JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche
Fakultät

Adaptive Systems: Development, Evaluation and Evolution

DISSERTATION

zur Erlangung des akademischen Grades

Doktor

im Doktoratsstudium der

TECHNISCHEN WISSENSCHAFTEN / NATURWISSENSCHAFTEN

Eingereicht von:
Alexandros Paramythis

Angefertigt am:
Institut für Informationsverarbeitung und Mikroprozessortechnik (FIM)

Beurteilung:
o. Univ. Prof. Dr. Jörg R. Mühlbacher
Univ. Prof. Dr. Gabriele Kotsis

Linz, Oktober 2009

Adaptive Systems:
Development, Evaluation and Evolution

ABSTRACT

In recent years we have been witnesses to a proliferation of computing technology which now supports or mediates most forms of human activity. At the same time, the sources and amounts of information available to the general public are expanding with breathtaking speed. These factors, coupled with the widening (in size and diversity) population base for the new computing "appliances" and information services, render it vitally important that interactive systems and applications actively take steps to ensure that individual users receive an interactive experience tailored to their individual abilities, skills, needs and preferences.

Adaptation refers to the capacity of software systems to tailor themselves to better suit their environment, including their end users. More specifically, user-adaptive systems, adapt their behavior to individual users on the basis of processes of user model acquisition and application that involve some form of learning, inference, or decision making.

This thesis summarizes the author's work in three thematic areas of user-adaptive systems: the development of user-adaptive systems; the evaluation of adaptation; and, the evolution of user-adaptive systems towards the incorporation of learning and self-regulation capabilities.

In more detail, this thesis presents two different architectures and corresponding software frameworks intended to support the development of desktop- and web- based adaptive systems; these have been used to develop a

universally accessible adaptive web browser, and a tourist information system respectively.

The thesis then goes on to tackle the problem of evaluating adaptation, which has proven to elude standard evaluation methodologies and approaches typically used for interactive software; the work presented introduces an evaluation framework specifically intended to assess the extent to which the individual stages of adaptation attain their design goals.

Lastly, work is presented that points the way towards a new generation of adaptive systems that are capable of so-called meta-adaptation, i.e., of modeling, reasoning about, assessing, and tailoring their own adaptive behavior, and can thus evolve dynamically even after they have been deployed, and while in active use.

Adaptive Systems:
Development, Evaluation and Evolution

ZUSAMMENFASSUNG

In den vergangenen Jahren wurden wir Zeugen der Verbreitung von Rechnertechnologie, die nun bereits die meisten Formen menschlicher Aktivitäten unterstützt und vermittelt. Gleichzeitig wuchsen die für die allgemeine Öffentlichkeit verfügbaren Quellen und die verfügbare Informationsmenge mit atemberaubender Geschwindigkeit. Diese Faktoren, verknüpft mit der Ausweitung der Nutzerbasis dieser neuen "Information Appliances" und Informationsdienste (in Größe und Diversität), machen es erforderlich, dass interaktive Systeme und Anwendungen aktiv für eine an die jeweiligen Fähigkeiten, Kenntnisse, Bedürfnisse und Vorlieben der einzelnen Benutzer angepasste Interaktion sorgen.

Adaptation bezieht sich auf die Fähigkeit seitens des Softwaresystems, sich selbst anzupassen, um besser auf die Umgebung, inklusive der Endbenutzer, einzugehen. So genannte Benutzer-adaptive Systeme passen ihr Verhalten an den jeweiligen Benutzer und bedienen sich dabei Prozessen zur Erstellung und Anwendung eines Benutzermodells auf Basis von Lernen, Schlussfolgerung oder Entscheidungsfindung.

Diese Arbeit fasst die Forschungsergebnisse des Autors in drei thematischen Gebieten Benutzer-adaptiver Systeme zusammen: die Entwicklung von Benutzer-adaptiven Systemen; die Evaluierung der Adaption; und die Weiterentwicklung von Benutzer-adaptiven Systemen hin zu Fähigkeiten wie Lernen und Selbst-Regulierung.

In dieser Arbeit werden zwei unterschiedliche Architekturen und entsprechende Software-Frameworks vorgestellt, deren Ziel die Unterstützung der Entwicklung von Desktop- und Web-basierten adaptiven Systemen ist. Diese wurden verwendet, um einen universell zugänglichen, adaptiven Web-Browser und ein Touristen-Informationssystem zu entwickeln.

Des Weiteren widmet sich die Arbeit dem Problem der Evaluierung von Adaption. Es konnte gezeigt werden, dass dieser Anwendungsbereich nicht durch standardisierte Evaluierungsmethoden und -ansätze, wie sie typischerweise bei interaktiver Software angewendet werden, abgedeckt ist. Präsentiert wird ein Evaluierungs-Framework, welches speziell dazu gedacht ist zu beurteilen, in welchem Ausmaß in den einzelnen Stufen der Adaption die Entwurfsziele erreicht werden.

Abschließend werden Arbeitsergebnisse präsentiert, die den Weg zu einer neuen Generation adaptiver Systeme mit Fähigkeiten zur so genannten Meta-Adaption zeigen. Diese Systeme modellieren ihr eigenes adaptives Verhalten, ziehen daraus Schlüsse, bewerten es und passen es an. Sie können sich so dynamisch weiterentwickeln, sogar nachdem sie in Betrieb genommen wurden und auch während der aktiven Verwendung.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Like most long journeys in life, the one that brought me to this PhD has been full of people. Bear with me as I try to remember and thank at least those without whom my work and life would be a different color.

First and foremost I would like to express my sincere gratitude to my supervisor, o. Univ. Prof. Dr. Jörg R. Mühlbacher, Head of the Institute for Information Processing and Microprocessor Technology (FIM Institute) of the Johannes Kepler University, for his guidance, trust, open-mindedness, never-ending patience, and continuous support over the years. His scientific prowess, coupled with ethos and integrity, have inspired both my work and my person. This thesis would not have been possible without him.

I would also like to thank Univ. Prof. Dr. Gabriele Kotsis for her thorough review and fruitful discussions, which have been of great assistance in improving this thesis.

Thanks are also due to my friends and colleagues at the FIM Institute with whom I have gone through the joys and sorrows of university life and scientific work, as well as plenty of after-lunch cakes: Michael Sonntag, Susanne Loidl-Reisinger, Rudolf Hörmanseder, Franz Bauer, Christine Stimmeder, Inge Ritzinger, Christian Praher, Sabine Link and Muhammad Asif Habib. I hope I will be forgiven for singling out the past and present members of the "adaptive wing", with whom I have happily shared long working hours and adaptive barbeques: to Andreas Putzinger, Florian König, Mirjam Köck, David Hauger, Michael Moucka, and (the now officially Austrianized, and Krapfen-addict) Lex van Velsen, a personalized thanks!

I wouldn't be where I am today if I had not had the opportunity to spend many exciting and productive years at the Human-Computer Interaction Laboratory (HCI Lab) of the Institute of Computer Science, of the Foundation for Research and Technology - Hellas (ICS-FORTH). I am exceptionally indebted to Prof. Constantine Stephanidis, Head of the HCI Lab and Director of ICS-FORTH, who introduced me to the world of Human-Computer Interaction, taught me how to do research, and gave me the opportunities I needed to find out who I am and what I can achieve. I still remember the cloudy afternoon when, despite his tiredness and mountain of urgent things to do, he took the time to show me Brussels and buy me a gyros in the Grand Place.

When I left Greece, I left behind a lot of friends at the HCI Lab. For the endless hours of discussion, the enjoyable joint work, and the unbelievable amounts of collaboratively consumed coffee, my thanks to: Chrisoula Alexandraki, Napoleon Maou, Harald Weber, Alexandra Totter, Frank Leidermann, Margherita Antona, Michalis Sfyrakis, Athina Stergiou, Anthony Savidis, Demosthenes Akoumianakis, Adamantios Koumpis, George Paparoulis, Maria Pitsoulaki, Ioannis Segkos, Adam Leventis. A separate thanks to my close friend, "uncle" Dimitris Grammenos, for always being there (once even with a birthday cake and myself missing in action), for his endless creativity, for never letting me forget that users always come first, and for his excellent taste that has brought Anthi into my circle of friends!

Stephan Weibelzahl needs to be mentioned separately because he is a special case. Although we haven't had the pleasure of working in the same place (yet), we have collaborated closely for the past nine years. Stephan's extensive knowledge on adaptation and evaluation, but also his humor, kindness and unassuming nature are inspiring and have been driving forces in seeing our work through to fruition.

Parts of the work reported in this thesis have been supported by funding from European and National R&D projects: the ACTS-AC042 AVANTI project, the IST-1999-20656 PALIO project, the project "Integrating Agents into Teleteaching Webportals" funded by Austria's FWF, and the Socrates / Minerva 229714-CP-1-2006-1-MINERVA-M "Adaptive Learning Spaces" project. I am grateful for the support received, and for having had the opportunity to work with, and learn from, leading experts within these projects.

Hard though it may be to believe (for those that know me at least), I have managed to sneak a bit of non-computer-dominated life in with the rest, and fill it with wonderful people.

For their friendship and hospitality in Grüeziland I am indebted to Alexandra, the Frank-Uli-Matilda-Oskar quartet, Christine and Louisa. I can never repay, but at least I will never forget your kindness.

Eva has stood by me with unwavering patience and understanding in good and in bad times, relentlessly manipulating the Universe to my advantage, and teaching me little by little that one can rely on others without losing oneself. To her and the now-too-old-to-still-be-called babies, a heart-felt thanks for brightening my days.

What words would ever be enough to thank people who have always been there, loving and selfless beyond belief, supportive even when they disagreed, proud of all of one's achievements however small, for ever caring and worried because of and despite the distance? To my parents, Spyros and Niki, I owe a big part of who I am. I hope I will be able to repay them a little part of what I have generously received by always being there for them too when they need me. Dimitra is simply the *bestest* sister in the world, and without herself, Yiannis, my wonderful niece Clio, and the forthcoming addition(s?), my life would have been poorer.

Nothing has changed me more than the coming of my daughter Elena. Her huge eyes, warm heart, intelligence, creativity, and unconditional love have seen me through times and situations that I hadn't even imagined I could manage, and have taught me what really matters in life. She has been patiently waiting for years for "papa to finish his book", painting me pictures of the sea the all along, so that I don't miss it too much. Now that the "book" is done, I can only hope to be able to give her more of what she so richly deserves. I wouldn't want to live in a world without her.

ACRONYMS

AH. Adaptive Hypermedia.

AHS. Adaptive Hypermedia Systems. Asynchronous JavaScript and XML

AJAX. Asynchronous JavaScript and XML

API. Application Programming Interface.

AT. Assistive Technology.

CL. (User) Communication Layer.

CM. Context Model / Context Modeling.

CMS. Context Modeling Server.

CnM. Content Model.

CSCL. Computer Supported Collaborative Learning.

CSCW. Computer Supported Collaborative Work.

CSS. Cascading Style Sheets.

DPS. Dynamic Personalization Server.

DSL. Device Software Layer.

GIS. Geographic Information System.

GPS. Global Positioning System.

GUI. Graphical User Interface(s).

HCI. Human-Computer Interaction.

HSA. Hyper-Structure Adaptor.

HTML. Hypertext Markup Language.

HTTP. Hypertext Transfer Protocol.

IAS. Interactive Adaptive Systems.

IST. Information Society Technologies.

ITS. Intelligent Tutoring Systems.

KQML. Knowledge Query and Markup Language.

LDAP. Lightweight Directory Access Protocol.

MDI. Multimedia Database Interface.

MIME. Multipurpose Internet Mail Extensions.

MMS. Multimedia Messaging Service.

MSAA. Microsoft Active Accessibility.

MSIE. Microsoft Internet Explorer (browser).

NN. Netscape Navigator (browser).

ODL. Open and Distant Learning.

PDF. (Adobe) Portable Document Format.

SCC. Service Control Center.

SMS. Short Message Service.

SOA. Service-Oriented Architecture.

SOAP. Simple Object Access Protocol.

UA. Universal Access.

UAS. User-Adaptive Systems.

UCD. User-Centered Design.

UM. User Model / User Modeling.

UMS. User Modeling Server.

URI. Uniform Resource Identifier.

URL. Uniform Resource Locator.

UUI. Unified User Interface.

UUID. Unified User Interface Design.

WAP. Wireless Application Protocol.

WWW. World Wide Web.

XHTML. Extensible Hypertext Markup Language.

XML. Extensible Markup Language.

XSLT. Extensible Stylesheet Language Transformations.

Unit A

INTRODUCTION

This first Unit of this thesis provides a thorough overview of the area of adaptive systems, starting from the motivation for introducing adaptation in interactive software in the first place. It then moves on to explore the characteristics of different forms of adaptation, both terminologically, and from an engineering perspective. Taxonomies and models of adaptation are also discussed to provide a setting for introducing the individual elements of the work presented here, and for relating them to other efforts in the area. Following that, the potential problems that may be caused and the challenges that must be overcome when adaptivity is introduced in an interactive system are addressed.

This Unit also provides an overview of the main areas of work reported herein, namely, the design and implementation of architectures and frameworks for desktop- and web- based adaptive systems; the evaluation of adaptive systems; and, finally, the introduction of meta-adaptive capabilities to next-generation adaptive systems. Furthermore, it gives an overview of the research and employment context in which the reported work was carried out.

A.1 What Is Adaptation And Why Do We Need It?

A.1.1 The Need for Adaptation

Recent years have seen a phenomenal change in the way we perceive and interact with computing systems: information and communication "devices" keep getting smaller, incorporating functionality and capabilities not even conceived of a few years ago; computing power and interactivity are becoming ubiquitous in our environment; systems and applications are increasingly targeted to a public with little or no prior computer experience; etc. As a result, a rapidly increasing number of human activities are supported by computers, while the relation of users with computers is steadily evolving into a more interactive dialogue than ever before.

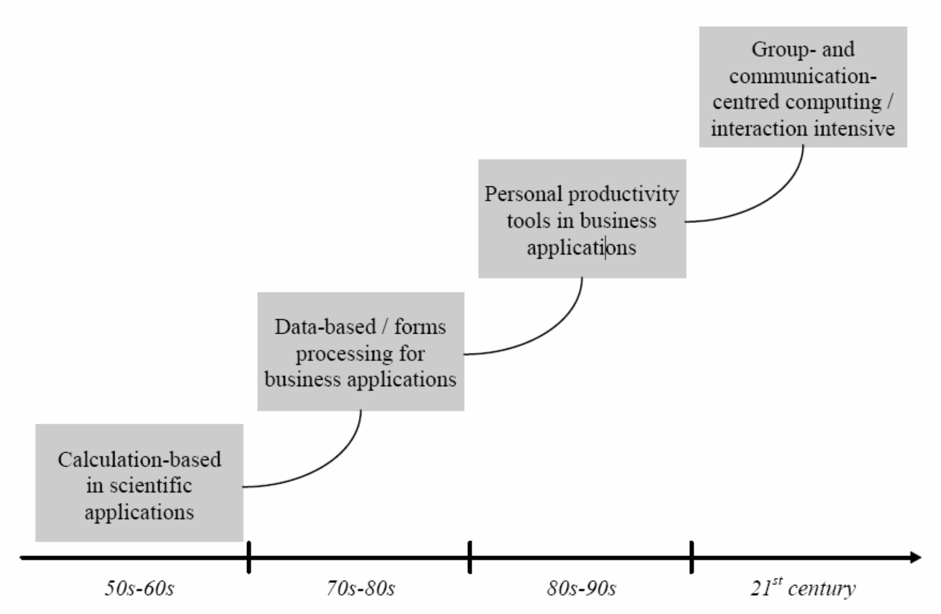


Figure 1: Shift in computing paradigms [Stephanidis et al., 1998b].

The inevitable changes in the role of, and the ways in which we use computers –as progressively manifested since the beginnings of their widespread use, and into the 21st century (see Figure 1)–, are already playing a catalytic role in how we approach the design and development of interactive systems. The major challenge for Human-Computer Interaction (HCI) theory and practice, in this evolving context, is to ensure high-quality, "accessible" interaction for all potential users of interactive software, while ensuring that information, functionality and services remain relevant to the individual user.

To meet this challenge, HCI needs to revise a significant number of premises that have been amongst the field's basic tenants since its inception. One such popular premise is the concept of designing for the "average" user. The transition of computing from the narrow confines of the work environment, into everyday activities with an emphasis on communication, knowledge- and information- sharing, user-content creation and consumption, etc., has rendered the notion of a "typical", or "average" user far less applicable than when the computer was a calculation machine, or a productivity support tool [Stephanidis et al., 1998b]. Instead, HCI is increasingly focusing on identifying the unique abilities, skills, interest, knowledge, etc. of individual users, and devising approaches for accommodating them in the design of interactive systems. One such approach is the employment of software that can automatically personalize / tailor itself on the basis of the characteristics of the software's end users (as the most important among other potentially relevant factors). Before we proceed to discuss what this approach entails though, let us explore further what exactly it is that makes users different and mandates adaptation in the first place.

Individual differences have been a focus of study for several years in the context of HCI. For example, [Sternberg, 1985] presents analyses of the dimensions that differentiate the potential users of a system from the perspective of their information processing capabilities. The capabilities that are identified and analyzed are: general intellectual ability; verbal ability; reading ability; second-language abilities; individual differences in learning and memory; mathematical ability; mental imagery ability; deductive reasoning ability; inductive reasoning ability; and, problem-solving ability. The more general recognition of the important role played by the resulting heterogeneity amongst users in the perceived quality, usefulness and overall value of a system has been a driving force in the area of interactive adaptive systems.

One of the best known argumentations for the need for adaptive interaction is set forth in [Browne, Norman & Riches, 1990]. The argumentation therein moves along two complementary axes: the reasons (needs) for adaptation within interaction, and the purpose (expected benefits) of individual adaptations. As far as the first axis is concerned, one of the basic factors cited is the increasing heterogeneity in the population of computers and applications. Identified areas where the users' individual differences may occur include[1]: psycho-motor skills, capability, understanding, expectations, motives, requirements, cognitive strategies, cognitive abilities, and preferences. The authors also point out that the aforementioned factors are directly dependent upon time (i.e., they may change during short or longer

1 Some of these areas are to be interpreted in relation to interaction with a specific system, and not necessarily in their general incarnations. For instance, "expectations" refers to the users' expectations of the system at hand, and "motives" refers to their motivation for using the system in the first place.

interaction sessions, or even over several interactive sessions), as well as upon the particular situation in which they manifest themselves[2].

As far as the second axis of argumentation in [Browne, Norman & Riches, 1990] is concerned ("expected benefits"), the authors identify the following potential benefits: extension of system's lifespan; extension of the system's potential population basis; better support for the (user's) interaction goals; satisfaction of user needs; increase of the speed of interaction; enhancement of a system's ease of learning; reduction of a system's learning requirements; and, facilitation of a user's understanding of system functionality (and respective interactive options).

The literature provides a wealth of work that, moving along lines similar to the above, enumerates dimensions of individual differences amongst humans, and argues for the employment of adaptation to accommodate for these in the context of interaction (see, e.g., [Benyon, 1993] for a thorough analysis).

These general dimensions of individual differences are, of course, only indicative. One could intuitively add several further user characteristics that may call for tailoring of an interactive system to the user's needs, such as a user's mother tongue, educational level, family status, profession, or even the user's cultural and social background [Stephanidis et al., 1997b]. Or, borrowing elements from Ethnography, address topics such as the changes in a user's needs at different times of the user's life (including issues related to changing interests, deteriorating abilities, etc.) Or, even, following the approach proposed in [Paetau, 1994], use methods borrowed from Sociology to identify other pertinent differentiating factors and ways for addressing them.

The situation gets even more complicated when one considers users' needs and preferences in relation to a specific application domain: news one might be interested in, holiday destinations preferred, books to read, movies to watch, goods to buy …, the list is endless. For instance, using the domain of e-learning as an indicative case, one would have to consider differences among learners in their cognitive- and learning- styles [McLoughlin, 1999][3]: the differences between *wholists* and *analysts* would change the importance and role of learning material overviews; the differences between *verbalizers* and *imagers* would have definite implications on the preferred form of information presentation; etc. Furthermore, these types of differences are to be addressed *in addition to*, rather than instead of, the general ones already discussed.

---

2 The dimension of the "situation" that characterizes interaction, or within which interaction occurs, is a factor that is repeatedly emphasized in the literature. It has been established, terminologically, to refer to this and other related parameters of interaction with the term "context of use" [ISO 13407, 1999], which is the one primarily used in this thesis.

3 Note that there exists recent literature that both contradicts (e.g., [Brown et al., 2006]) and supports (e.g., [Vasilyeva et al., 2007]) the use of learning styles for personalizing (the delivery of) e-learning content. For an up to date overview please refer to [Popescu, 2009]

However many individual differences one could identify, and however distinct in nature these might be, though, the basic problem remains the same: even among users that superficially appear to have identical needs (as far as their interaction with systems and applications is concerned), there exists, in fact, a multitude of differences that, not only could, but should be addressed by software that aspires to provide individualized support to its users.

This individualized support can come in many guises and range from simple configurability that users have to carry out themselves, all the way to intelligent interactive support that is based on inferred user requirements and preferences. This range of capabilities and their corresponding methodological approaches and practical techniques are commonly referred to in the literature as "adaptation". The next section provides an in-depth look into the semantics of the related terms and the evolution of the corresponding scientific area in the past decades.

A.1.2 Definitions, Models and Taxonomies of Adaptation

Having discussed the rationale for adaptation in the first place, we now move our attention to exploring the term "adaptation" and its various meanings in the literature. We will look in detail into:

- *adaptation* in general, referring to the idea of having a system that can be tailored to one's individual needs;
- *self-adaptation*, adding the capability on the part of the system to perform the tailoring itself;
- *adaptability*, incorporating the notion of the system's being able to carry out by itself most of the steps required to decide upon and effect adaptation; and, finally,
- *adaptivity*, denoting, in addition to the above, that the system is capable of acquiring the user model, and performing non-trivial mapping between the contents of the said model and the range of possible forms of tailoring at runtime.

Figure 2 depicts the conceptual relationships between the above terms, as these are employed in this thesis, and will serve as a guide for discussion. Note that, as the figure suggests, the boundaries between the notions that the terms embody often overlap, a fact which will be further discussed in this section. Also note that the terms (and corresponding forms of adaptation) are not to be understood as mutually exclusive – several modern systems concurrently employ a range of these in practice. Towards the goal of exploring the aforementioned terms and concepts, this section provides a brief overview of definitions, models and taxonomies of adaptation in the context of interactive systems.

