcripts for Spoken Document Retrieval

2012-25 Silja Eckartz (UT)
Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application

2012-26 Emile de Maat (UVA)
Making Sense of Legal Text

2012-27 Hayrettin Gurkok (UT)
Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games

2012-28 Nancy Pascall (UvT)
Engendering Technology Empowering Women

2012-29 Almer Tigelaar (UT)
Peer-to-Peer Information Retrieval

2012-30 Alina Pommeranz (TUD)
Designing Human-Centered Systems for Reflective Decision Making

2012-31 Emily Bagarukayo (RUN)
A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure

2012-32 Wietske Visser (TUD)
Qualitative multi-criteria preference representation and reasoning

2012-33 Rory Sie (OUN)
Coalitions in Cooperation Networks (COCOON)

2012-34 Pavol Jancura (RUN)
Evolutionary analysis in PPI networks and applications

2012-35 Evert Haasdijk (VU)
Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics

2012-36 Denis Ssebugwawo (RUN)
Analysis and Evaluation of Collaborative Modeling Processes

2012-37 Agnes Nakakawa (RUN)
A Collaboration Process for Enterprise Architecture Creation

2012-38 Selmar Smit (VU)
Parameter Tuning and Scientific Testing in Evolutionary Algorithms

2012-39 Hassan Fatemi (UT)
Risk-aware design of value and coordination networks

2012-40 Agus Gunawan (UvT)
Information Access for SMEs in Indonesia

2012-41 Sebastian Kelle (OU)
Game Design Patterns for Learning

2012-42 Dominique Verpoorten (OU)
Reflection Amplifiers in self-regulated Learning

2012-43 Withdrawn

2012-44 Anna Tordai (VU)
On Combining Alignment Techniques

2012-45 Benedikt Kratz (UvT)
A Model and Language for Business-aware Transactions

2012-46 Simon Carter (UVA)
Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

2012-47 Manos Tsagkias (UVA)
Mining Social Media: Tracking Content and Predicting Behavior

2012-48 Jorn Bakker (TUE)
Handling Abrupt Changes in Evolving Time-series Data

2012-49 Michael Kaisers (UM)
Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

2012-50 Steven van Kervel (TUD)
Ontologogy driven Enterprise Information Systems Engineering

2012-51 Jeroen de Jong (TUD)
Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

2013
====

2013-01 Viorel Milea (EUR)
News Analytics for Financial Decision Support

2013-02 Erietta Liarou (CWI)
MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing

2013-03 Szymon Klarman (VU)
Reasoning with Contexts in Description Logics

2013-04 Chetan Yadati(TUD)
Coordinating autonomous planning and scheduling

2013-05 Dulce Pumareja (UT)
Groupware Requirements Evolutions Patterns

2013-06 Romulo Goncalves(CWI)
The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience

2013-07 Giel van Lankveld (UvT)
Quantifying Individual Player Differences

2013-08 Robbert-Jan Merk(VU)
Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

2013-09 Fabio Gori (RUN)
Metagenomic Data Analysis: Computational Methods and Applications

2013-10 Jeewanie Jayasinghe Arachchige(UvT)
A Unified Modeling Framework for Service Design

2013-11 Evangelos Pournaras(TUD)
Multi-level Reconfigurable Self-organization in Overlay Services

2013-12 Marian Razavian(VU)
Knowledge-driven Migration to Services

2013-13 Mohammad Safiri(UT)
Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly

2013-14 Jafar Tanha (UVA)
Ensemble Approaches to Semi-Supervised Learning

2013-15 Daniel Hennes (UM)
Multiagent Learning - Dynamic Games and Applications

2013-16 Eric Kok (UU)
Exploring the practical benefits of argumentation in multi-agent deliberation

2013-17 Koen Kok (VU)
The PowerMatcher: Smart Coordination for the Smart Electricity Grid