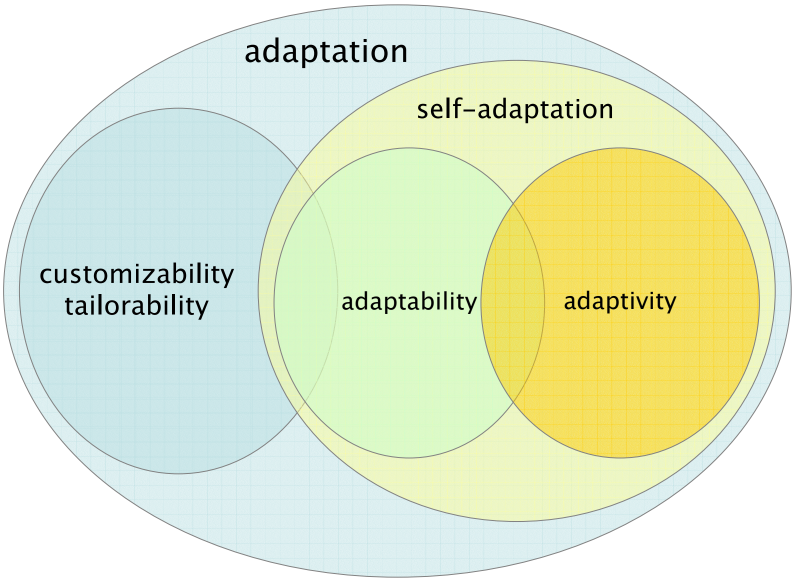


Figure 2: Conceptual relationship between adaptation terms, as employed in this thesis.

Early work specifically addressing adaptation in interactive software can be traced back to the late 80s and focused on adaptivity targeting the interaction itself (as the adaptation constituent), and often went under the banner of "Adaptive User Interfaces" (AUIs). The scope of work in the area expanded, as one might expect, over the years. The advent and widespread penetration of the World-Wide Web (here forth: *web*) marked a decisive turn of attention of the community towards adaptation in the context of hypermedia systems. The later, along with recommender systems, arguably constitute the main foci of related research today. It should be noted that although the discussion that follows draws from specific types of adaptation-capable interactive software (e.g., AUIs), the findings, categorizations, etc., highlighted are deemed to be of more general applicability, unless indicated otherwise.

Before embarking on our exploration of what adaptation is, and how it has been understood and applied to date, it is worth noting that, perhaps due to its relatively young age, the area of Interactive Adaptive Systems (IAS) –also often referred to as User-Adaptive Systems (UAS)– has only recently started converging on commonly accepted terms and taxonomies; and important differences still persist. The differences exhibited between definitions and taxonomies that have been proposed to date are several and important in nature (compare for example the definitions from [Oppermann & Simm, 1994] and [Krogsæter & Thomas, 1994] with those in [Chignell et al., 1989] and [Dieterich et al., 1993]). The overview provided here does not attempt to reconcile the said differences, but rather to bring forth the important elements in each proposition and, based on these, assemble the contextual definitions that underpin the presented work.

A.1.2.1 Adaptation Definitions and Models

The different approaches to adaptation in interactive systems can be broadly classified into two coarse categories: "hard-wired" adaptation and "system supported" adaptation. The first category of systems allows users to select (or activate) alternative characteristics for presentation, dialogue, or functionality, among those that are already designed ("hard-wired") into the system. Typical examples of this type of adaptation include the customization of the system's behavior, the modification of the position and size of interactive elements in Graphical User Interfaces (GUIs), etc., typically effected through preference dialogs. This category coincides with "customizability / tailorability" in Figure 2. The most important shortcomings in systems in this category are that the adaptations offered by the system are usually constrained, and most often pre-packaged (and, therefore, determined entirely) by the system developer. Furthermore, the utilization of such capabilities in a system presupposes considerable familiarity of the end user with the system, a fact that may by itself invalidate the entire approach: by the time users know enough to adapt the system to their needs, they may have well adapted themselves to the system instead.

The second of the approaches mentioned above is based on the principle that the system should be capable of identifying the situations in which adaptation is necessary (or may be beneficial to the end user), and further selecting and activating an appropriate course of action. This category of systems coincides with "self-adaptation" in Figure 2. The preceding description incorporates the notion that the system may also be capable of monitoring the user's interaction with it so as to extract information about the user, verifying, improving, reassessing, and, if necessary, retracting assumptions known to be true for a given user (which, as we will soon see, brings us to the area of adaptivity).

As far as the nature and level of control that end users have over adaptation, [Dieterich et al., 1993] identify two extreme cases:

- In the first case, the system is confined to an advisory role, taking over the task of recommending adaptations to the user, who is then responsible for acting upon (including effecting) these recommendations. This approach is termed "system support for user-controlled adaptation".
- In the second case, the system assumes a more active role and undertakes the task of effecting the adaptations that are deemed appropriate at a given point in time. This is in fact the most popular interpretation of the term "adaptive system" (always with respect to user control).

Naturally, there exist a lot of approaches that fall between these two extremes, such as, for instance, systems that ask users for permission before effecting an adaptation, or systems that "prepare" adaptations (e.g., in the form of macro commands) and allow the user to apply them within one interaction step [Krogsæter & Thomas, 1994].

Based on the above, the term "self-adaptation" is used in this thesis to refer to systems that are both capable of adaptation in general, and of effecting said adaptation themselves, irrespectively of the strategy employed to ensure that the end user has control over the adaptation process (see also Figure 2).

Self-adaptation can be further categorized along a number of more specific adaptation dimensions, which include: the strategy employed for user control over adaptation, the knowledge used by the system to decide upon adaptation, the decision making process used for associating existing knowledge with adaptation actions, etc.

In [Tyler et al., 1991] the following traits are identified as prerequisites for intelligent user interfaces[4] that encompass adaptation capabilities (therefore adhering to the concept of self-adaptation as just described): (a) representation of knowledge in the system; (b) system modularity; (c) capability on the part of the system to infer and interpret user plans; and (d) capability on the part of the system to tailor its behavior to the individual user and task the user is performing.

From a somewhat different perspective, Szekely [1991] asserts that a user interface can "intelligently" tailor communication with the user to the degree that any adaptation decision is based on the following information sources: (a) program model – supported objects and functions, design choices; (b) user model; (c) task model; (d) workstation model – characteristics of the hardware, software and input / output devices; and (e) knowledge about the user interface design.

Browne, Norman & Adhami [1990] identify four basic methods that must characterize the design and implementation of adaptive user interfaces. Specifically, they observe that an adaptive system needs a representation of the dimensions of differentiation upon which adaptations will be based. In the case of adaptive interactive systems this implies, among other things, the presence of user models and corresponding mechanisms for their creation and maintenance. The proposed methods of modeling include: static updateable models, comparison models, alternative static models, plan-recognition-based modeling, and usage modeling[5]. The second method that

---

4 Intelligence, in the context of the cited work, refers to the employment of Artificial Intelligence techniques within a user interface, with the goal of tailoring that interface's presentation and behavior to the needs of its end users. An introduction to the field of Intelligent User Interfaces can be found in [Chignell et al., 1989].

5 A comprehensive overview of the area and of the existing techniques for modeling users and their use in the development of adaptive systems can be found in [Kobsa, 1993].

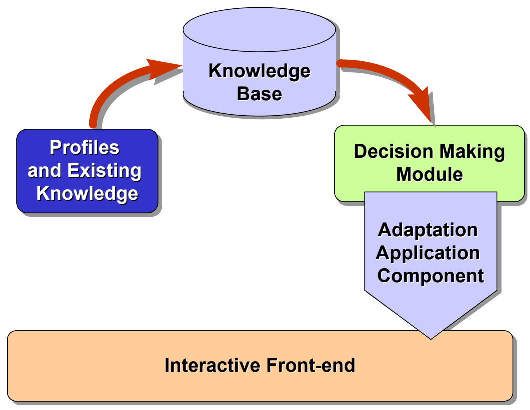
the authors address is the modeling of the dialogue between the user and the interface, so as to support and facilitate modifications within that dialogue. References to common dialogue modeling techniques include: state transition networks, context-free grammars, and event models of user interaction. The third method refers to the modeling of user tasks, with the goal of specializing adaptations to the characteristics of the interaction circumstances at hand. Finally, Browne, Norman & Adhami [1990] propose the creation of an application model, through which it becomes possible to better separate between the user interface and the application backend, allowing for the independent adaptation of the two.

A more technical perspective on the techniques that can be used to develop AUIs can be found in [Totterdell et al., 1990], which seeks to enumerate known and tested techniques (not necessarily coming from the domain of HCI) that can be used as the basis for adaptation. The techniques proposed include: genetic algorithms and genetic programming, adaptive scheduling, pattern matching, context-oriented adaptation, and user modeling.

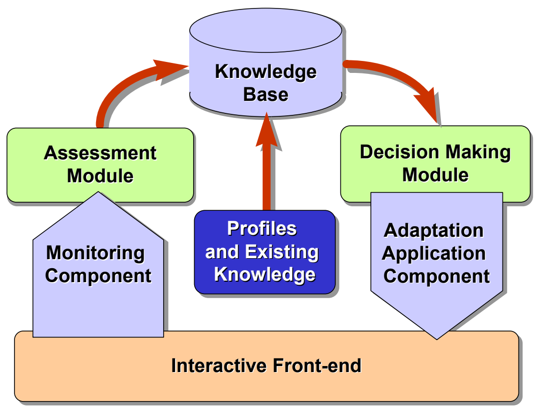
One terminological distinction that is important to make before proceeding further in this analysis, is that between *adaptability* and *adaptivity*, which was first introduced in Figure 2 (page 7). Both terms are often used to describe the capabilities of self-adapting systems that have several of the aforementioned characteristics. However, adaptability is also often used to refer to customizability by the end user (see, e.g., [Jameson, 2008]), among other things. In the context of this thesis the terms are used as follows:

- The term *adaptability* is used to refer to self-adaptation that is based on knowledge (regarding the user, the interaction environment, the context of use, etc.) that is available to (or is collected by) the system prior to the commencement of interaction, and leads to adaptations which also precede the commencement of interaction. Note that this definition does not imply that all adaptations need to be applied before interaction starts, but rather that they can be potentially decided upon prior to interaction (or between interaction sessions). In other words, the adaptation decisions a system would make would not be different at any point of an interaction session. Figure 3 (a) depicts an abstract architectural view summarizing the characteristics that need to be supported by systems capable of the form of adaptability described [Stephanidis, Paramythis et al., 1998b].

- The term *adaptivity* refers to self-adaptation that is based on knowledge (again, regarding the user, the interaction environment, the context of use, etc.) which is collected and / or maintained by the system during interaction sessions (either directly from the user, or through monitoring / inferencing techniques) and which leads to adaptations that take place while the user is interacting with the system. Figure 3 (b) depicts an abstract architectural view summarizing the characteristics that need to be supported by systems capable of the form of adaptavity described. Note that the main differentiating factor to adaptability is the capability on the part of the system to gather, assess and act upon new information about the user, context of use, etc. [Stephanidis, Paramythis et al., 1998b].



(a) Abstract system architecture for supporting adaptability.



(b) Abstract system architecture for supporting adaptivity.

Figure 3: Abstract architectures of systems that support: (a) adaptability, and (b) adaptivity [Stephanidis, Paramythis et al., 1998b].

Given the fact that the above definitions are generic in nature, it is important to address some "marginal" cases of self-adaptation for clarification purposes. One such case concerns adaptations that take place immediately after the commencement of interaction, and following the users' completing an interactive questionnaire (through which users provide additional information to the system about their skills, interests, knowledge, etc.) This case of self-adaptation is considered to fall in the category of adaptability, when: (a) it is equivalent to the use of a user- / context- / etc. profile, which, instead of being available beforehand, is collected directly from the user; and (b) the knowledge thus gathered is stored and used in subsequent interaction sessions without ever being revisited, unless with the direct intervention of the user.

Another interesting "marginal" case concerns adaptations that take place before the commencement of interaction, but are based on knowledge that has been collected *dynamically* during past interaction sessions (apparently this can only be the case in systems that maintain collected knowledge between interaction sessions). This case of self-adaptation is considered to fall in the category of adaptivity when the knowledge used to determine the system's adaptation behavior is reexamined / reassessed during subsequent interaction sessions.

An excellent summarization that binds together several of the elements discussed thus far is given by Jameson [2008], who provides the following description of how a user-adaptive system works in general terms (see also Figure 4):

> "A UAS makes use of some type of information about the current individual user, such as the choices [the user] has made when [interacting with the system]. In the process of user model acquisition, [the system] performs some type of learning and/or inference on the basis of the information about [the user] in order to arrive at some sort of user model, which in general concerns only limited aspects of [the user]. In the process of user model application, [the system] applies the user model to the relevant features of the current situation in order to determine how to adapt its behavior to [the user]."

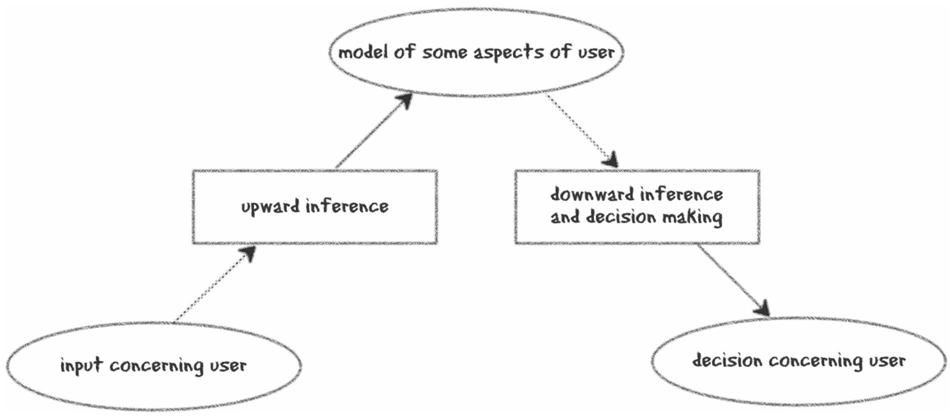


Figure 4: General schema for the processing in a user-adaptive system (adapted from [Jameson, 2008]).

Based on this description, Jameson goes on to define UAS as follows:

> "An interactive system that adapts its behavior to individual users on the basis of processes of user model acquisition and application that involve some form of learning, inference, or decision making."

More recently, [Knutov, De Bra, & Pechenizkiy, 2009] proposed that the "core" of adaptation be defined by posing and answering six major questions (see also Figure 5 that depicts the authors' view of how these questions relate to AH methods and techniques):

- What can we adapt? (What?)
- What can we adapt to? (To What?)
- Why do we need adaptation? (Why?)
- Where can we apply adaptation? (Where?)
- When can we apply adaptation? (When?)
- How do we adapt? (How?)

As we will see in the next section, most of these questions also constitute dimensions along which one can classify and categorize the different incarnations of adaptation in a variety of interactive systems and application domains.

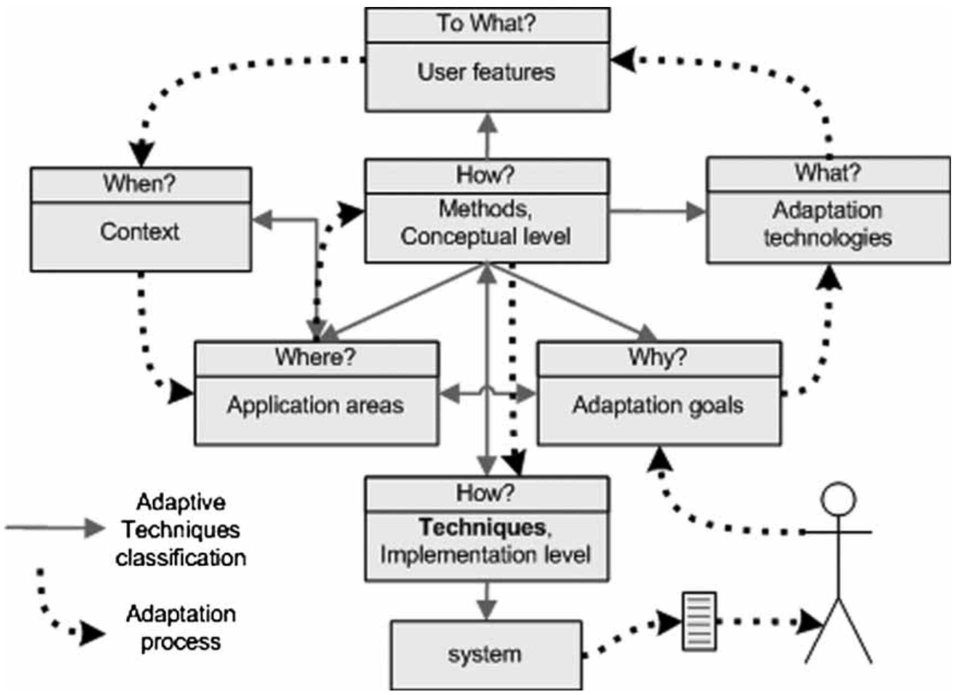


Figure 5: Classification of AH methods and techniques, adaptation process highlights [Knutov, De Bra, & Pechenizkiy, 2009].

A.1.2.2 Adaptation Taxonomies

A simple early taxonomy proposed by Totterdell [1990] suggests that an AUI has the capacity to modify itself as a result of the "experience" it gathers from the users' interacting with it. Three types of AUI are identified from that basis: (a) those that collect information about the user and tailor the dialogue and their responses accordingly, either within an interactive session, or between interactive sessions; (b) those that classify the user into a specific category and then tailor the interface to accommodate the characteristics of that category (a process that takes place only once); and, (c) those in which the user's dialogue with the system is not modified, but where the system's performance increases over time (e.g., in handling user errors in the interaction with the system).

Totterdell & Rautenbach [1990] present a more "complex" taxonomy, which relates the possible behavior / strategy of a player in the classical Artificial Intelligence problem "Prisoner's Dilemma" with the behavior of living organisms at various levels of evolution, and, finally, with the characteristics of adaptive systems (see Table 1). This taxonomy is particularly interesting, as it shows that adaptation in interactive software systems is still at its early stages (mostly in the area of "Adaptable / Tailorable" and "Adaptive") and that many challenging steps remain before we can attain (self-) adapting systems with which we will be able to interact in ways similar to the ones that humans use to communicate with each other. Totterdell & Rautenbach [1990] also use the classification of the above taxonomy to identify the roles served by the designer of a system, the end user of the system, and the system itself in the "design" of the final interface that the user interacts with (Table 2). The roles of the preceding "design agents" are identified in relation to three design facets: who is responsible for introducing any variation capability in the system, who performs the selection of the variants, and who tests the appropriateness of the selected variants for their intended purpose. What is interesting in this further elaboration of the taxonomy is the elevation of the system into the main agent for selecting / effecting and testing interaction variants, as we go further up the "evolutionary" path of adaptation.

With reference to the above taxonomy, this thesis is mainly concerned with adaptive and self-regulating systems. Note that, the term "adaptable", as used in the taxonomy does not entirely coincide with the interpretation adopted herein. The employed semantics of the term would make it overlap with the "adaptable / tailorable" and "adaptive" categories of the taxonomy.

Dieterich et al. [1993] present an alternative perspective on AUI taxonomies. In essence, they differentiate them according to which one might classify AUI and focus specifically on five of these dimensions: (a) the phases of adaptation and the entities responsible for carrying them out; (b) the characteristics of the user interface that participate in / are affected by the adaptations; (c) the information that is taken into account for deciding upon adaptations; (d) the adaptation goals; and, (e) the adaptation strategy employed.

Table 1: Levels and features of adaptation in computer systems as compared to: (a) different strategies in the Prisoner's Dilemma game; (b) evolutionary processes. (adapted from [Totterdell & Rautenbach, 1990])

Prisoner's Dilemma	**Evolution**	**Features**	**Computer Systems**
Nasty / Friendly	Natural selection	Selection by external agent	Designed systems
Player selects from range of strategies	Genetic engineering	Deferred selection	Adaptable / Tailorable
Tit for Tat	Tropism / reflexes	Apparent learning (i.e. fully determined by design)	Adaptive
Learner	Operant conditioning	Learning; varied responses selected for different situations, evaluation by trial and error	Self-regulating
Modeller	Internal evaluation	Planning; problem solving; rule mediated representation; initial evaluation internal to the system	Self-mediating
Introspector	Abstraction	Evaluating the evaluation; generalisation; meta knowledge	Self-modifying

Levels

Table 2: Agents (designer, user, or system) responsible for the design facets of variation, selection, and testing, in the different levels of adaptation in computer systems. (adapted from [Totterdell & Rautenbach, 1990])

	Design Facets		
Level of system	***Variation***	***Selection***	***Testing***
Designed	*Designer*	*Designer*	*Designer*
Adaptable / tailorable	*Designer*	*User*	*Designer*
Adaptive	*Designer*	*System*	*Designer*
Self-regulating	*Designer*	*System*	*System*
Self-mediating	*Designer*	*System*	*System*
Self-modifying	*System*	*System*	*System*

Perhaps the most interesting of these five dimensions is the first, as it defines to large degree the adaptation capabilities of an adaptive system. Classification along this dimension utilizes in fact two complementary sub-dimensions: the *phases* of adaptation and their *agents*. The phases identified are (see Figure 6): (a) "initiative", i.e., the identification of a need or opportunity for adaptation; (b) "proposal", i.e., the identification of alternative possible adaptations and their characteristics (the term "proposal" is used to suggest that the system might "propose" these alternatives to the user); (c) "decision", i.e., the selection among the possible alternatives and / or their characteristics; and, (d) "execution", which refers to the actual enactment of the adaptation. Along the sub-dimension of "agents" the most interesting ones[6] identified are the end user and the system itself (see Figure 6).

	System	**User**	
Initiative	●		System initiates adaptation
Proposal	●		System proposes some change / alternatives
Decision		●	User decides upon action to be taken
Execution	●		System executes user's choice

Figure 6: Phases of adaptation and their agents: example distribution from [Dieterich et al., 1993].

On the basis of these dimensions Dieterich et al. [1993] arrive at a classification scheme for adaptation-capable user interfaces, which is presented in Figure 7. This scheme positions instances of combinations of responsibility allocations along two dimensions of system intelligence: for proposal generation and evaluation; and, for context analysis and plan recognition. Of particular interest in the context of this thesis are the "self-adaptation" instances. It is noteworthy that these three instances of self-adaptation differ in whether the initiative for adaptation, and the final decision as to whether an adaptation get applied, lies with the user or with the system. The "proposal", however, i.e., the capability to determine what adaptations are applicable and potentially optimal for a given set of constraints, is the responsibility of the system. The same is true for the "execution" of adaptation, which is similarly a system task.

Of these three types of self-adaptation, "pure" and "user-controlled" self-adaptation is what most of the work reported herein has been concerned with.

[6] Other agents, such as the system designer, cannot participate in the adaptation process *dynamically*, which is the focus of the taxonomy being discussed.

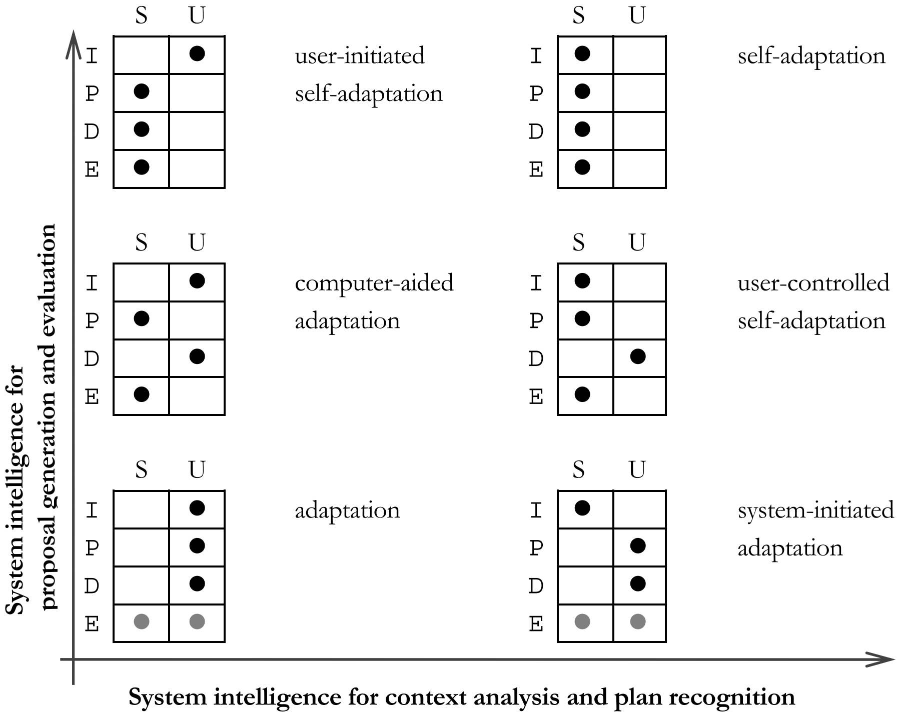


Figure 7: Phases of adaptation and their agents: categorization scheme [Dieterich et al., 1993].

The classification of adaptation-capable systems in general (and user interfaces in particular) can also be done on the basis of differentiating factors other than the ones addressed above. Some examples in this category include (note the relation to the questions in [Knutov, De Bra, & Pechenizkiy, 2009] discussed earlier and depicted in Figure 5):

- The adapted constituents (i.e., which parts of a system are adapted) – Answering the question "What?" (Figure 8).
- The specific methods used in adapting the interaction dialogue – Answering the question "How?" (Figure 9).
- The adaptation determinants (i.e., the types and sources of information utilized for adaptation decision making) – Answering the question "To What?" (Figure 10).
- The timing of adaptation in relation to the interactive sessions – Answering the question "When?" (Figure 11).
- The elements of the system's structural model – Answering, in part, the question "To What?" (Figure 12).
- Etc.

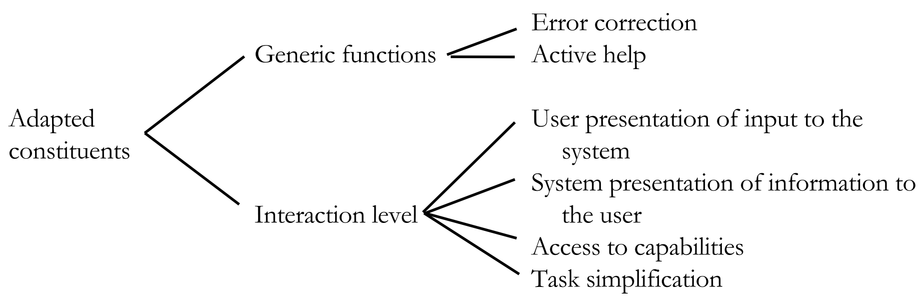


Figure 8: Adapted constituents in AUIs, according to [Dieterich et al., 1993].

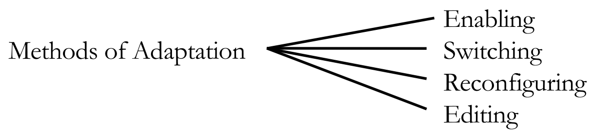


Figure 9: Methods of adaptation in AUIs, according to [Dieterich et al., 1993]

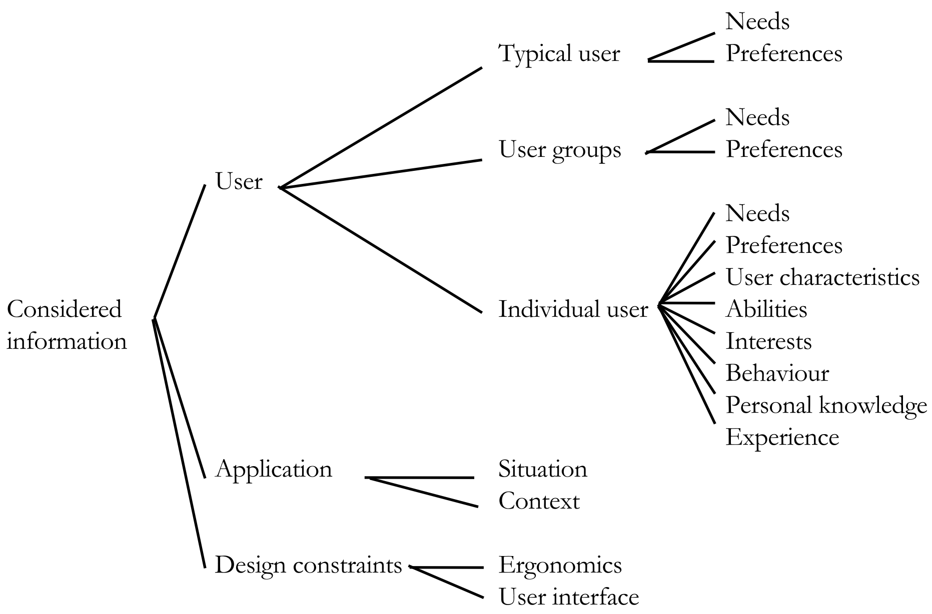


Figure 10: Adaptation determinants (information considered to decide upon adaptation) in AUIs, according to [Dieterich et al., 1993].

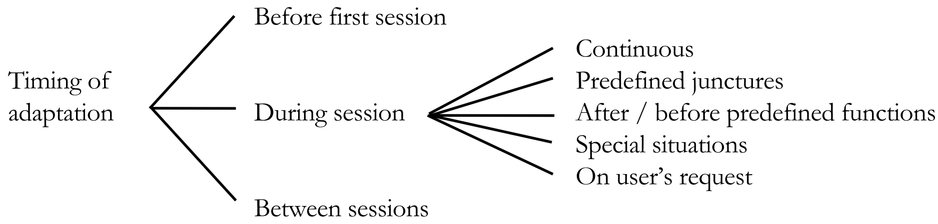


Figure 11: Timing of adaptation in AUIs, according to [Dieterich et al., 1993].

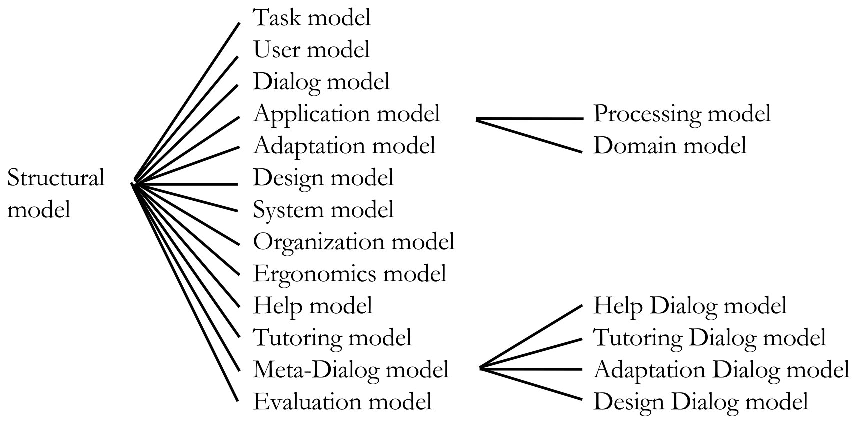


Figure 12: AUI structural model elements, according to [Dieterich et al., 1993].

The classifications and taxonomies we have encountered until now have been either general in nature, or stemming from the domain of adaptive interaction. Further classification is of course possible when one narrows down their view to a specific type of adaptive system. As an example we will consider the case of Adaptive Hypermedia (AH), where a seminal paper by Brusilovsky [1996] introduced a taxonomy of adaptation methods and techniques specifically intended for hypermedia systems. This taxonomy, further elaborated upon in [Brusilovsky, 2001] and, most recently, in [Knutov, De Bra, & Pechenizkiy, 2009], classifies techniques that can be employed for adapting content, presentation, and navigation within hypermedia systems. The most recent version of the said taxonomy, depicted in Figure 13 incorporates practically all the adaptation techniques in Adaptive Hypermedia Systems (AHS) thus far reported in the literature, and serves as an excellent overview of what can be achieved in web-based adaptive systems today.

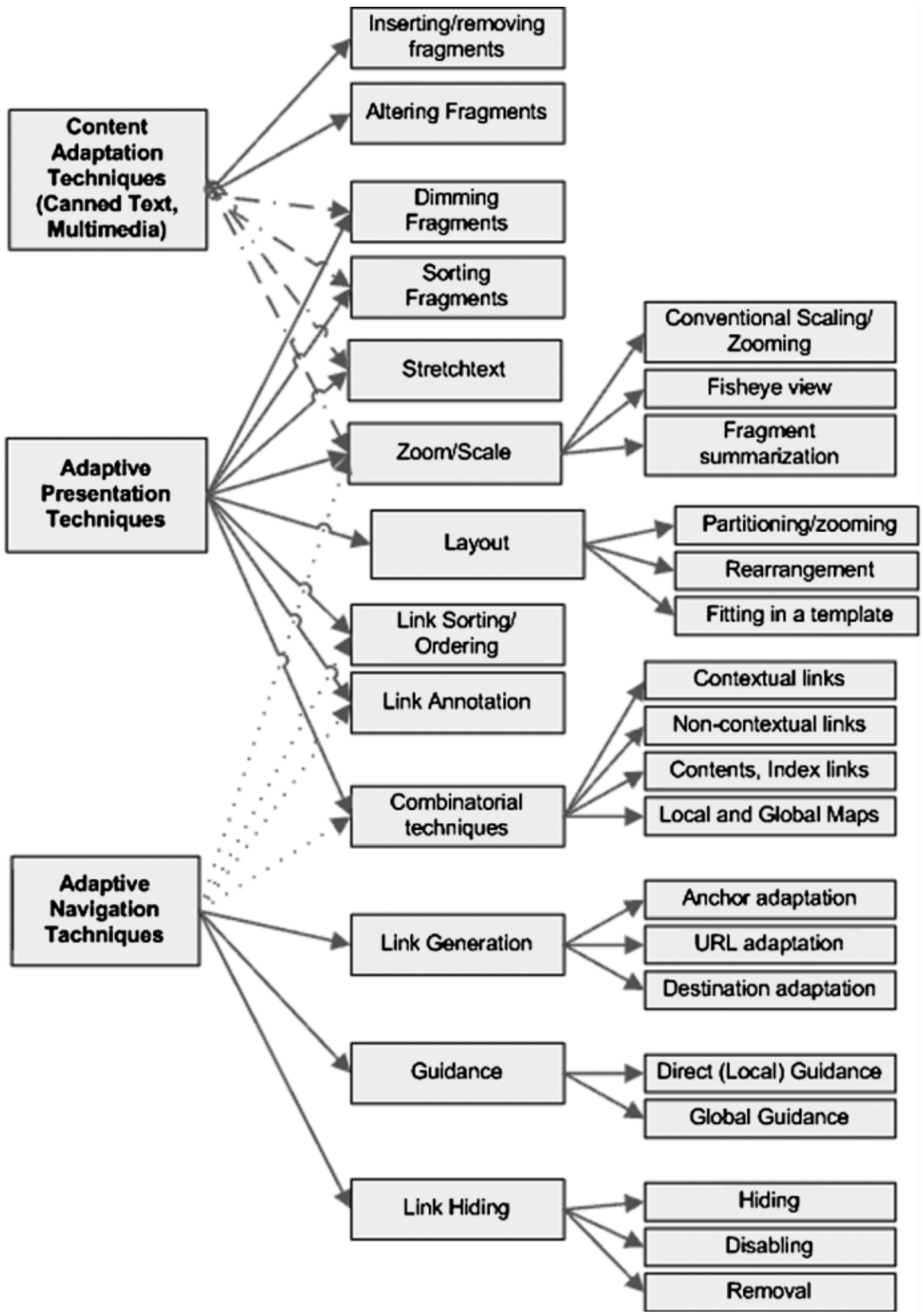


Figure 13: Taxonomy of adaptation techniques in AHS [Knutov, De Bra, & Pechenizkiy, 2009]

In concluding this section, we will revert again to a more general level of classification, to consider the categorization of adaptive systems proposed by

Jameson [2008], distinguishing systems by the intended "function" (purpose) of adaptation. The functions enumerated include [Jameson, 2008]:

- Supporting system use – Taking over parts of routine tasks; adapting the interface; mediating interaction with the real world; giving advice about system use; controlling a dialogue.
- Supporting information acquisition – Helping users to find information; tailoring information presentation; recommending products; supporting collaboration; supporting learning.

Although the above enumeration of functions is not necessarily complete (e.g., adaptation for the purpose of accessibility, which will be discussed in chapter "B.1 The AVANTI Adaptive Web Browser" is not included), Jameson's categorization is widely used today as a high level approach to classifying adaptive systems in relation to the goals of employing adaptation in a system in the first place.

A.1.3 Potential Problems and Challenges

The discussion of adaptation until now has mainly focused on the potential and established benefits of introducing adaptation in an interactive system. In this section we will instead direct our attention towards potential problems and challenges adaptation may incur.

To start with, there is dissent in the scientific community as to whether system intelligence and autonomy with respect to interaction are indeed desirable traits. A classic debate on the subject between Schneiderman and Maes (reported in [Shneiderman & Maes, 1997]) set forth a number of points often voiced against adaptive systems[7]: that well-designed visualization interfaces are better suited for exploring rich information spaces; that when the system takes over initiative and responsibility, the user actually becomes "dumb" (in that the user doesn't know any more how to perform certain tasks, or how the tasks are carried out at all); and that the user has to give up control over the functioning of the system. These points are of course valid, but represent a strict view of a computer as an entirely deterministic tool, like

[7] In the debate the term "interface agents" is used instead of "adaptivity", but the definition provided coincides with the definitions we have seen until now [Shneiderman & Maes, 1997, p.49]:

> "In particular, the way in which [software] agents differ from the software that we use today is that a software agent is personalized. A software agent knows the individual user's habits, preferences, and interests. Second, a software agent is proactive. It can take initiative because it knows what your interests are. It can, for example, tell you about something that you may want to know about based on the fact that you have particular interests. Current software, again, is not at all proactive. It doesn't take any initiative. All of the initiative has to come from the user. A third difference with current software is that software agents are more long-lived. They keep running, and they can run autonomously while the user goes about and does other things. Finally, software agents are adaptive in that they track the user's interests as they change over time."

any other tool that supports human activities in our physical environment. Arguably, this is a perspective that obscures and precludes the many advantages and adaptivity can bring to the table.

Undoubtedly the workings of a personalized system differ fundamentally from the way a traditional, non-adaptive system works. However, most people are unfamiliar with personalization and have no mental model of these systems. When encountering a novel system for the first time, users will unsurprisingly think it works like a system that they are already familiar with [Collins & Gentner, 1987]. Thus, in the case of personalization, they may not understand that a user model is being generated, or that this model is used to base personalized output upon. However, knowledge of how a system works can be beneficial for users when learning how to complete complicated system procedures [Kieras & Bovair, 1984]. For example, the success of inspecting and altering a user model (a subject we will return to later in this section) can be increased when users understand exactly why they are doing this. Furthermore, knowledge of the purpose of this action may work as an incentive to perform this action and in the case of personalized systems, a correct mental model can make a user more tolerable of incorrectly personalized system output [Schmidt-Belz, 2005]. Especially in the immediate future, while personalization is still a novelty to most users, the absence of an appropriate mental model needs to be accounted for in system design.

Going deeper into this topic, one can easily see that the users' mental model of a system directly affects their expectations for the system. The current paradigm / mental model that most users have for software systems, is that of "tools", in the sense of, for instance, construction equipment. This metaphor highlights a number of important traits that people often assign to software. Among them are perfect and unequivocal controllability and predictability, as well as user-managed flexibility – for some categories of tools at least. This notion of software does not accommodate the possibility of the system taking initiative under any circumstances. Nor does it allow for anything other than entirely deterministic system behavior, from the perspective of the user. These, however, are exactly the premises of adaptivity and personalization, which inevitably leads to tensions (e.g., when searching, users would both like the best result for them personally to appear at the top of the list of search results, but, at the same time, they would also like the process to be deterministic).

In contrast to the "tools" model, a more desirable mental model for users to have when assessing an adaptive system is that of a communication partner (akin to a human partner) available to assist the user in achieving a goal. There are several important elements in this paradigm [Paramythis & Van Velsen, 2009]:

- Predictability and controllability notions are "relaxed": when it comes to human communication partners, people expect to be able to anticipate the general disposition of others on a given topic, but they have no expectation of total determinism when concrete behavior in relation to said topic is exhibited.

- Variations in behavior are more readily accepted: people change their minds as additional information and experience becomes available to them; in fact this is considered a trait rather than a liability.
- Slow introduction of acceptable personalized behavior appears more natural, as well as the notion that the user may need to actively contribute to improving the situation: it is understandable that someone may take some time (and even make a few mistakes) until they "know" their communication partner.

Recent studies have shown that, even when users are not able to understand exactly how the system worked, but only grasp the big picture behind it, this still appears to contribute to their feeling of control over the system, and may also influence positively their general attitude towards a system [Paramythis & Van Velsen, 2009]. For example, providing the system with private data may be acceptable exactly because users understand in broad strokes how it is used to better tailor the system to them.

Instilling a "communication partner" mental model to an adaptive system's users will obviously be easier when the system replicates behavior typically associated with humans – a major example being recommender systems.

Against this background, let us return now to the points raised in [Shneiderman & Maes, 1997]. To start with, new forms of interactive, direct-manipulation visualizations are not mutually exclusive with adaptivity. Instead, such visualizations can be enhanced through augmentation on the basis of a user's model. Furthermore, there are information spaces that are simply too large for straightforward visualization of the entire space (e.g., containing millions of data points); in such cases, adaptivity may be indispensable in constraining the space so that the resulting sub-space can then be meaningfully visualized and interacted with.

The users' becoming unfamiliar with tasks as a result of them being undertaken or largely supported by adaptive system behavior, is a better substantiated concern, which does potentially engender problems. This is definitely the case when the said tasks are central to the development of cognitive or other skills important to everyday life. At the same time, however, it may be worth considering that the changing nature of computer-supported activities may indeed make it inevitable that humans relinquish certain skill sets and obtain new ones in their transition into the Information Society. This is, in fact, an often repeating pattern in human evolution and history. For instance, the introduction of written language, and the invention of the printing press, have both signified immense departures in how human knowledge is recorded, propagated and used, and have obsoleted what may have been considered fundamental skills before the respective innovations. Nonetheless, modern society is not necessarily worse off for the lack of individuals trained to recite epic poems from memory, or ones trained in calligraphy and drawing for reproducing texts by hand. These are admittedly major paradigm shifts, and not always applicable to the situations where adaptivity is applied. It would appear advisable then to consider the potential detrimental effects of adaptation on a per case basis, and balance them against the expected benefits – or, even decide whether these benefits are desirable to start with.

The third point discussed in [Shneiderman & Maes, 1997] concerned the lack of user control in the presence of automatic personalization. Again, this is a well substantiated problem that needs to be explicitly addressed in the design of adaptation. As we have already seen in the previous section, one approach would be to let the user be responsible for either taking the initiative, or for deciding upon adaptations. Although effective, this approach is not without its problems. On the one hand, the user may not be the best agent to determine when and what type of adaptation may be possible and beneficial to them. On the other hand, asking the user to continuously make decisions with respect to adaptation may result in an overly obtrusive interactive experience. An alternative approach, which we will discuss later in this section, is to introduce the means into the system that would allow users to observe, understand and directly influence adaptation. This approach is more in line with the proposed new paradigm of interaction put forward earlier, and strives to: (a) acquaint the user with the system's behavior and its "rationale"; and, (b) control the system's behavior at a level above the individual instantiations of that behavior.

A more recent debate on the "wisdom of personalization" in [Ashman, Brailsford & Brusilovsky, 2009] brings forth a different set of points that may potentially render personalization "harmful": security and data privacy; reliability; human implications of personalization; and, inconsistent presentation and outputs. We will discuss each of these points in turn.

Security and privacy are indeed a major concern in adaptive systems, exacerbated by the proliferation of centralized user models and user profile sharing between online services. [Kobsa, 2007] provides a thorough overview of the issues involved, legal and other regulations that have direct implications on these issues, and possible approaches to reconciling the tension between personalization and privacy. Although it is beyond the scope of this thesis to discuss all facets of the potential problems, it is worth highlighting the main factors that make adaptive systems more susceptible to breaches, and the breaches themselves more detrimental, when compared with traditional interactive software. Firstly, as we have already seen, adaptive systems require much more detailed information about the user than their traditional counterparts. There already exist lively discussions in the community about the ownership of that data, and the ways in which the entity that maintains may share the data with associated entities or "the rest of the world", either intentionally or inadvertently [Ashman, Brailsford & Brusilovsky, 2009].