2013-18 Jeroen Janssens (UvT)
Outlier Selection and One-Class Classification

2013-19 Renze Steenhuizen (TUD)
Coordinated Multi-Agent Planning and Scheduling

2013-20 Katja Hofmann (UvA)
Fast and Reliable Online Learning to Rank for Information Retrieval

2013-21 Sander Wubben (UvT)
Text-to-text generation by monolingual machine translation

2013-22 Tom Claassen (RUN)
Causal Discovery and Logic

2013-23 Patricio de Alencar Silva(UvT)
Value Activity Monitoring

2013-24 Haitham Bou Ammar (UM)
Automated Transfer in Reinforcement Learning

2013-25 Agnieszka Anna Latoszek-Berendsen (UM)
Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

2013-26 Alireza Zarghami (UT)
Architectural Support for Dynamic Homecare Service Provisioning

2013-27 Mohammad Huq (UT)
Inference-based Framework Managing Data Provenance

2013-28 Frans van der Sluis (UT)
When Complexity becomes Interesting: An Inquiry into the Information eXperience

2013-29 Iwan de Kok (UT)
Listening Heads

2013-30 Joyce Nakatumba (TUE)
Resource-Aware Business Process Management: Analysis and Support

2013-31 Dinh Khoa Nguyen (UvT)
Blueprint Model and Language for Engineering Cloud Applications

2013-32 Kamakshi Rajagopal (OUN)
Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development

2013-33 Qi Gao (TUD)
User Modeling and Personalization in the Microblogging Sphere

2013-34 Kien Tjin-Kam-Jet (UT)
Distributed Deep Web Search

2013-35 Abdallah El Ali (UvA)
Minimal Mobile Human Computer Interaction

2013-36 Than Lam Hoang (TUe)
Pattern Mining in Data Streams

2013-37 Dirk Börner (OUN)
Ambient Learning Displays

2013-38 Eelco den Heijer (VU)
Autonomous Evolutionary Art

2013-39 Joop de Jong (TUD)
A Method for Enterprise Ontology based Design of Enterprise Information Systems

2013-40 Pim Nijssen (UM)
Monte-Carlo Tree Search for Multi-Player Games

2013-41 Jochem Liem (UVA)
Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning

2013-42 Léon Planken (TUD)
Algorithms for Simple Temporal Reasoning

2013-43 Marc Bron (UVA)
Exploration and Contextualization through Interaction and Concepts

2014
====

2014-01 Nicola Barile (UU)
Studies in Learning Monotone Models from Data

2014-02 Fiona Tuliyano (RUN)
Combining System Dynamics with a Domain Modeling Method

2014-03 Sergio Raul Duarte Torres (UT)
Information Retrieval for Children: Search Behavior and Solutions

2014-04 Hanna Jochmann-Mannak (UT)
Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation

2014-05 Jurriaan van Reijsen (UU)
Knowledge Perspectives on Advancing Dynamic Capability

2014-06 Damian Tamburri (VU)
Supporting Networked Software Development

2014-07 Arya Adriansyah (TUE)
Aligning Observed and Modeled Behavior

2014-08 Samur Araujo (TUD)
Data Integration over Distributed and Heterogeneous Data Endpoints

2014-09 Philip Jackson (UvT)
Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language

2014-10 Ivan Razo-Zapata (VU)
Service Value Networks

2014-11 Janneke van der Zwaan (TUD)
An Empathic Virtual Buddy for Social Support

2014-12 Willem van Willigen (VU)
Look Ma, No Hands: Aspects of Autonomous Vehicle Control

2014-13 Arlette van Wissen (VU)
Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains

2014-14 Yangyang Shi (TUD)
Language Models with Meta-information

2014-15 Nataliya Mogles (VU)
Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare

2014-16 Krystyna Milian (VU)
Supporting trial recruitment and design by automatically interpreting eligibility criteria

2014-17 Kathrin Dentler (VU/UVA)
Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability

2014-18 Mattijs Ghijsen (UVA)
Methods and Models for the Design and Study of Dynamic Agent Organizations