Secondly, it has been shown that even data that is considered anonymized (i.e., stripped of information that could be used to identify individual users) can be combined with publicly available information for purposes of de-anonymization; [Narayanan & Shmatikov, 2008] demonstrates this using recommender system data, whereas [Narayanan & Shmatikov, 2009] demonstrates de-anonymization using social network graphs. Thirdly, the adaptation algorithms and logic themselves may serve as an attack vector against adaptive systems. This is not just a theoretical possibility, as the widely publicized case of an attack against Amazon.com illustrates [Williams, Mobasher & Burke, 2007], and has attracted significant attention in recent years (see, e.g., [Mobasher et al., 2007]). Although there exists no "silver bullet" for the protection of privacy and security in adaptive systems, there do exist strategies that, when carefully applied, can, in combination, reduce the risks involved [Kobsa, 2007].

The question of reliability concerns the accuracy of the user models and algorithms used in user-adaptive systems. Although there are many different approaches to user modeling, most of these work by categorizing the users' knowledge, interests, goals, background and individual traits in some way [Brusilovsky & Millán, 2007]. Although several of these, and corresponding adaptation algorithms, have been tested in practice, and are highly regarded by adaptation practitioners, arguably none of them is perfect. This gives rise to the question of how reliable a system with an (even partially) incomplete or incorrect user model is, and how reliable algorithms that operate on such model data may in turn be [Ashman, Brailsford & Brusilovsky, 2009]. The answers to these questions depend on the nature of the system, the data, the algorithm, the adapted constituents, and potentially other factors as well. In the last decade and a half, *transparency* has been gaining ground as a means of empowering users to determine and improve system reliability themselves. Transparency here refers to the system's capacity to present a user's model to the user for inspection [Höök et al., 1996]. This can take several forms, from an explicit representation of the model, where the later is simple enough to be comprehended by the end user, to implicit simplified versions aiming at understandability without exposing the system's complexity to the user (e.g., enumerating the adaptation determinants, as done in the case of the Amazon.com recommender). User model *scrutability*, a sibling concept to transparency, further proposes that the user model be manageable and modifiable by the end users, an approach which is applied with increasing frequency and has the additional benefits of ensuring higher levels of user control over adaptation and results in higher trust in the system [Kay, 1995].

In discussing human implications of personalization, Ashman, Brailsford & Brusilovsky [2009] raise the question of whether it is indeed desirable to rely on the benefits of adaptive systems under certain circumstances. The exemplary case offered is the adaptive support for learning, which may result in learners relying on the system for tasks that would normally foster the development of a skill set for acquiring and synthesizing knowledge, or diminish the serendipitous exposure of learners to alternative beliefs, lifestyles, culture, etc. This argument is directly related to the one about making users "dumb" discussed earlier. As already pointed out in that case, this is a real concern that the design of adaptation needs to be balanced against.

The final criticism put forth in [Ashman, Brailsford & Brusilovsky, 2009] is that personalization in adaptive systems inevitably results in inconsistent presentation and outputs amongst users, which may, in turn, have detrimental effects, such as giving rise to the potential for inequality and inequities in opportunity. This criticism is, of course, not without base, but again exhibits a strict view of "computers as tools". In Chapter "B.1 The AVANTI Adaptive Web Browser" we will see a system that was explicitly designed to have "inconsistent outputs" for different categories of users, so as to accommodate the different (dis-)abilities of each category. Even more, it was designed to have "inconsistent presentation" for individual users, to accommodate (among many other factors) their increasing level of familiarity with the system. This does not mean, however, that adaptation design does not need to heed this problem. Of particular danger are adaptive modifications made within a single interaction session. These need to be introduced in a way that prevents confusion, and is, where possible, gradual, transparent and reversible. This would prevent "vicious cycles" where the user attempts to adjust to changes made in the system, while the system interpret adjustment behavior as evidence for the need for further adaptations.

A different perspective on potential problems emanating from the introduction of adaptivity in a system is taken by Jameson [2008], who identifies a number of usability challenges related to adaptivity, their typical properties, and possible preventive and compensatory measures that can be employed to address these challenges. The challenges are expressed as usability goals to be met and include (see Figure 14):

- *Predictability and Transparency* – The concept of predictability refers to the extent to which users can predict the effects of their actions. Transparency is the extent to which users can understand system actions and/or have a clear picture of how the system works.
- *Controllability* – Controllability refers to the extent to which users can bring about or prevent particular actions or states of the system if they have the goal of doing so.
- *Unobtrusiveness* – The term obtrusiveness refers to the extent to which the system places demands on the user's attention which reduce the users' ability to concentrate on their primary tasks.
- *Privacy* – Privacy refers to the protection of data that the system collects about individual users from unauthorized access, and improper use.

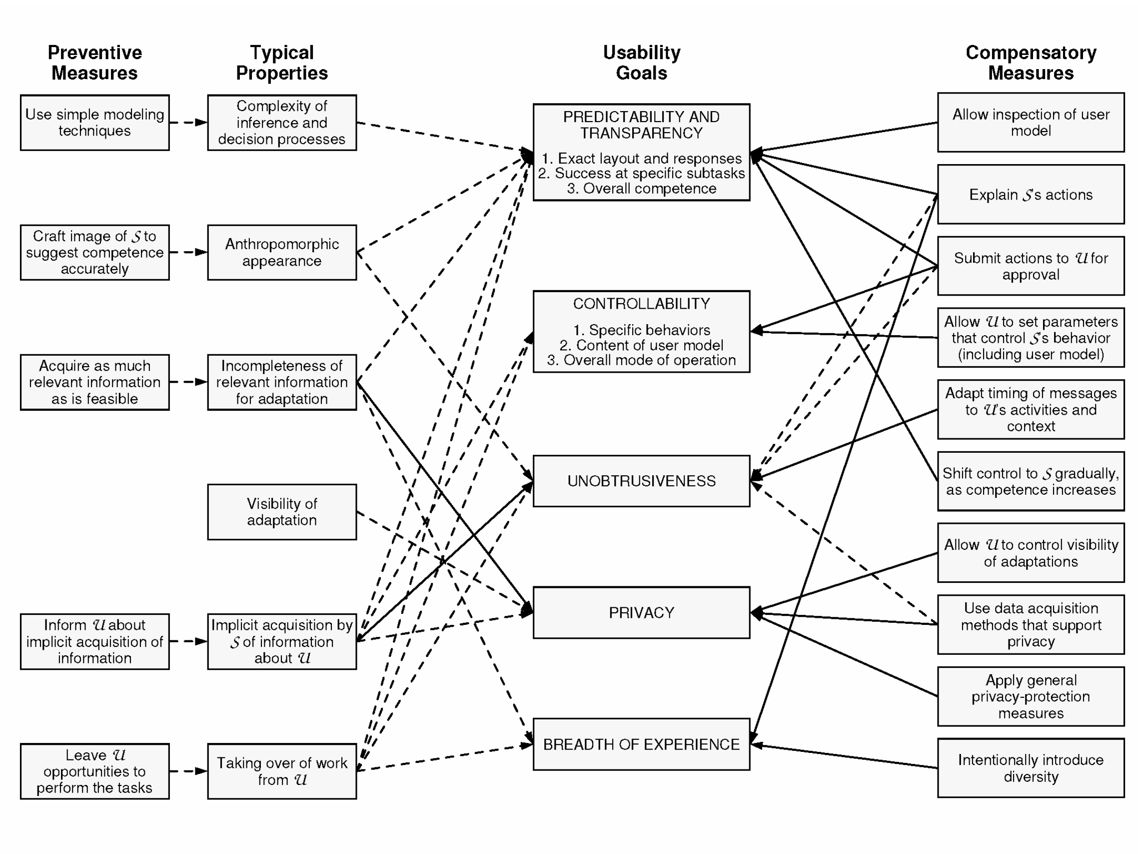


Figure 14: Overview of usability challenges for user-adaptive systems [Jameson, 2008]

- *Breadth of experience* – This refers to the potential problem of users trusting entirely the system's adaptivity in dealing with an application, knowledge, information, or other domain, with the result being that the users end up with a limited view and poor understanding of the domain itself.

Of particular interest in this list of challenges is breadth of experience, which we have not explicitly discussed before. This is a special case of "human implications of personalization", and of the user becoming "dumb" that were addressed above, but merits special attention because it is possibly the case with the most easily identifiable side effects in a user's interaction with an adaptive system. One can observe that this phenomenon is already very real in certain areas of adaptation, the most prominent perhaps being recommender systems. An increasing number of users rely exclusively on recommendations made for goods to buy, movies to watch, music to listen to, etc., so that they end up confining themselves in already acquired tastes and preferences, and miss opportunities that lie outside their thus far proclaimed or inferred interest niches. There does not exist yet a principled body of knowledge on how to address this problem, but promising attempts are pointing towards mitigating this problem by explicitly exposing users to parts of the domain that are not in their inferred sphere of interest [Shearin & Lieberman, 2001].

The very real potential problems and challenges discussed in this section point unequivocally towards the need for principled design and evaluation of adaptation. Furthermore, they serve as clear supporting evidence that adaptation is, and needs to be treated as, a multi-dimensional element of interaction, that potentially requires the development and employment of new design and evaluation methods and tools. We will return to this topic in Unit C, "Adaptive System Evaluation", where work will be reported that builds upon these guiding principles.

A.2 Thesis Overview

Having discussed what adaptation in general, and adaptivity in particular entail, and complementary ways of classifying alternative approaches in the area, this section will position the work described in this thesis in the outlined landscape.

This thesis is comprised of selected individual and joint publications, summarizing research work of the author spanning approximately ten years and three thematic areas. The publications do not appear here in their original form. First and foremost, all used material has been edited to include only work undertaken or led by the author. Secondly, the content has been modified to attain cohesion in the presentation of interrelated elements of research. Thirdly, comparisons of the presented results with the current State-of-the-Art in the corresponding areas of work were added where appropriate.

In total, this thesis is primarily based on material from seven publications, which span three thematic areas: the design and implementation of architectures and frameworks for desktop- and web- based adaptive systems; the evaluation of adaptive systems; and, finally, the introduction of meta-adaptive capabilities to next-generation adaptive systems. Each of these areas is briefly outlined below.

This section is concluded with an account of the work / employment context and the main research projects in which the presented pieces of work have been carried out.

A.2.1 Main Areas of Work

Architectures and Frameworks

The second Unit of this thesis, "Adaptive System Architectures and Frameworks" presents work addressing the design and implementation of architectures and frameworks for desktop- and web- based adaptive systems.

The desktop-based framework was employed in the development of an extended web browser, which utilizes adaptability and adaptivity techniques to tailor itself to the abilities, skills, requirements and preferences of individual users, the different contexts of use, and the changing characteristics of run-time interaction between the user and the system.

The web-based framework supports several of the adaptation techniques for AHS reported in the literature, while remaining orthogonal to web "serving" approaches, and poses only minimal requirements in that direction. It was employed in the development of an information system capable of tailoring interaction and information content to the needs of its end users.

Evaluation

The third Unit of the thesis, "Adaptive System Evaluation", addresses one of the difficult problems in the area, namely the evaluation of adaptation in a way that provides sufficient design feedback for the identification and remedying of problems arising therefrom.

Traditionally, adaptive systems were evaluated using methodologies, techniques and assessment tools intended for general interactive systems. Experience, however, has shown that the dynamic nature of adaptive system behavior sometimes leads to incorrect results, and, more often than not, renders it impossible to pinpoint the elements of adaptation that have their desired effects.

The work presented in this thesis delineates the problem in detail, and then goes on to propose approaches that can be used to address it. Specifically, it is proposed that evaluation treat adaptation as a multi-dimensional quantity, and that assessment targets individual stages in the adaptation process. The work presented has formed the basis for an evaluation framework that is currently considered to represent the State-of-the-Art in the area.

Meta-Adaptation

The fourth and final Unit of the thesis, "Meta-Adaptive Systems", focuses on concrete steps towards the attainment of "next generation" adaptive systems, characterized by their ability to model, reason about, assess and modify their own adaptive behavior. These meta-adaptive systems can, in effect, learn and apply adaptation strategies that have not been designed into them, but were rather built dynamically on the basis of evidence collected from the users' interaction with the system.

The work presented here sets the basis for designing and developing self-regulating adaptive systems, which are a sub-category of meta-adaptive systems. Self-regulation exhibits the characteristics of meta-adaptation briefly outlined above, but does not require that the system be capable of synthesizing new adaptation strategies, but rather be capable of learning how and when to use existing strategies to achieve specific adaptation goals. The work presented here arguably constitutes a significant step in the evolution of adaptive systems, and represents the State-of-the-Art in the area.

A.2.2 Work Context and Research Projects

Some of the work presented in this thesis was carried out while the author was employed at the Human-Computer Interaction Laboratory (HCI Lab), of the Institute for Computer Science, Foundation for Research and Technology – Hellas (FORTH-ICS). The HCI Lab is headed by Prof. Constantine Stephanidis, who was the scientific responsible for the corresponding research activities, and provided valuable guidance along the way. The aforementioned research activities were carried out in the context of two major European Commission funded projects, namely ACTS AC042 AVANTI, and IST-1999-20656 PALIO which are described in more detail below.

The rest of the work presented here was carried out while the author was employed at the Institute for Information Processing and Microprocessor Technology (FIM Institute) of the Johannes Kepler University (JKU). The FIM Institute is headed by o. Univ. Prof. Dr. Jörg R. Mühlbacher, who was the scientific responsible for the corresponding research activities (primarily carried out in the context of the projects "Integrating Agents into Teleteaching Webportals" ad "Adaptive Learning Spaces" described below), and has also overseen this thesis.

The AVANTI project

The ACTS AC042 AVANTI ("AdaptiVe and Adaptable INteractions to Multimedia Telecommunications ApplIcations") project (1996-1998), was partially funded by the European Commission (DG XIII). The AVANTI consortium comprises: ALCATEL Siette (Italy) – Prime contractor; CNR-IROE (Italy); ICS-FORTH (Greece); GMD (Germany); University of Sienna (Italy); MA Systems (UK); MATHEMA (Italy); VTT (Finland); ECG (Italy); University of Linz (Austria); TELECOM ITALIA (Italy); and, EUROGICIEL (France).

The main objective of the AVANTI project was to demonstrate that it is possible to develop generic multimedia telecommunications applications, which are adaptable and adaptive to the requirements of most potential users including disabled people, elderly people, occasional users and professionals. Adaptation was applied in the system at both the user interface and the content levels in order to optimize the way the information was compiled and presented, and how it was delivered over the network.

The author was involved in the design and implementation of the adaptation infrastructure of the project's user interface / browser, which is described in detail in chapter "B.1 The AVANTI Adaptive Web Browser". In this context, the author worked under the guidance of Prof. Constantine Stephanidis, and collaborated closely with the following members of the HCI Lab of FORTH-ICS: Athina Stergiou, Napoleon Maou, Adam Leventis, Anthony Savidis, and George Paparoulis.

The PALIO project

The IST-1999-20656 PALIO ("Personalised Access to Local Information and Services for Tourists") project (1999-2002) was partly funded by the Information Society Technologies Programme of the European Commission – DG Information Society. The partners in the PALIO consortium were:

ASSIOMA S.p.A. (Italy) – Prime Contractor; CNR-IROE (Italy); Comune di Firenze (Italy); FORTH-ICS (Greece); GMD (Germany); Telecom Italia Mobile S.p.A. (Italy); University of Sienna (Italy); Comune di Siena (Italy); MA Systems and Control Ltd (UK); and, FORTHnet (Greece).

The PALIO project addressed the issue of Universal Access to community-wide services, based on content- and interface- level adaptation, beyond desktop access. The main challenge of PALIO was the creation of an open system for the unconstrained access and retrieval of information (i.e., not limited by space, time, access technology, etc.). Under this scenario, mobile communication systems play an essential role, because they enable access to services from anywhere and at anytime. One important aspect of the PALIO system has the support for a wide range of communication technologies (mobile or wired) to facilitate access to services.

The author was involved in the design and implementation of the adaptation infrastructure of the project's software platform, which is described in detail in chapter "B.2 A Generic Adaptation Framework for Web-based Hypermedia Systems". In this context, the author worked under the guidance of Prof. Constantine Stephanidis, and collaborated closely with the following members of the Human-Computer Interaction Laboratory of FORTH-ICS, who were also involved in the development work: Chrisoula Alexandraki, Napoleon Maou, and Ioannis Segkos.

The Integrating Agents into Teleteaching Webportals project

The FWF P15947-N04, "Integrating Agents into Teleteaching Webportals", project (2002-2005) was partially funded by the Austrian Science Fund (Fonds zur Förderung der wissenschaftlichen Forschung – FWF) and was carried out by the FIM Institute, in loose cooperation with external research partners.

The aim of this project was to combine an existing agent platform for mobile agents with an E-Learning platform, both of which were developed at the FIM Institute. This combination yielded a number of tangible advantages for users, including the identification of user interests and the introduction of system support on the basis of said interests (e.g., for automated notifications, filtering of resources, etc.), the provision of support for awareness of peer activities, and other miscellaneous elements of support in the teletaching / telelearning process.

The author was involved in the development activities of this project, led by Dr. Michael Sonntag, and scientifically overseen by o. Univ. Prof. Dr. Jörg R. Mühlbacher. The author's participation concentrated mainly in the integration activities, and specifically on the side of WeLearn (see [Mühlbacher & Putzinger, 2006]), the E-Learning platform employed in the project. In these tasks, the author collaborated closely with Andreas Putzinger, and other members of the FIM Institute.

The Adaptive Learning Spaces Project

The 229714-CP-1-2006-1-MINERVA-M "Adaptive Learning Spaces" (ALS) project (2006-2009) was partially funded by the EC Socrates - Minerva Programme, and overseen by the European Commission's Education, Audiovisual and Culture Agency Executive Agency. The partners of the ALS consortium were: Technical University of Eindhoven (The Netherlands) – Prime Contractor; Warwick University (UK) – General Coordinator; Johannes Kepler University (Austria); National College of Ireland (Ireland); Bauhaus-University Weimar (Germany); Turpin Vision Ltd. (The Netherlands); and, SOFTWIN (Romania).

The goal of the ALS project was to provide a set of technological means through which lack of (or limited amounts of) face-to-face contact between instructors and learners, as well as amongst learners can be partially compensated for. To achieve this, ALS worked towards: (a) widening the range of, as well as increasing the amount of, guidance and support that ODL systems can provide to learners and instructors; and, (b) providing novel means to support the social cohesion of groups of learners, as well as the engagement of their members in collaborative / team tasks and processes. The project's main outputs include the software technologies developed and integrated into the Sakai E-Learning platform[8], as well as a series of 'Best Practices' reports that will accumulate the project's acquired knowledge and experience in integrating these technologies into existing Open and Distant Learning (ODL) and Blended Learning settings.

The author led the activities of the FIM Institute in this project, under the scientific oversight of o. Univ. Prof. Dr. Jörg R. Mühlbacher. This work included the design and implementation of a number of tools that adaptively supported peer awareness, event and resource filtering, etc., as well as the integration of the AHA! adaptive course delivery system (see [De Bra & Stash, 2002; De Bra et al., 2002b]) into the Sakai platform. In these tasks the author collaborated closely with the following members of the FIM Institute: Mirjam Köck, David Hauger, Michael Sonntag, and Florian König.

[8] http://www.sakaiproject.org/

U n i t   B

ADAPTIVE SYSTEM ARCHITECTURES AND FRAMEWORKS

Adaptation for Desktop and Web Applications

This Unit is concerned with the design and implementation of architectures and frameworks to support desktop- and web- based adaptation. The first Chapter, "B.1 The AVANTI Adaptive Web Browser" reports work carried out in the context of the ACTS AVANTI AC042 project, with the goal of providing accessibility and high quality interaction in web-based multimedia applications and services, to people with disabilities. Along these lines, one of the main objectives of the work undertaken within the AVANTI project was the design and development of a user interface that would provide equitable access and quality in use to all potential end users. The work presented concerns the design and development of a specialized user interface component for web browsing, which employs adaptability and adaptivity techniques to tailor itself to the abilities, skills, requirements and preferences of individual users, the different contexts of use, and the changing characteristics of run-time interaction between the user and the system.

The second Chapter, "B.2 A Generic Adaptation Framework for Web-based Hypermedia Systems", introduces a framework intended for facilitating the implementation of web-based Adaptive Hypermedia Systems. The framework supports several of the related adaptation techniques reported in the literature, yet remains orthogonal to web "serving" approaches, and poses only minimal requirements in that direction. As such, it can be easily integrated into existing, non-adaptive web-publishing solutions. This chapter presents in detail several aspects of the framework, and provides an overview of its application in the IST-1999-20656 PALIO project. Furthermore, it discusses some of the lessons learnt from our work on the framework thus far, as well as what we consider the most likely directions of future work in the area.

A comparison between the above two adaptation frameworks is provided at the end of the second chapter in section "B.2.5 A Comparison between the AVANTI and PALIO Adaptation Frameworks".

B.1 The AVANTI Adaptive Web Browser[9]

The User Interface (UI) of the AVANTI information system is a component which provides interactive views of adaptive multimedia web documents. The distinctive characteristic of the AVANTI UI is its capability to dynamically tailor itself to the abilities, skills, requirements and preferences of the users, to the different contexts of use, as well as to the changing characteristics of users, as they interact with the system. The AVANTI UI also features integrated support for various "special" input and output devices, along with a number of appropriate interaction techniques that facilitate the interaction of disabled end-users with the system. The categories of disabled users supported by the system are people with light, or severe motor disabilities, and blind people. As the design of the UI has followed the principles of *design for all* (*user interfaces for all* [Stephanidis, 1995]), inclusion of additional target user groups is facilitated. When functioning as part of the AVANTI system, the UI is externally conceived by the user as a specialized front-end through which access to the information in the AVANTI multimedia databases is achieved. The UI is also capable of functioning as an independent web browser, providing access to traditional web documents to able, motor-impaired and blind people.

The rest of this chapter is structured as follows: It first presents the architecture of the AVANTI UI and its place in the overall architecture of the AVANTI information system. Then, it moves on to describe the special input / output devices supported and the method used for their integration into the system. Subsequently, the methodology used to design the UI is outlined, and its correlation to the UI adaptation capabilities is discussed. Following that, the adaptation mechanism developed is presented, and the distinctive characteristics of adaptability and adaptivity are analyzed. The chapter concludes with a synthetic view of the AVANTI UI, and with a overview of progress in related technology since the reported developments.

B.1.1 Architecture of the User Interface Component

The AVANTI information system comprises five main modules: (i) a collection of multimedia databases which are accessed through a common communication interface (Multimedia Database Interface - MDI) and provide mobility information for disabled people; (ii) the User Modeling Server (UMS), which maintains and updates individual user profiles, as well as user

---

9 This chapter is based on [Stephanidis, Paramythis et al., 1998a]. The R&D work described here has been carried out while the author was employed by the HCI-Lab of ICS-FORTH (Heraklion, Greece), and in the context of the ACTS AC042 AVANTI project (please refer to section "A.2.2 Work Context and Research Projects" on page 32 for additional information).

stereotypes; (iii) the Content Model (CnM), which retains a meta-description of the information available in the system; (iv) the Hyper-Structure Adaptor (HSA), which adapts the information content, according to user characteristics[10]; and, (v) the User Interface (UI) component, which is capable of tailoring itself to individual users.

The requirements of the project dictated the development of a new experimental front-end, which would not be based on existing web browser technology; the main reasons for that were: (i) although commercially available browsers of that time supported customizability through "add-on" components, etc., the level of adaptations planned within the project could not be effected using such approaches (e.g., integrating guidance in system dialogues), and (ii) the accessibility requirements posed by the disabled user categories addressed within the project could not be met, either by then existing browsers in isolation, or through the use of third-party assistive products. To gain a better understanding of the issues involved, the reader is referred to section "B.1.2 Integration of Input / Output Devices", which outlines some of the accessibility requirements of end users, in terms of input and output media and modalities, as well as to sections "B.1.3 Unified Design and Rule-Based Adaptation" and "B.1.4 Adaptation Mechanism", which describe the type and range of adaptations employed in the AVANTI project.

The UI component is composed of six main software modules (see Figure 15):

- The *HTTP communications* module; this is used to communicate with the HSA and the MDI, to retrieve the information content; the HTTP communications module can also be used to communicate with traditional HTTP servers, thus providing full standard browser functionality.
- The *KQML communications* module; this is a module that enables the UI to communicate with the UMS (using the Knowledge Querying and Manipulation Language [Finin et al., 1993]), in order to exchange interaction monitoring information, and inferences about user states and interaction situations respectively.
- The *monitoring* module; the role of this module is to monitor user interaction and dispatch appropriate messages to the UMS. The information sent concerns both lexical and syntactic aspects of the interaction. The communication protocols between the UMS and the UI incorporate negotiation capabilities, so that, at any point in a session, the UMS is sent only information that is necessary for the inferences it attempts to make.

[10] Note that content adaptation is not part of the work presented here. The author's work and contributions were made exclusively in the scope of the AVANTI UI.

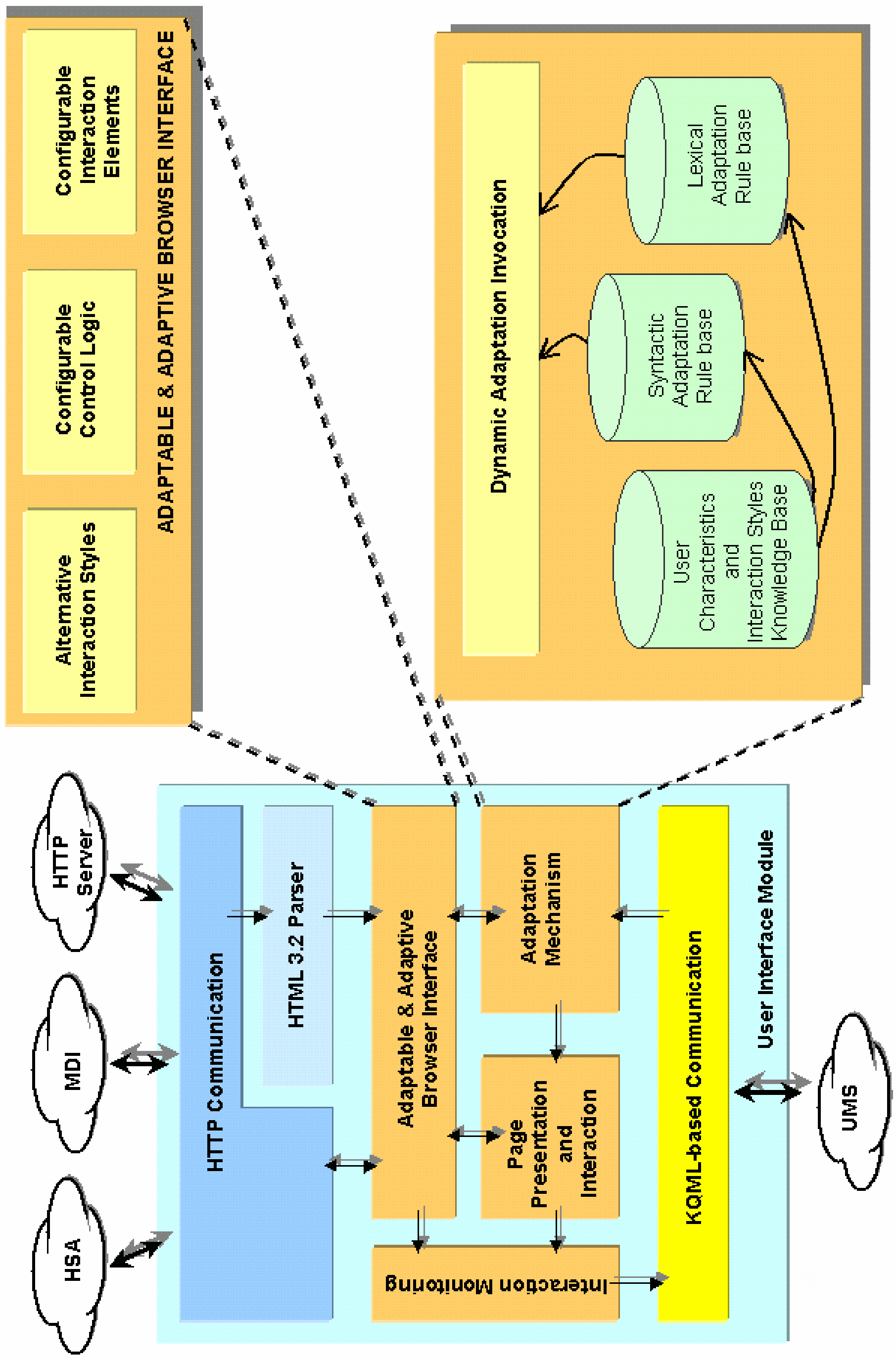


Figure 15: Architecture of the AVANTI UI component

- The *adaptation mechanism* module; this module is responsible for retaining and applying adaptation rules that concern syntactic and lexical, adaptability and adaptivity at the level of the user interface, as well as for maintaining a knowledge space in which static user information and dynamically inferred (by the UMS) user states and interaction situations are held.

- The *adaptable and adaptive browser interface* module; this module is responsible for the presentation of the actual user interface of the AVANTI system. It instantiates the task decomposition and dialogue design, by implementing all the tasks and styles therein. The different dialogue alternatives are selected for execution dynamically, by consulting the adaptation mechanism and receiving appropriate decisions as a reply.
- The *page presentation and interaction* module; this module is responsible for presenting the user with an HTML document and allowing for interaction with the elements contained therein. The modality, as well as other aspects of the presentation, are determined through user characteristics, with the assistance of the adaptation mechanism.
- The *HTML parser* module; this module implements an HTML 3.2 parser, specifically developed to cater for the requirements of the AVANTI system. Special meta-tag syntax has been introduced in the context of the AVANTI system, so that it is possible to affect the presentation of the user interface from within HTML documents (e.g., it is possible to enhance the command toolbar with new buttons and associated commands). Additionally, content tags have been introduced, in order to support the "inline" incorporation of multimedia content (audio and video) in HTML documents. The implementation of the AVANTI user interface did not include support for scripting languages (e.g., Javascript), or extensions to HTML, commonly supported by commercial web-browsing applications (e.g., frames).

B.1.2 Integration of Input / Output Devices

The problems that AVANTI's target user categories face at the terminal level mainly concern: (i) the output devices and the compatibility of the presentation medium; (ii) the input devices and methods; and, (iii) the complex operational procedures required to control the terminal. In order to address these problems within the UI component, special software and hardware modules have been integrated in the terminal. Furthermore, alternative interaction techniques have been built into the user interface, to facilitate the process of controlling the resulting terminal configurations and interacting with the system.

The implementation of the terminal adaptations has adopted an architecture for the integration of special I/O devices, whereby an additional Device Software Layer (DSL) provides a way to uniformly control, and communicate with, special hardware and accompanying software. This software layer also allows the parallel operation of different I/O devices, resolving any potential conflict, and/or communication malfunction. A schematic representation of the adopted architecture can be found in Figure 16.

The above implementation has two advantages. Firstly, multiple instances of the AVANTI UI can share the input and output devices consistently. Secondly, new devices can be easily integrated by enhancing the DSL (e.g., by adding appropriate device structures) and without modifying the UI directly.

Standard I/O devices and systems that are supported by the AVANTI terminals include: keyboard (or any keyboard emulation device), mouse / trackball (or any mouse emulation device), non-speech audio output and touch screen. These are directly controlled by the UI component itself. The special I/O devices and systems supported (and controlled through the DSL) are: Braille display, touch tablet, binary switches, joystick, speech synthesis (output) and speech / command recognition (input).

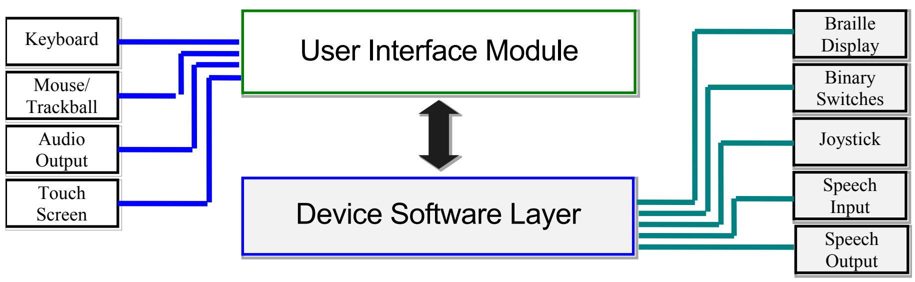


Figure 16: Architecture for the integration of I/O devices

To facilitate the use of the special devices by disabled users, specific interaction techniques have been developed. For example: switch interaction with the interface is achieved through (automatic, or user controlled) scanning and on-screen keyboards; touch tablets can be used by blind users through demarcated areas (raised edges, Braille labels, etc.), each of which corresponds to specific functionality; speech synthesis is used to present textual information to blind users and to signify attributes related to the possible hypermedia nature of the presented documents (e.g., links); speech recognition can be used to allow blind users to issue vocal commands to the system, through a special set of control and navigation commands; gesture recognition permits the use of a joystick by blind users, by coupling specific gestures to command sequences; tactile presentation of hypertext in Braille is augmented with special symbolic annotations, that facilitate the comprehension on the part of the user, of the exact type of item being presented.

B.1.3 Unified Design and Rule-Based Adaptation

The design of the user interface component of the AVANTI system has followed the Unified User Interface Design methodology (UUID), which has been proposed as an efficient and effective method for achieving the goal of user interfaces for all (see [Stephanidis, Savidis & Akoumianakis, 1997], [Stephanidis, 1995]), including disabled and elderly users. Following UUID, only a single unified user interface is designed and developed, which comprises alternative interaction components, appropriate for different target user categories. This single design artifact may have multiple instantiations during initiation of interaction (adaptability), in order to ensure accessibility for a wide range of users. Moreover, each interface instance is continuously enhanced at run-time (adaptivity), in order to provide high-quality of interaction to all potential users (see [Stephanidis & Savidis, 1995]).

Two dimensions of adaptations are addressed within the user interface of the AVANTI system, in relation to: (i) whether adaptations take place during the initiation of interaction (adaptability), or at run-time (adaptivity); and, (ii) the level of interaction at which adaptations are applied, i.e. syntactic and lexical level adaptations. Thus, four types of adaptations can be distinguished: lexical adaptability, syntactic adaptability, lexical adaptivity and syntactic adaptivity.

In the present context, *adaptability* refers to the process of selecting / modifying (aspects of) the user interface during initiation of each interaction session, according to user characteristics that are known prior to interaction (e.g., user abilities) and are assumed to remain unchanged within a single session (e.g., particular user expertise). *Adaptivity*, on the other hand, refers to the process of selecting / modifying (aspects of) the user interface dynamically, according to *dynamic user characteristics* and *situations* that are detected at run-time (e.g., high error rate, inability to complete a task, etc.)

Syntactic level adaptations concern the selection of different *styles* for each abstract interaction task. Following the UUID methodology, the user tasks that can be performed through the user interface of the AVANTI system have been hierarchically structured and incrementally decomposed in a polymorphic fashion, defining alternative styles and task hierarchies, according to requirements and preferences of different user categories. In other words, different styles define alternative ways in which a specific task can be realized. Styles can be either compatible or incompatible to each other (depending on whether they can be simultaneously active), and are synthesized using the operators `BEFORE`, `OR`, `XOR`, `*` (simple repetition) and + (absolute repetition) (see [Savidis et al., 1997]). En example decomposition for the task "Go to Previous Document" is presented in Figure 17.

During the design stage of the user interface, it was found that certain styles exist that need to be included in the decomposition of most of the tasks. These styles are not specific to browsing and can be expected to be equally common in other types of applications. Styles in this category include: (i) *explicit feedback*, either during task performance (interim feedback) or after task completion (completion feedback); (ii) *confirmation*, which may belong to one of two types: either a brief request for explicit approval before the system carries out an action, or a more elaborate explanation of the possible consequences / side effects of the action, in conjunction with the request for approval; (iii) *guidance*, which provides help for the completion of a task (e.g., the sequencing of actions, the types of data required in each field, etc.), when, for example, there is evidence that the user is unable to complete this task; (iv) *prompting*, which provides information concerning the initiation and completion of a specific task, when, for example, there is evidence that the user is unable to initiate this task.

Lexical level adaptations concern the selection of interaction object attributes for each task, or style. In particular, the lexical level interface objects of each style can be instantiated with multiple attributes. The attributes of the interaction objects that are subject to adaptations in the present implementation include scanning (for severely motor-impaired users), font, color and size parameters for the case of visual interaction, and speech, sound and presentation parameters for the case of non-visual interaction. Lexical level adaptations also concern the selection of the appropriate overall metaphor of interaction. Two metaphors have been designed and developed for the needs of the AVANTI project, namely a "Public Information System" and a "Web-Browser" metaphor.

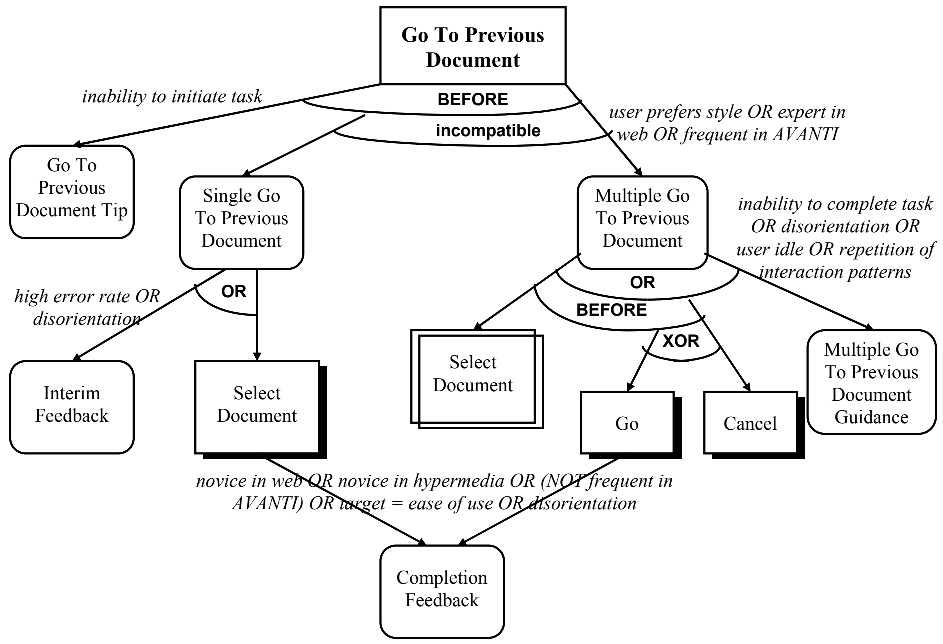


Figure 17: Example of a polymorphic task decomposition

"Static" user characteristics (i.e. characteristics for which knowledge exists prior to interaction), have been selected, after an initial requirements analysis phase, to serve as the basis for adaptability. These include: (i) *physical abilities*, i.e. whether the user is able-bodied, blind or motor-impaired; (ii) the *language* of the user (the system supports English, Italian and Finnish); (iii) *familiarity* of the user with: *computing*, *networking*, *hypermedia* applications, the *web* and the *AVANTI* system itself; (iv) the overall *interaction target*: speed, ease, accuracy, error tolerance; and, (v) *user preferences* regarding specific aspects of the application and the interaction; e.g., whether the user prefers a specific style for a given task; or the preferred speech volume when links are read; etc.

Adaptability Rules

```
IF "user is novice in hypermedia"
   THEN LinkType = Button

IF "user is novice in computing"
AND "user is motor impaired"
   THEN ScanRate = Slow

IF "user is novice in computing"
AND "user is motor impaired"
   THEN Font = Large AND Size = Large

FOR THE TASK "review bookmarks" {
   IF "user unable to complete task"
     THEN ACTIVATE STYLE "review bookmarks
                              with guidance"
}
```

Adaptivity Rules

```
IF "user is motor impaired"
AND ("user has high error rate"
     OR "user is unable to navigate")
   THEN ScanRate = Slow

IF "user is disoriented"
OR "user is idle"
   THEN SpeechVolume = High

FOR THE TASK "review bookmarks" {
    IF "user unable to initiate task"
      THEN ACTIVATE STYLE "awareness notification
                   for review bookmarks facility"
}
```

Listing 1: Examples of adaptability and adaptivity rules

The selection of the above characteristics was made so as to ensure that adequate knowledge exists for the system to cater for a wide range of users, taking into account not only possible disabilities, but also characteristics that differentiate individual users -that may in general belong to the same broad category- between each other. In this version of the system, these characteristics are acquired through an initial "questionnaire" session; more automated solutions were implemented in subsequent versions (e.g., smart-cards). It should be noted, that although these characteristics are uniformly termed "static", they are not all assumed to remain unchanged "permanently". In fact, it is foreseen that future versions of the system will detect and record changes in these characteristics over time, thus causing different adaptations to be effected in the user interface, in terms of adaptability. The *dynamic "user states"* and "*interaction situations"* that are taken into account in adaptivity (also selected during the initial requirements analysis phase) concern: (i) *user familiarity with specific tasks* (capability to successfully initiate and complete certain tasks); (ii) *ability to navigate* (move from one document to another in a consistent way); (iii) *error rate*; (iv) *disorientation* (inability to cope with the current state of the system); (v) *user idle time*; and (vi) *repetition of interaction patterns* (commonly encountered sequences of interaction steps).

A set of syntactic adaptability and adaptivity rules has been defined and associated with each user task, providing the mechanism for the selection of appropriate interaction styles. Lexical level adaptations are also effected through respective rules, that assign different values to the attributes of the realized interaction objects. Figure 17 presents an example task decomposition for a task, namely "Go To Previous Document", together with the syntactic adaptability and adaptivity rules that specify the conditions under which each style is being activated, while Listing 1 presents simplified examples of lexical adaptability and adaptivity rules.

The categories of interface adaptation supported by the AVANTI UI include: (i) support for different *interaction modalities and input / output devices*; (ii) automatic adaptation of the *presentation of interaction elements*; (iii) *task-based adaptive assistance*; (iv) *awareness prompting*; (v) limited support for *error prevention*; (vi) limited support for *metaphor*-level adaptation. Table 3 summarizes the dependency between user-, and usage context knowledge and the adaptations that are triggered upon that knowledge in the AVANTI UI component.

B.1.4 Adaptation Mechanism

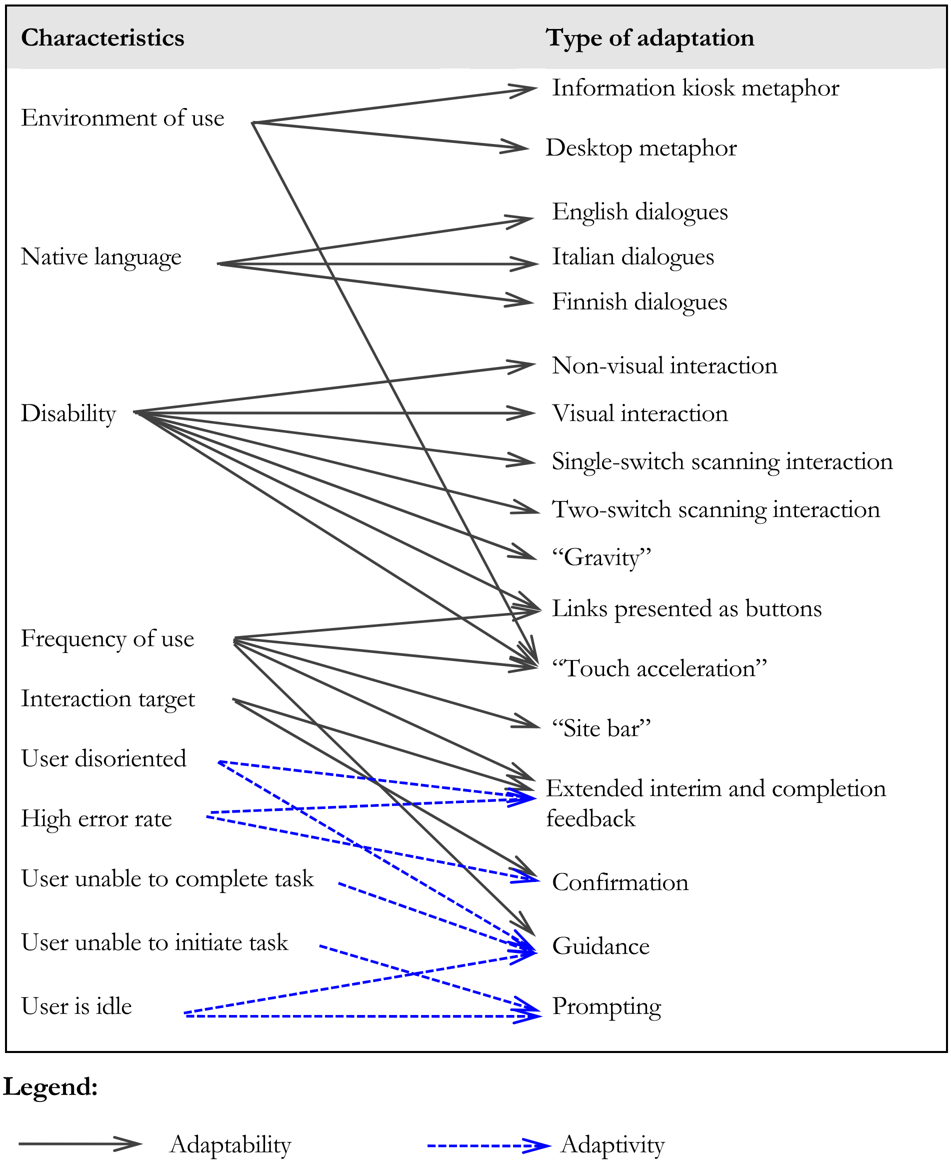
The adaptation mechanism of the AVANTI user interface component comprises sub-components which collectively allow for rule-based adaptation decisions to be made. It is based on a two-fold approach, which is briefly discussed below:

1. Implementation of the user interface must be carried out in a task-, and style-aware manner, i.e. the design knowledge and alternatives of

the task decomposition and dialogue design must be clearly represented in the actual interface.

2. There must exist a decision mechanism, which will undertake the task of maintaining, evaluating and administering adaptation rules. The decision mechanism should: firstly, provide ways in which it can be *consulted* for the provision of decisions for the syntactic and lexical levels of adaptations; secondly, be capable of *propagating* adaptation decisions (thus *triggering adaptations*) at either level of the interaction.

Table 3: Interface-level adaptations in AVANTI.



The UI decision mechanism adheres to the above description and comprises the following sub-components (Figure 18 and Figure 19): (i) the *syntactic adaptability rule base*, which retains the task- and style-related rules, referring to "static" user characteristics and preferences; (ii) the *syntactic adaptivity rule base*, which retains the task- and style-related rules, referring to dynamic user characteristics and situations; (iii) the *lexical adaptability rule base*, which retains the lexical element-related rules, referring to "static" user characteristics and preferences; (iv) the *lexical adaptivity rule base*, which retains the lexical element-related rules, referring to dynamic user characteristics and situations; and, (v) the *knowledge space*, which maintains knowledge on "static" and dynamic user characteristics and preferences.

B.1.4.1 Adaptability

Adaptability is based on user characteristics and preferences that are known prior to interaction and are, in any case, assumed to remain static throughout a single interaction session. As a consequence, the corresponding rules can be evaluated during the initiation of the system and the resulting decisions can be directly applied for the instantiation of the interaction dialogues. The procedure followed is depicted in Figure 18:

- A task x is triggered, either automatically (e.g., during system start-up), or as a response to a user action. The embedded communication facilities of the task structure consult the decision mechanism for the appropriate style(s) to be instantiated. The parameter passed is the identification of the task itself. (Figure 18: (1))
- The syntactic adaptability rule base consults the knowledge space for the "current" user characteristics and preferences and evaluates its rules. The result returned is a (list of) style(s) that should be instantiated. (Figure 18: (2))
- The task structure invokes the styles specified in the previous step, passing any required application-specific parameters. (Figure 18: (3))
- Any instantiated style creates / modifies specific "portions" of the user interface, comprising individual interactive components that are at some point created for presentation to the user. The communication facilities embedded to the *proxy adaptation object* attached to each such component, consult the decision mechanism for the appropriate attributes to be implemented (e.g., size, color, volume). The parameters passed to the decision mechanism in this case are the task and style to which the component belongs, as well as the class / category of the component. (Figure 18: (4))
- The lexical adaptability rule base consults the knowledge space for the "current" user characteristics and preferences, and evaluates its rules.

The result returned is a list of attribute-value pairs that represent specific attributes of the component class and the respective values for the object that initiated the consultation. (Figure 18: (5))

- The interface component applies the attributes to itself and proceeds to complete the steps required for its initialization and presentation to the user. (Figure 18: (6))

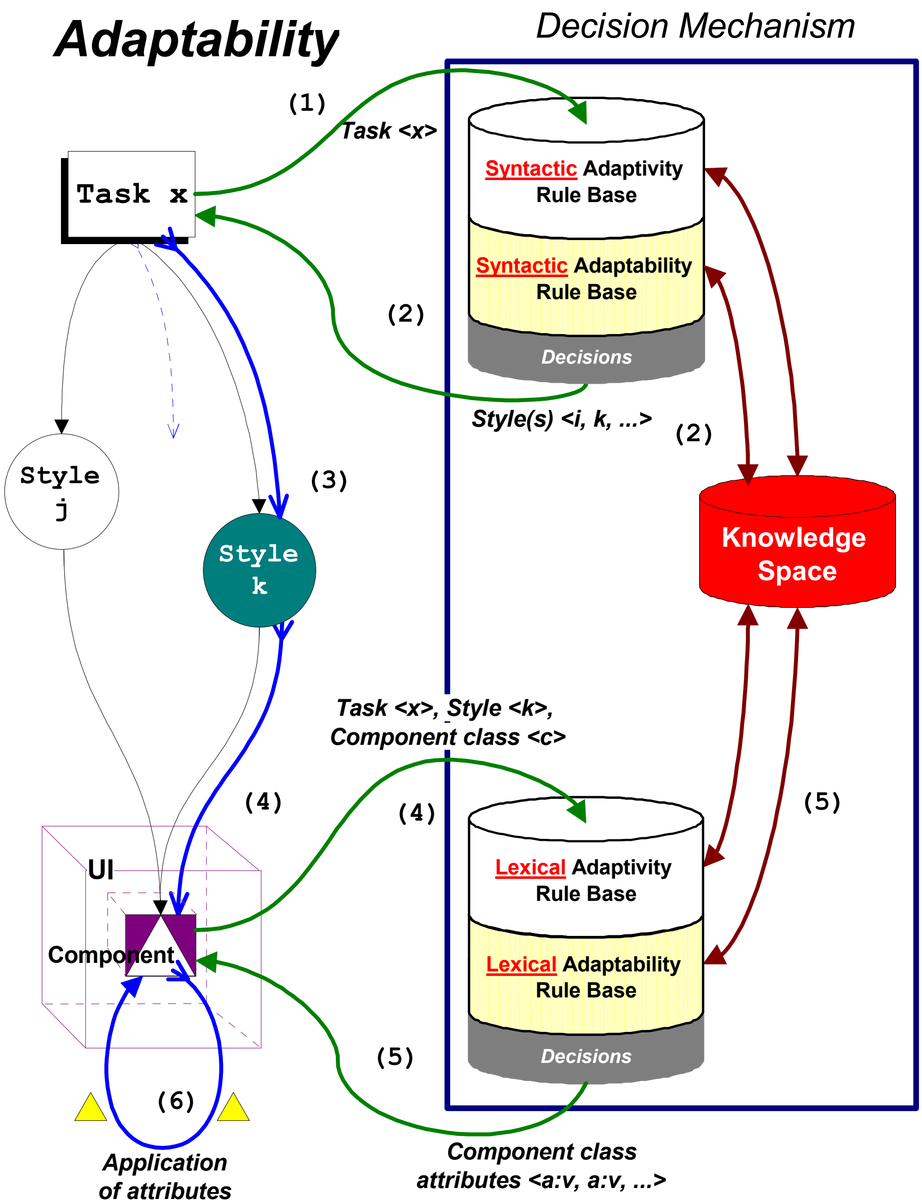


Figure 18: Adaptability Mechanism in the AVANTI UI

A main characteristic of the way in which adaptability is achieved (as opposed to adaptivity), is that communication between the decision mechanism and the user interface is initiated by the user interface constituents.

B.1.4.2 Adaptivity

Adaptivity is applicable at run-time and cannot be initiated by the interface constituents, as they do not have knowledge of *changing* user characteristics and situations. Thus, it is necessary that the decision mechanism triggers the adaptations itself. The procedure followed in the case of adaptivity, is depicted in Figure 19:

- The UMS *utilizes* monitoring data sent continuously by the user interface, and makes inferences on dynamic user characteristic(s) or situation(s) and informs accordingly the user interface decision mechanism (more specifically, it communicates new situations to the user interface knowledge space through a standard communication module. (Figure 19: (1))
- The knowledge space triggers the re-evaluation of rules in the syntactic and lexical adaptivity rule bases. (Figure 19: (2))
- Once the evaluation mechanism of the syntactic adaptivity rule base is triggered by the knowledge space, all rules that (partially, or entirely) depend on the modified knowledge are evaluated. This may result in new decisions regarding the styles that should be used to instantiate specific tasks, and notification is sent to the affected task structures accordingly. (Figure 19: (3))
- When a task structure receives notification from the decision mechanism that a different set of styles should be used for its instantiation, it performs two distinct steps: (i) it stores this piece of information for use in future invocations, and (ii) it checks whether it is currently active (i.e. if the corresponding task is being carried out); if so, it may be necessary to dynamically deactivate certain styles and possibly also activate alternative ones in their place. (Figure 19: (4))
- In parallel, the evaluation mechanism of the lexical adaptivity rule base is triggered by the knowledge space, and all rules that (partially, or entirely) depend on the modified knowledge are evaluated. This may result in new decisions regarding the values of the attributes that certain interface objects (participating in specific tasks and styles) should have, and notification is sent to the affected objects accordingly. (Figure 19: (5))
- When an affected object receives notification from the decision mechanism that a different set of attributes should be exhibited, it

applies the new attributes to itself, possibly after retracting any other conflicting attributes set in the past. (Figure 19: (6))

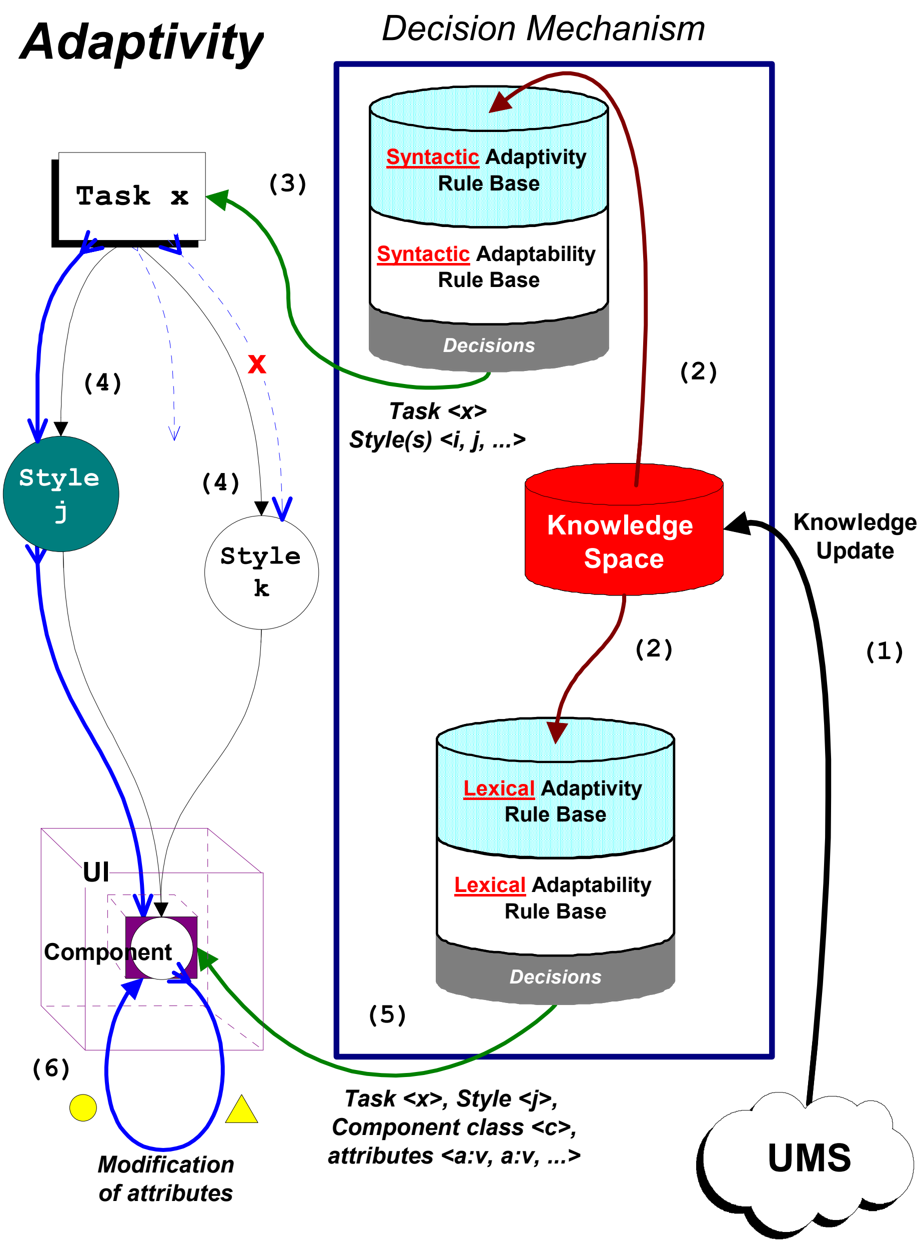


Figure 19: Adaptivity mechanism in the AVANTI UI

Central to the overall adaptivity mechanism is the communication with the UMS, which actually triggers the modifications in the user interface, by dynamically providing inferences drawn from knowledge provided through monitoring, as well as through user group stereotypes and static user-specific characteristics (see [Fink, Kobsa & Nill, 1999]).

B.1.5 Instances of Adaptations in the AVANTI User Interface

To illustrate some of the concepts of adaptation discussed above, we provide some instances of adaptability and adaptivity of the AVANTI user interface in Figure 20 - Figure 24 (from [Stephanidis, Paramythis et al., 2001]). In particular, the figures illustrate: (a) a "typical" instance of the interface, resembling generic browsers (Figure 20); (b) a mixture of adaptations at the syntactic level (enabled "site-bar", explicit feedback for the "add bookmark" operation) and lexical level of interaction (links presented as buttons, instead of as underlined text) (Figure 21); (c) a guidance dialog presented to the user due to the detection of the user's inability to complete a specific task (Figure 22); and, (d) a second case of task guidance provision, within the dialog encapsulating the user task to which guidance refers (Figure 24 and Figure 24).

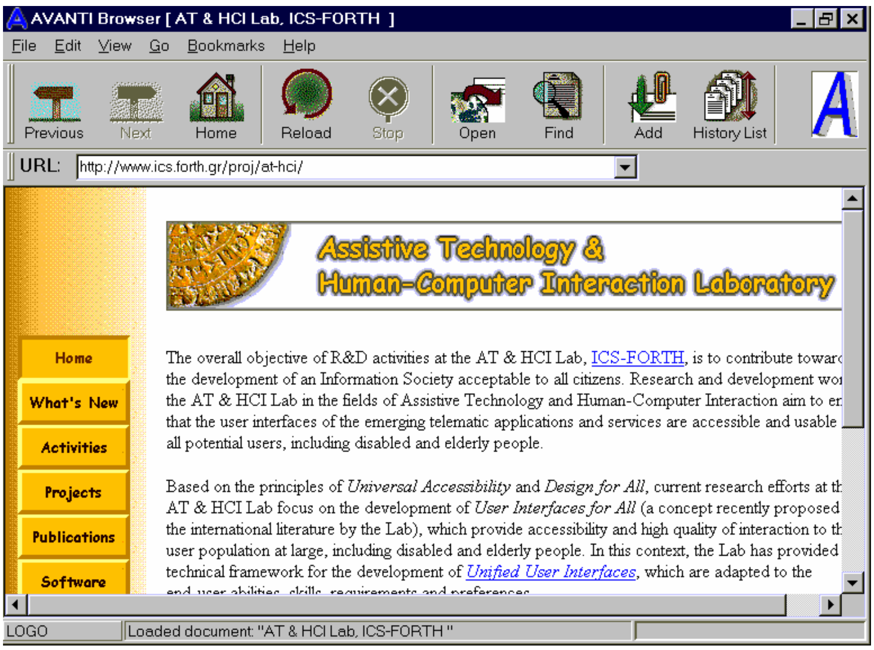


Figure 20: Conventional instance of the AVANTI browser interface.

As Figure 23 and Figure 24 illustrate, AVANTI has also explored different approaches to applying adaptations and introducing them to the user, as well as different approaches to ensuring that the user retains final control over what gets modified into the system and how.

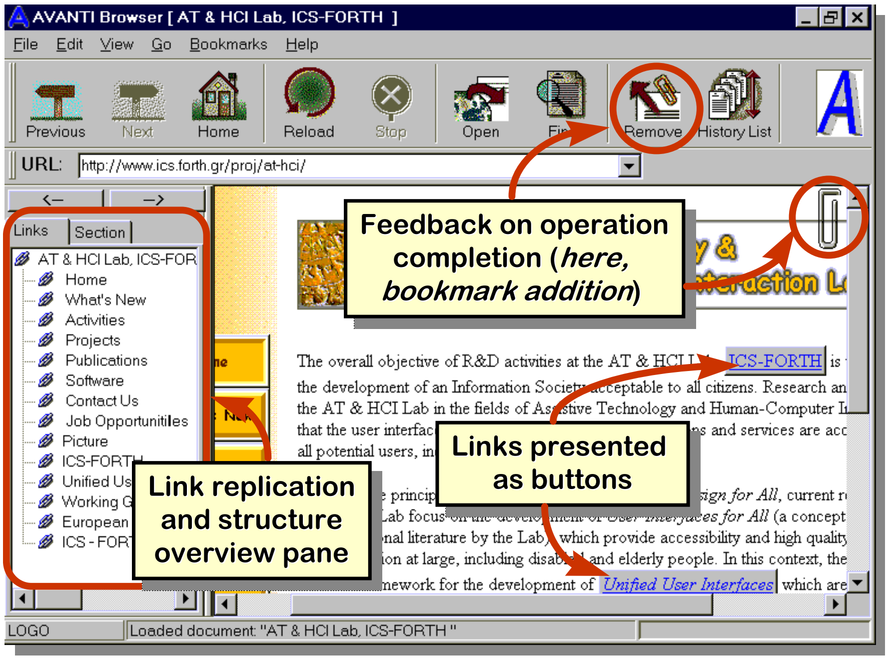


Figure 21: A mixture of adaptability features activated.

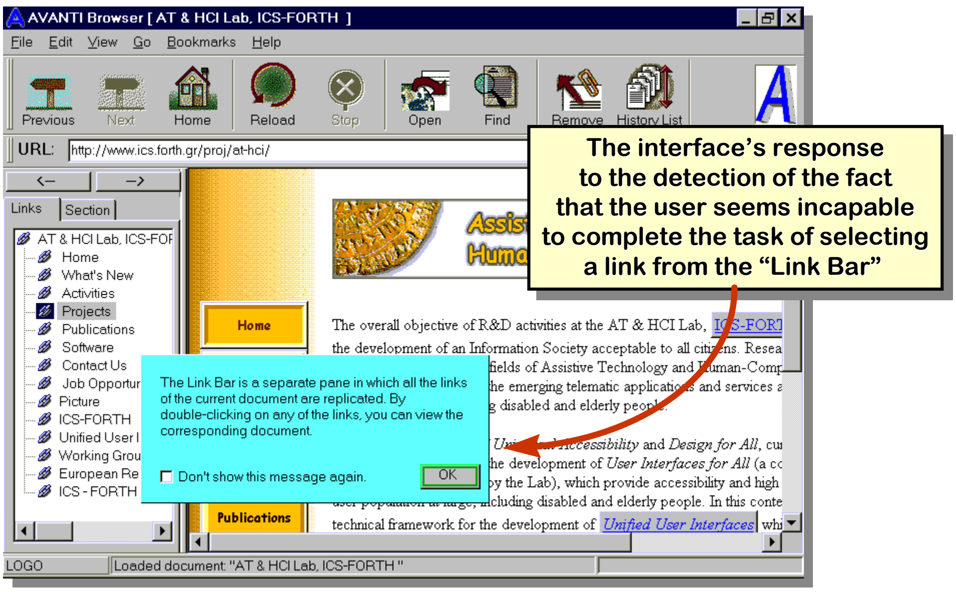
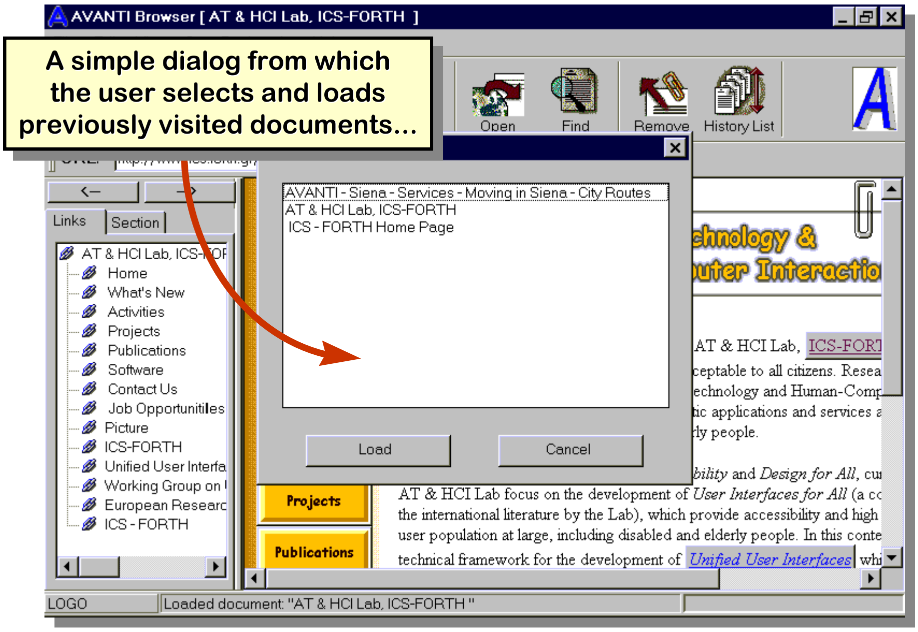
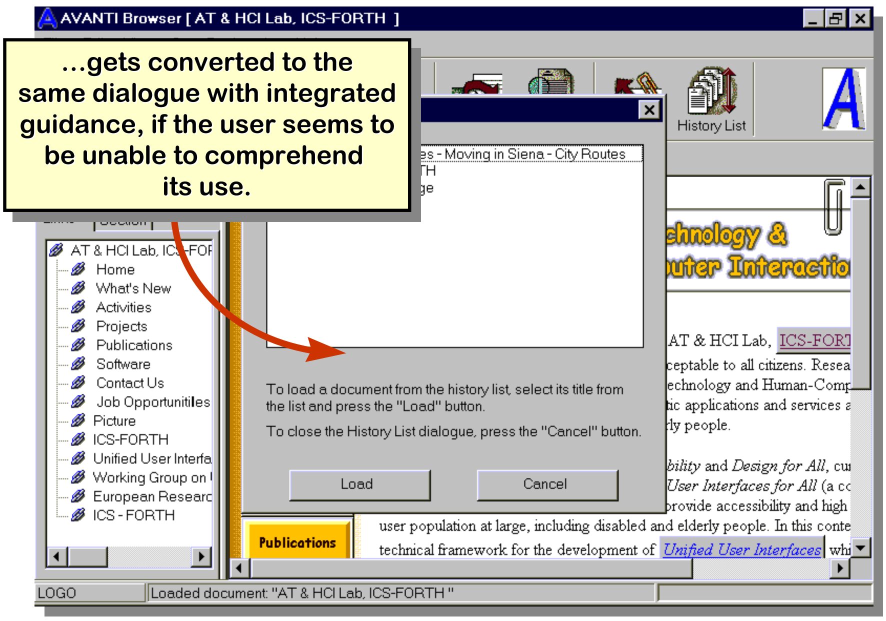


Figure 22: Instance of external task guidance.



Instance of embedded task guidance (*before*).

Figure 23: First instance of embedded task guidance.



Instance of embedded task guidance (*after*).

Figure 24: Second instance of embedded task guidance.

B.1.6 Discussion

This chapter has presented the user interface component of the AVANTI information system. The design and development have followed the Unified User Interface Development methodology, rendering the resulting unified interface capable of adapting itself to suit the requirements of three user categories: able-bodied, blind and motor impaired. Adaptability and adaptivity are used extensively to tailor and enhance the interface respectively, in order to effectively and efficiently meet the target of interface individualization for end users. To support interaction by disabled users, special I/O devices and respective interaction techniques have been integrated into the system.

In addition to the above, the AVANTI user interface offers a number of features that are aimed at assisting and enhancing user interaction with the system, as well as improving the accessibility of the resulting interface by specific user categories (see for example [Gunderson, 1998], [Vanderheiden & Chisholm, 1998], [Stephanidis et al., 1998a]). Such features include: (i) enhanced history control for blind users, as well as linear and non-linear (graph) history visualization for sighted users; (ii) resident pages that enable users to review different pieces of information in parallel; (iii) link review and selection acceleration facilities; (iv) document review and navigation acceleration facilities; (v) enhanced mechanisms for document annotation and classification; and, (vi) enhanced intra-document searching facilities. The design and development of these features has been based on techniques used to support user navigation and orientation in large hypermedia systems (see, e.g., [Nielsen, 1995]) and are not available in commercial browsers, as well as on existing empirical studies of user interaction patterns on the web (e.g., [Catledge & Pitkow, 1995], [Tauscher & Greenberg, 1997]).

The AVANTI UI underwent summative evaluation at three trial sites, using three different underlying information systems. In total, more than 200 able-bodied and disabled end users participated in these summative evaluation activities, some of which targeted adaptivity[11]. Further to this project-scope evaluation work, more formative evaluation activities also took place, focusing explicitly on the evaluation of the system's adaptive features; these are discussed in some detail in chapter "C.1 Evaluating Adaptable and Adaptive User Interfaces: Lessons Learned from the Development of the AVANTI Web Browser", page 107.

B.1.7 Reported Work against the State-of-the-Art

To date, AVANTI remains the only case in the literature of both *adaptability and adaptivity* employed at the level of the user interface of a desktop

[11] Additional information about the AVANTI project's user trials can be found on the project's web site, and specifically at the address: http://www.ifac.cnr.it/avanti/contents/contents/user.htm

application to improve both *accessibility and usability* for its end users. There exist, however, approaches and systems clearly related to aspects of the work described that merit attention.

Starting with accessibility in hypermedia browsing applications, worth mentioning is DAHNI [Petrie et al., 1997] a hypermedia system with a non-visual interface, in development cotemporally with AVANTI. This system supported a large variety of input and output devices but offered quite limited interaction options, and exhibited no adaptivity capabilities at all. WebAdapter [Hermsdorf, Gappa & Pieper, 1998], also developed at approximately the same time as AVANTI, was based on the principles of User Interfaces for All, and employed adaptability to reactively support different categories of end users, focusing on users with disabilities. A more recent effort reported in [Tan, Yu & McAllister, 2006] utilizes adaptation to provide access to graphics embedded in web pages for blind users. The approach presented is particularly noteworthy in that it is based on an external, componentized architecture, which works in tandem with normal mainstream browsers (making use of Microsoft Active Accessibility, a technology that will be discussed later in this section). However, the solution is highly specialized (it is specifically tailored to the presentation of graphics for blind users), and appears to be limited to adaptability techniques only (the authors claim that their solution is adaptive, but provide no information on whether and how the user model is populated by inferences made through user monitoring; the system does have a user preferences' profile, but users have to maintain that themselves).

Other specialized browsers in the literature aimed at supporting users with specific types of disabilities (most often visual impairments), but without adaptation capabilities, included the pwWebSpeak browser [De Witt & Hakkinen, 1998], and IBM's Home Page Reader [Chieko & Lewis, 1998]. Considerable efforts have also been reported in the development of custom interaction techniques intended to be used in conjunction with mainstream browsers. Spalteholz, Li and Livingston [2007], for instance, describe a specialized text input technique for users only capable of operating single switches, which unlike the one used in AVANTI, is non-linear and may potentially result in faster selection of links (the task for which this technique has been optimized).

Not strictly in the domain of hypermedia browsing applications, but closely related to AVANTI in terms of the UUI approach and the simultaneous support for different categories of end users is the Starlight Platform [Grammenos et al., 2007], an e-book reader with a dual (visual and non-visual) user interface.

Moving to the domain of usability-oriented user interface adaptation in browsing applications, Henricksen and Indulska [2001] present an adaptive browser, which, with the collaboration of a customized web server, can adapt itself to characteristics of the network communication (e.g., throughput), and the availability of input / output devices.

The preceding effort is indicative of a lot of work in the area of user interface adaptation in the last decade, which considered the interaction context as the main (although not necessarily the only) driver of interface adaptation. An influential approach in this direction, introduced after the end of the work in AVANTI, is presented in [Thevenin, Coutaz & Calvary, 2004], which proposes a reference conceptual framework that helps structure the development process of "plastic" user interfaces. According to the authors, the term *plasticity* is inspired from materials that expand and contract under natural constraints without breaking, thus preserving continuous usage. Applied to HCI, plasticity is the "capacity of an interactive system to withstand variations of contexts of use while preserving usability" [Thevenin & Coutaz, 1999].

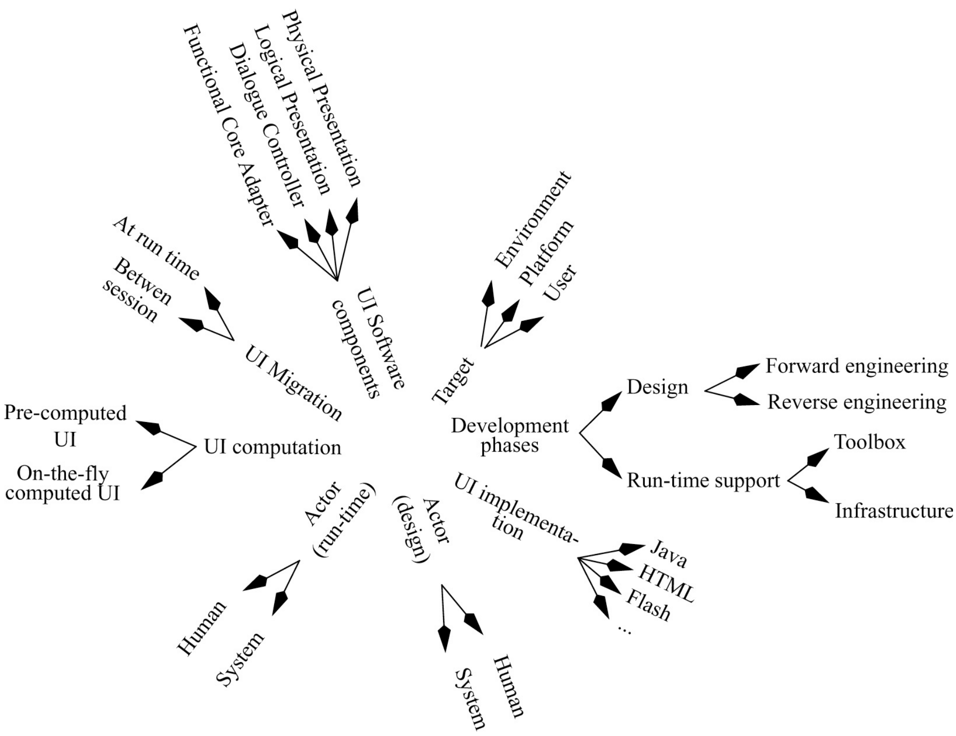


Figure 25: The *Plastic UI Snowflake*: a problem space for characterizing software tools, and for expressing requirements for software tools aimed at plastic user interfaces [Thevenin, Coutaz & Calvary, 2004].

The aforementioned framework aims to address an expanded problem space as compared to AVANTI (see Figure 25), addressing three dimensions of adaptation determinants: users, platforms and environments. The framework was also accompanied by ARTStudio [Thevenin, 2002], a tool supporting the development of plastic user interfaces, which, however, only supported so-called "multi-platform targeting" (i.e., adaptation to different classes of interaction platforms), and did not explicitly include support for targeting development to multiple users and environments.

The AVANTI UI / browser represented a major development effort, not only in terms of the adaptation capabilities described in the preceding sections, but also in implementing HTTP communication and HTML rendering, designing and developing the different styles for accessibility and usability, implementing the augmented navigation facilities, etc. In the rest of this section we will consider how the reported work relates to modern web-related accessibility and personalization approaches, and what the amount of effort required to meet the same set of goals today is.

At the time the AVANTI project was started, there were only three web browsers available for the Microsoft Windows operating system (out of the total four that were in active development at the time)[12]: NCSA Mosaic, Netscape Navigator (NN), and Microsoft Internet Explorer (MSIE). Of these, NCSA Mosaic was the only one for which source code was obtainable (the X Window System / Unix version publicly provided source code; source code for the other versions was available after agreements were signed)[13], although development had all but ceased in favor of Netscape Navigator, and the obtainable code was lagging behind the web standards of the time (notably missing features required by the HTML 3.2 specification, such as nested tables). As a result, the NCSA Mosaic code base did not offer itself as a promising choice for basing a new web browser on. The MSIE WebBrowser control[14], which enables the programmatic embedding of MSIE's HTTP modules and rendering engine in other applications, was not available yet (it was introduced with version 4 of MSIE, released in 1997).

At the same time, although there was some support for plug-ins in NN, this was rudimentary and was intended for the development of add-on components capable of handling MIME types other than HTML (e.g., PDF documents); there was practically no support for effecting modifications to the browser's user interface, or in the rendering of HTML pages. A direct implication of this situation was that it was not possible to interact with the Document Object Model (DOM) of an HTML page unless doing so from within the rendering engine code of a browser.

Another technology that was missing at the time development in the AVANTI project begun was the Microsoft Active Accessibility (MSAA) Application Programming Interface (API) [15]. The MSAA API, which was

---

12 See http://en.wikipedia.org/wiki/History_of_the_web_browser

13 See http://en.wikipedia.org/wiki/Mosaic_(web_browser)

14 See http://msdn.microsoft.com/en-us/library/aa752040(VS.85).aspx

15 See http://msdn.microsoft.com/en-us/library/dd373592(VS.85).aspx

introduced as an add-on to the Microsoft Windows 95 operating system in 1997, is designed to help Assistive Technology (AT) products (e.g., screen readers, on-screen keyboards, etc.) interact with standard and custom user interface elements of an application (or the operating system), as well as to access, identify, and manipulate an application's interface elements. The absence of this technology at the time of development meant that, even if an embedded rendering engine were available, we would still not be able to inspect or modify the contents of the rendering pane in the ways that were necessary for the project's goals.

As stated already, things have changed dramatically in the intervening years. Starting from the last technology discussed, the MSAA API has enabled the development of sophisticated AT products that are specialized in assisting people with specific types of disabilities (e.g., the JAWS screen reader[16]), which are often specifically tailored to facilitate the use of popular browsers. In fact, a lot of accessibility solutions (such as a workable on-screen keyboard) are now embedded into the Microsoft Windows operating system itself, in many cases obviating the need for external solutions. This is evidenced by the declining number of custom accessible browsers available as stand-alone applications. Older products, such the pwWebSpeak browser [De Witt & Hakkinen, 1998], and IBM's Home Page Reader [Chieko & Lewis, 1998] are not available any more, whereas products still under development, such as the WebbIE[17] [King, Evans & Blenkhorn, 2004] appear to be very specialized in nature (e.g., WebbIE is highly tailored for accessing the UK BBC's web site).

But, whereas specialized browsers seem to be eclipsing, the opposite is true for browser plug-ins intended for enhancing accessibility. This has been made possible by new plug-in architectures and APIs, which allow for much greater control over the presentation of, and interaction with, web content, and specifically with a rendered page's DOM. A typical example is the Firefox Accessibility Extension developed the University of Illinois Center for Information Technology and Web Accessibility[18], and there exist many more[19].

Further to the above, accessibility of web content in general has been facilitated by the emergence of sets of web-related accessibility guidelines, such as those issued by the World Wide Web Consortium's Web Accessibility Initiative[20], which cover web content, rich internet applications, user agents (including browsers), and authoring tools, as well as of software to enable the automatic assessment of levels of compliance with the said guidelines.

---

16 See http://www.freedomscientific.com/products/fs/jaws-product-page.asp

17 See http://www.webbie.org.uk/

18 See http://firefox.cita.uiuc.edu/

19 See, for instance, http://www.accessfirefox.org/ for a list of accessibility extensions for the Firefox browser.

20 See http://www.w3.org/WAI/guid-tech.html

Moving beyond accessibility, one can observe that the powerful and now standardized technologies that are available in all modern browsers (such as Javascript, DOM access, rendering based on Cascading Style Sheets, etc.), coupled with the aforementioned extended capabilities of browser plug-in architectures and APIs, have had a significant impact on how adaptivity and personalization of web content is approached. A characteristic example of this trend is the Firefox plug-in described in [Eynard, 2008], which uses semantic data and a user model created from explicit and implicit modeling of users' browsing activities to augment the contents of visited pages with links to related pages, etc.

Against this background, were the AVANTI UI / browser to be developed today, it is a relatively safe assumption that it would be implemented as an extensive browser plug-in, or at least as a modified version of an open-source web browser such as Firefox. In fact, to enable the full range of adaptations that AVANTI was capable of, such as augmentation of interactive dialogs, one would have to go with the later option of custom development "on top" of an existing browser.

B.2 A Generic Adaptation Framework for Web-based Hypermedia Systems[21]

The advent of the web as a global communication infrastructure, enabling concurrent access to heterogeneous and distributed information sources through a wide variety of media and access devices, has brought about fundamental changes in the way computer-mediated human activities are conceived, designed, developed and experienced, giving rise to the progressive emergence of the Information Society.

This dynamic evolution is characterized by several dimensions of diversity that become evident when considering the broad range of user characteristics, the changing nature of human activities, the variety of contexts of use, the increasing availability and diversification of information, knowledge sources and services, the proliferation of diverse technological platforms, etc. In this context, the notion of Universal Access [Stephanidis, 2001a] has become critically important for ensuring social acceptability of the emerging Information Society. Universal Access implies the accessibility and usability of Information Society Technologies (IST) by anyone, anywhere, anytime. Its aim is to enable equitable access and active participation of potentially all citizens in existing and emerging computer-mediated human activities.

The concept of automatic adaptation has been investigated under the perspective of proactively supporting Universal Access by providing built-in accessibility and high interaction quality in applications and services in the emerging Information Society [Stephanidis, 2001b]. In the context of the web, Universal Access concerns both the interactive behavior and the content of applications and services, and requires global approaches to adaptation.

Adaptable and adaptive software systems have been considered in a wide range of research efforts. The relevant literature offers numerous examples illustrating tools for constructing adaptive interaction (e.g., [Brusilovsky, Kobsa & Vassileva, 1998; Horvitz, 1999; Kobsa & Pohl, 1995; Sukaviriya & Foley, 1993]), and case studies in which adaptive interface technology has improved, or has the potential to improve, the usability of an interactive system (e.g., [Dieterich et al., 1993; Benyon, 1993; Benyon, 1997]).

In particular, Adaptive Hypermedia Systems (AHS) are a relatively new area, which, however, has drawn considerable attention since the advent of the web (which can be practically considered as a "universal", widely deployed hypermedia system). Major categories of adaptive hypermedia systems include

21 This chapter is based on [Paramythis & Stephanidis, 2005] and [Paramythis et al., 2003b]. The R&D work described here has been carried out while the author was employed by the HCI-Lab of ICS-FORTH (Heraklion, Greece), and in the context of the IST-1999-20656 PALIO project (please refer to section "A.2.2 Work Context and Research Projects" on page 32 for additional information).

educational hypermedia, on-line information systems, on-line help systems, information retrieval systems, and institutional hypermedia. There exist today numerous adaptive systems, in various applications domains, with a great variety of capabilities (see, e.g., [Ardissono & Goy, 1999; Balabanovic & Shoham, 1997; Brusilovsky, Kobsa & Vassileva, 1998; Henze, 2001; Oppermann & Specht, 1998; Kobsa, 2001]).

This chapter presents a generic hypermedia framework for the development of adaptive web services, based on a perspective on adaptation which is claimed to go beyond previous efforts and characterizes software products that automatically modify (adapt) their interactive behavior according to the individual attributes of users (e.g., mental / motor / sensory characteristics, preferences), and to the particular context of use (e.g., hardware and software platform, environment of use), as well as the content of applications and services. In such a framework, adaptation implies the capability, on the part of the system, of capturing and representing knowledge concerning alternative instantiations suitable for different users, contexts, purposes, etc., as well as for reasoning about those alternatives to arrive at adaptation decisions. Furthermore, adaptation implies the capability of assembling, coherently presenting, and managing at run-time, the appropriate alternatives for the current user, purpose and context of use.

In the context of this chapter, the term "framework" is used to refer to an architectural design describing the components of the system and the way they interact [Campbell et al., 1991]. The confines of an architectural framework for software systems are perhaps best described as per [Jacobson, Griss & Johnson, 1997]:

> "The software architecture, first of all, defines a structure. Software components have to fit into some kind of design. […] Second, the architecture defines the interfaces between components. It defines the patterns by which information is passed back and forth through these interfaces."

The presented framework comprises both implemented components and specifications (in the form of programmatic interfaces and associated semantic "contracts") of how core- and external- components interact to attain adaptive system behavior. The framework has been implemented in Java, and comprises concrete classes, which implement the functionality of the core components, as well as abstract classes and interfaces, which are used when integrating external components with the framework.

The main characteristics of the framework presented here are: support for declarative (vs. programmatic) specification of adaptive system behavior; composition of adaptive hypermedia techniques from lower-level adaptation actions; inherent support for different approaches to representing and evaluating user- and context- models, as well as adaptation logic itself; domain-independence, coupled with provisions for capturing the semantics and specificities of individual application domains; and, finally, orthogonal applicability to any document-centric hypermedia system with XML-compliant output. As we will see towards the end of the chapter, there are similarities but also important differences in how adaptation is approached in this framework as compared to the employed in the AVANTI UI.

The following topics are addressed in this chapter: hypermedia adaptation techniques, as these are defined in the literature; adaptation actions supported by the framework; synthesizing higher-level adaptation techniques through action composition; coupling actions with different forms of adaptation "logic" (i.e., abstracting over the decision-making functionality); coupling actions with different types of dynamic models (i.e., abstracting over user-, context-, etc., model representations); coupling actions with different domain models (i.e., abstracting over system aspects specific to the application domain at hand, using the concept of domain ontologies); "baseline" implementations of the modeling and decision-making components, in support of declarative specification.

The framework under discussion was employed in the development of the PALIO tourist information system (see "A.2.2 Work Context and Research Projects" on page 32, for additional information on the PALIO project) addressed the issue of Universal Access to community-wide services, based on content and user interface adaptation beyond desktop access. The framework being presented here was used to enable adaptive system behavior at the interaction and content levels, on the basis user- and context-characteristics (including terminal device capabilities, user location, etc.) The evaluation of the resulting information system by end users provided very positive feedback with respect to the system's adaptive features. The PALIO system and the results of its evaluation will be briefly presented in the chapter.

The rest of this chapter is structured as follows. The next section, "B.2.1 Background and Related Work", introduces the framework itself. The presentation commences with a brief account of the main premises of the framework ("B.2.2.1 Overview"). Following that, *adaptation actions*, one of the cornerstone concepts of the framework, is discussed in detail ("B.2.2.3 Adaptation Actions"). The discussion covers both the types of adaptation actions that the framework currently supports, and, at a more theoretical level, the relationship between adaptation actions and adaptive hypermedia techniques in the literature. Section "B.2.2.4 Adaptation Decisions", addresses another major aspect of adaptive systems, namely, deciding upon the need for, and the type of, adaptation. Subsections discuss the framework's support for alternative (to the default) approaches to decision-making, the default rule-based implementation, and the way in which the framework abstracts over different dynamic and static system models in the context of decision making. Section "B.2.3 Applying the Framework in PALIO", provides an overview of our experiences in using the adaptation framework to enable adaptive behaviors in the PALIO tourist information systems, along with the valuable insight gained along the way. Section "B.2.4 Discussion" discusses our findings from the employment of the framework in PALIO. The chapter's last two sections address, respectively, the differences between the framework described here and: (a) the one used in AVANTI, (b) other AH frameworks in the recent literature.

B.2.1 Background and Related Work

The main goals in the development of the presented adaptation framework were to:

- Support a wide range of adaptive hypermedia techniques, in a domain-independent way.
- Provide constructs that facilitate the declarative specification of adaptive behavior.
- Achieve, at the architectural level, "orthogonality" with existing web-publishing approaches, so that the framework can be easily integrated into existing non-adaptive systems and services.
- Enable the clear separation of adaptation components, so that their implementation can be varied independently.

These goals are discussed briefly below and contrasted against related work in the field.

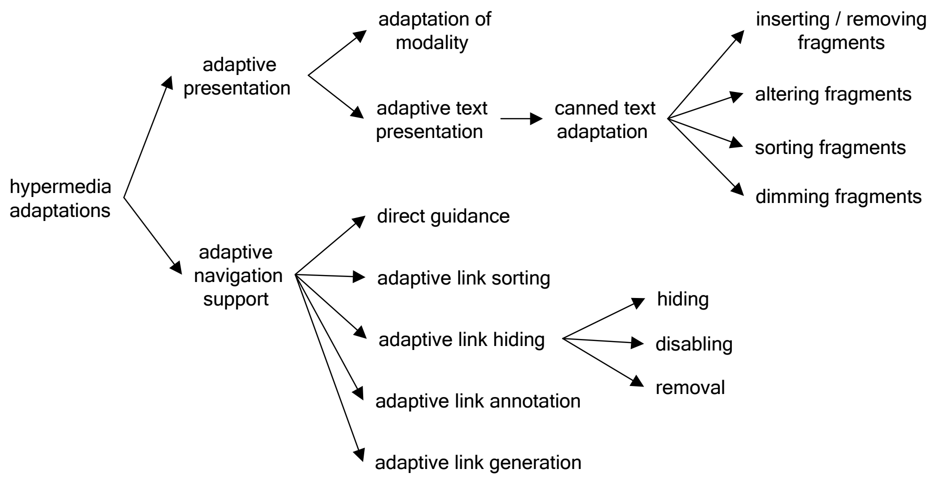


Figure 26: Classification of adaptation techniques supported by the framework, as per [Brusilovsky, 1996; Brusilovsky, 2001] (from [Paramythis et al., 2003b])

To start with, the term "adaptive hypermedia techniques" [Brusilovsky, 1996] is used to refer to modifications which can be adaptively applied to hypermedia documents and which can be synthesized to arrive at higher-level adaptation *method*, such as "additional explanations" and "global guidance". Figure 26 presents a partial classification of "server-side" adaptation techniques. The figure has been based on the classification introduced in [Brusilovsky, 1996] and refined in [Brusilovsky, 2001] and contains the techniques that should be directly supported by the framework. Techniques not shown include *adaptive multimedia adaptation* (a sub-category of *adaptive presentation*), *natural language adaptation* (a sub-category of *adaptive text presentation*), and *map adaptation* (a sub-category of *adaptive navigation support*).

There exist today several web-based AHS, which address a number of these techniques in a more or less domain-independent manner. A representative example in this category is the AHA! system [De Bra & Stash, 2002; De Bra et al., 2002b]. In its second generation, AHA! has become a rather comprehensive system, including also integrated authoring tools. Although this greatly facilitates the creation of new adaptive systems, there is at least one significant problem with the approach taken: the AHS is rather "monolithic" and cannot be easily integrated with pre-existing, non-adaptive systems; rather the AHS *is* the web-publishing system. This, along with a specific approach to achieving adaptations (based on domain concepts and their relationships [De Bra et al., 2002a]) seriously impedes the applicability of the system outside its original domain of adaptive course provision.

A different approach is represented by the KnowledgeTree framework [Brusilovsky & Nijhavan, 2002]. Whereas the majority of AHS are designed to exist as stand alone systems, KnowledgeTree has been designed to source adaptive content and functionality externally, not encapsulating them into a monolithic core. KnowledgeTree is specifically intended to facilitate interoperation and reuse at the level of distributed, reusable learning activities (with the emphasis being on learning activities, as opposed to learning objects). To this extent, KnowledgeTree goes into the realm of run-time communication and interoperation standards, seeking to standardize the ways in which different specialized subsystems supporting aspects of the (adaptive) learning process can communicate and exchange information that would allow them to be aggregated into a "whole". Although KnowledgeTree is explicitly targeted towards adaptive learning environments, its main concepts can easily be generalized across application domains. However, KnowledgeTree imposes a specific portal-oriented structure to the web-publishing system, which may inappropriate in certain scenarios.

The concept of declarative specification of adaptation logic is not new. Several AHS (including, for example, AHA!) make it possible to specify the adaptive behavior of the system through relations between domain concepts and actions to be taken when specific conditions are met. However, the decomposition of adaptations applied into more basic building blocks that can be reused individually has only been recently addressed. The body of work that is perhaps closest to the framework presented herein, in this respect, is the "LAG" model [Cristea & Calvi, 2003]. This is a three-layer model and classification method for adaptive techniques comprising the following levels: direct adaptation rules, adaptation language and adaptation strategies. The model is aimed at standardizing adaptation techniques at the different levels and, thus, enable the exchange of adaptive techniques between different applications. It also aims to help the authors of adaptive hypermedia by giving them higher-level "handlers" of low-level adaptation techniques [Cristea & Calvi, 2003]. Although the objectives of the two approaches are quite similar, there exist fundamental differences in the way they are achieved. In LAG, for instance, the lowest layer addresses adaptation techniques as functions that map the current state of the AHS and its models to a subsequent (adapted) state. The middle layer comprises the adaptation rules, and the third layer addressed adaptation "strategies", as these relate to the user's information processing characteristics and cognitive styles. To contrast this with the approach taken in the presented framework, please refer mainly to sections "B.2.2.3 Adaptation Actions" and "B.2.2.4 Adaptation Decisions".

B.2.2 Generic Adaptation Framework

B.2.2.1 Overview

One of the main premises of the developed framework is that it follows a *document-centric* approach. Specifically, the framework explicitly embeds notions for the request-response cycles typical of web-based systems, and assumes that the result of each cycle is one or more documents (or document fragments) that constitute the "response" to the user's "request". These documents / fragments are the framework's adaptation constituents.

The framework is geared towards *XML-based document representation*. This necessitates that either the documents are "expressed" in an XML-based language, or that they can be easily converted into such a representation. It should be clarified that the framework does not assume that this is the final step in the document processing cycle; it only requires that, at some stage of the aforementioned cycle, documents be represented in XML, so that adaptations can be applied to them.

The process of adapting documents requires the cooperation of at least two different types of components, namely the *decision-making component*, and the *adaptation engine*. The former is responsible deciding upon adaptations to be performed. The latter is responsible for applying adaptation decisions, expressed through adaptation actions. Adaptation decisions, in turn, typically require access to the adaptation models (e.g., user model, context model, domain / application model), which are encapsulated by the *modeling components*. Communication with the modeling components is also necessary in the case of models that are updated dynamically. In this case, the communication concerns the exchange of interaction data that will be used as "evidence" towards the dynamic updates.

Communication between the components that make up the framework is done through a set of well-defined programmatic interfaces, intended to enable the decoupling of the components and facilitate their replacement with alternative implementations. This is also the case for accessing and manipulating information available from sources "external" to the framework, such as user modeling servers and domain-specific information servers.

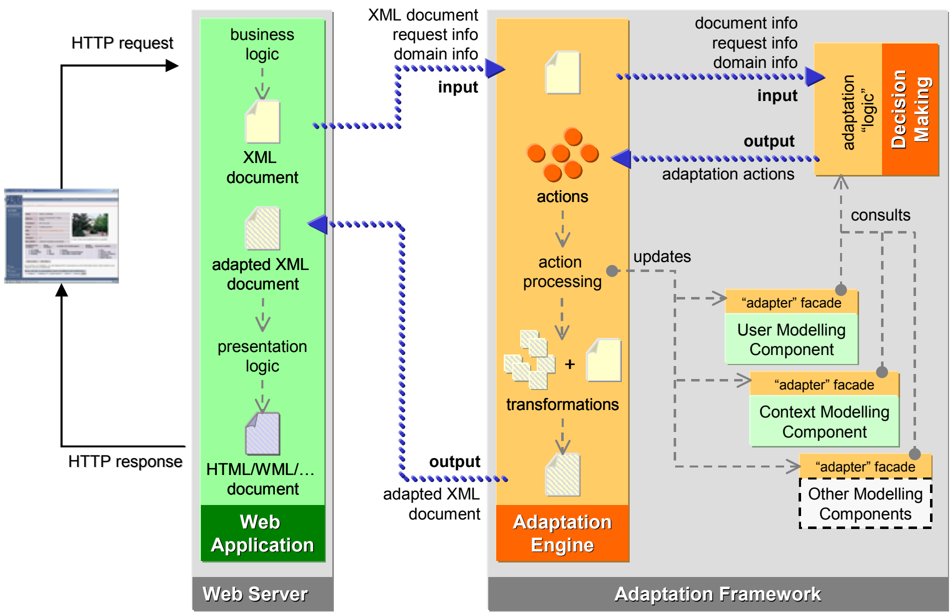


Figure 27: A high level view of the adaptation process, showing framework components and their interactions

Figure 27 depicts a high-level view of the framework components and their interactions. Furthermore, it shows the adaptation process, as this is implemented by the framework components, and relates it to the document processing cycle typical of current web publishing architectures. The next section describes the figure in detail and outlines the responsibilities of the framework's main component, the adaptation engine.

B.2.2.2 Adaptation Engine

The responsibilities of the adaptation engine include the invocation of the decision-making component and the realization of adaptation decisions (expressed through adaptation actions). The rest of this section outlines the adaptation process supported by the framework (and depicted in Figure 27) and the role of the adaptation engine within that process.

The left part of Figure 27 shows an abstract view of a typical web-based system or service: a request is received from the client; business logic is executed as a result of the request; a document is selected / assembled / generated / etc., as the basis for the response; in some cases, the basic document may then be further processed (e.g., to add navigation information, render it to an appropriate output format, etc.); the final document constitutes the system's response to the user's request.

Integration of the presented framework (right part of Figure 27) in an existing web-based system or service involves primarily the invocation of the adaptation engine, at any point in the document processing cycle. In Java-based systems, the adaptation engine can be invoked through a Java class, which acts as a "gateway" into the framework. For non-Java based systems, or when distribution is desirable, there is the possibility to set up the engine as a remotely accessible web service.

When the engine is invoked, the following are passed as parameters: the XML document / fragment itself; information about the request that resulted in the generation of the document / fragment; optionally, domain-specific information, which can be utilized by the decision-making component (see below).

The adaptation engine communicates with the *decision-making component*, which is responsible for determining what type of adaptations are to be performed to the document / fragment. The information sent to the decision-making component include: information about the document itself; information about the request (as above); and, domain-specific information (as above). The decision-making component, in turn, consults the various dynamic and static models that relate to adaptation, in order to decide upon the necessity for, and type of, adaptations to be applied. The result of the decision-making process is a set of *adaptation actions* that are communicated back to the adaptation engine (sections "B.2.2.3 Adaptation Actions" and "B.2.2.4 Adaptation Decisions" provide additional details on the decision-making component and the modeling components).

The adaptation actions are interpreted by the adaptation engine and result into either: (a) XSLT[22] transformations to be applied to the document at hand, or (b) "update" actions which directly or indirectly modify the contents of dynamic models. "Update" actions are communicated to their corresponding modeling components. XSLT transformations are applied directly to the XML document. The transformed document is the output of the adaptation process and is sent back to the invoker (section "B.2.2.3 Adaptation Actions" provides an overview of supported actions).

[22] http://www.w3.org/TR/xslt

It should be noted that, although in the figure the steps described above are depicted as occurring only once per cycle, there is no such restriction in the framework. In fact, when the underlying system creates responses through the composition of fragments, it is necessary that the steps be repeated for each of the fragments (and possibly for the whole as well). Furthermore, fragment-based document composition is not the only scenario that requires repetition of the steps. For example, in PALIO (see "B.2.3 Applying the Framework in PALIO") the adaptation cycle is repeated twice: adaptations in the first cycle, or "stage" are intended to adapt information queries embedded in documents, while adaptations in the second "stage" are intended to adapt the results of the queries and the rest of the document.

Communication between the components that make up the framework is done through a set of well-defined programmatic protocols / interfaces that have been purposely developed to enable the decoupling of the components and facilitate their replacement with alternative implementations. This is also the case for accessing and manipulating information available from sources external to the framework, such as user modeling servers and domain-specific information servers. Implementations of these programmatic interfaces are termed "model adapters".

Additional features of the framework, which are not discussed in further detail here, include the following. The framework is capable of "pushing" information to the user, as a result of adaptation decisions made at run-time, and outside the scope of the request-response cycle. Specifically, the framework supports: retrieval of user- and context-related information through channels other than the browser-server connection (e.g., user location); explicit triggering of the decision-making component to ensure that adaptation logic is evaluated against the new information artifact; and, immediate execution of specialized adaptation actions that may be the outcome of the above process. These actions are apparently focused on the generation / selection of documents that are subsequently communicated to the user. This capacity of the framework has been used in the context of the PALIO information systems to adaptively send information to users (through SMS) on the basis of the user's current location and the users' inferred interest in venues, tourist landmarks, public facilities, etc., in their vicinity.

B.2.2.3 Adaptation Actions

Introduction

Adaptation actions are the means for adaptively transforming documents (or fragments of documents). They are translated internally by the framework's *adaptation engine* into XSLT transformations that are subsequently applied to the document / fragment. It is exactly this role of the adaptation engine that renders it a focal point of the framework's implementation.

In order to support the declarative specification of adaptations, an XML-based language has been developed. The language circumscribes the actions that can be performed, as well as any additional information required by the adaptation engine to successfully carry out these actions. There is no restriction as to the number or type of actions that can be performed on an individual document or fragment. However, since actions are applied sequentially and may effect significant modifications to a document, it may be necessary to handle dependencies between them. The framework makes the fundamental assumption that such dependencies are external to the actions themselves and are expressed and handled externally. In other words, if actions need to be sequenced to achieve the desired adaptation effect, the adaptation engine assumes that such sequencing has already been applied by the decision-making engine, before actions to be performed are communicated back to it. For example, when applying the framework in PALIO, rule priorities and explicit sequencing were used to enforce the correct handling of action dependencies (see "Rule-based Decision Making ").

As already mentioned, adaptation actions are applied by the framework on XML-based documents and fragments. Therefore, it is often the case that authors need to specify the element (or elements), on which, or relatively to which, the actions are applied. This element or elements will be referred to in the rest of this chapter as the "reference" element or elements of the action. Reference elements can be identified in the following ways (common to all adaptation actions):

- Using the element's *tag* name. This method is well suited for applying adaptation actions to elements of a particular type (e.g., hiding / showing all "`<summary>`" elements).
- Using the element's *class* attribute. This method is well suited for the identification of multiple reference elements of different types.
- Using the element's *id* attribute. This method enables the identification of single document elements – an inherent assumption here is that the id attribute of any element should be unique within document scope.
- Using an arbitrarily specified attribute, or set of attributes, of an element. This method is intended to allow for greater flexibility in integrating the framework with existing approach to XML-based web applications and services, without necessitating modifications in the formats / languages already in use.
- Using an XPath[23]-style "selector" expression. This method is intended to support cases where more complex constraints need to be expressed with respect to an action's reference element(s).

[23] http://www.w3.org/TR/xpath

Further to the above, when adaptation actions are part of rules that are embedded within documents (supported by the default rule-based decision making modules), the default "reference" element is the rule (context) node itself. Readers may have noticed that at least two of the above methods for specifying actions' reference elements are directly related to (X)HTML (namely, using the *class* and *id* attributes of elements). This explicit support for (X)HTML constructs is intended to facilitate the application of the framework in systems that do not employ a more "abstract" XML-based document description language. Although their use in other contexts is expected to be limited, they can offer great support in the gradual transition from traditional HTML-centric systems to their adaptive XML-based incarnations.

Framework-supported Adaptation Actions

The basis for deciding which actions need to be inherently supported (as opposed to actions that can be "composed" from more basic ones) has been: (a) an analysis of possible low-level adaptations that can be performed in interactive systems (see, e.g., [Dieterich et al., 1993]), and (b) an analysis of known hypermedia techniques in the literature, with respect to the effort required for synthesizing them from basic actions (e.g., "dimming" would be generally rather straightforward, whereas "sorting" might not even be realistically practical to compose).

Currently, the framework supports the following adaptation action categories: inserting document fragments; removing document fragments; replacing document elements / fragments; sorting document fragments; setting and removing element attributes; selecting among alternative document fragments; applying arbitrary document transformations expressed in XSLT. In addition to the above basic adaptation actions, the PALIO framework also supports the manipulation of variables (i.e., storing and retrieving values from different run-time storage "areas"). Although this feature is not discussed in more detail here, interested readers are referred to the example in section "B.2.2.4 Adaptation Decisions" which demonstrates access to variables in the user model.

Inserting document elements / fragments

This adaptation action enables authors to insert new elements or document fragments into the document. To perform the insertion the author needs to specify the reference element(s) at which the insertion will take place, as well as the relative position of the new element / fragment with respect to the reference element(s). The relative position can be:

- Immediately *before* the reference element.
- Immediately *after* the reference element. If the position attribute is omitted, this is the default position for insertion.

- *Inside* the reference element, as its *first child* (in Document Object Model traversal order). Prerequisites: (a) the reference element must not be an empty element; (b) the element / fragment to be inserted must be valid (i.e., allowed) in the scope of the reference element.
- *Inside* the reference element, as its last child (in Document Object Model traversal order). Prerequisites: (a) the reference element must not be an empty element; (b) the element / fragment to be inserted must be valid (i.e., allowed) in the scope of the reference element.
- *Outside* the reference element, as its parent (in Document Object Model traversal order). Prerequisites: (a) only a single element, or a single-path fragment can be inserted in this fashion (so that the reference element and any contained elements have a unique and unambiguous connection point); (b) the connection point between the inserted element / fragment and the reference element must not be an empty element; (c) the reference element must be valid (i.e., allowed) in the scope of the connection point element.

Example:

```
<adapt:insert
  element-class="example" position="before">
  <div>
    <img
      src="http://url/to/example-icon.jpg"
      alt="Example: " />
  </div>
</adapt:insert>
```

Removing document elements / fragments

This adaptation action enables authors to remove individual or multiple elements or document fragments contained within elements, from the document. To perform the removal the author needs to specify the element(s) to be removed, as well as whether any child elements should also be removed. Specifically, when removing an element, authors have the options to:

- Also *remove* all its *children*: This results in the entire document fragment contained within the identified element to be removed and is the default behavior.
- *Keep* the element's *children*: This results in the element being removed, and its children (in Document Object Model traversal order) to be retained. Prerequisite: all the element's children must be valid (i.e., allowed) in their new scope (the removed element's parent).

Note that the prerequisite for maintaining document validity when one removes an element and keeps its children is quite strict; one should use this capability of the framework very sparingly and only when such validity can be guaranteed.

Example:

```
<adapt:remove
  element-class="example"
  keep-children="false" />
```

Replacing document elements / fragments

This adaptation action enables authors to replace individual or multiple elements or document fragments contained within elements. To perform the replacement, the author needs to specify the element (or elements) to be removed, whether any child elements should also be removed, and the element (or elements) that are to be inserted in the place of the removed element(s). This action can be treated, from a semantic point of view, as a removal action combined with a subsequent insertion action. The distinguishing characteristic of this action, in relation to the aforementioned combination, concerns the possibility to easily transfer the child elements of the element(s) being removed to the element(s) being inserted.

Example:

```
<adapt:replace
  element-tag="header"
  keep-children="true">
  <h1/>
</adapt:replace>
```

Manipulating element attributes

This adaptation action enables authors to set the value of specific element attributes, or remove element attributes entirely. To set an attribute value, the author needs to specify the element (or elements) affected, as well as the name of the attribute to be set and the actual value that it will assume. To entirely remove an element attribute, the author only needs to specify the element (or elements) affected, and the name of the attribute to be removed.

When setting an attribute value, authors have the following options:

- To set the value using the *set-attribute* action, through the respective attribute-value parameter. This approach is better suited for adaptation actions that do not depend on context, but rather assign standard values, known a priori.
- To set the value using the *set-attribute-parameterized* action, which allows for the dynamic definition of the value to be assigned (by encapsulating an expression that resolves to the assignable value). This approach is necessary when one needs to dynamically decide an

element attribute, e.g., when creating a new link towards a document that is believed to be within the user's interests.

Authors must ensure that the following conditions are met when manipulating attributes: (a) when setting an attribute, one must ensure that the new value is a valid (i.e., allowed) one; and, (b) when removing an attribute from an element, one must ensure that this is not a required attribute of the element. The preceding preconditions are not checked automatically.

Examples:

```
<adapt:set-attribute element-id="recommendation"
                     attribute-name="visible"
                     attribute-value="false" />
```

```
<adapt:set-attribute-parameterised>
  <adapt:element-id> recommended-link-1
  </adapt:element-id>
  <adapt:attribute-name> href
  </adapt:attribute-name>
  <adapt:variable
    src="Temporary"
    name="first-rec-link" />
</adapt: set-attribute-by-expression>
```

```
<adapt:remove-attribute
  element-path="//*[@align='left']"
  attribute-name="align" />
```

Selecting among alternative document elements / fragments

This adaptation action is intended to facilitate the selection of one (or more) among the alternative fragments declared within the document. In the current version of the adaptation framework, support for this action is "intrusive" to the document being adapted, as it requires that alternatives be marked as such in the document itself. It is planned that future versions of the framework will lift this restriction.

The use and selection of alternative content fragments requires authors to:

- Firstly, introduce into the document (and identify as such) the alternative (but not necessarily mutually exclusive) fragments.
- Select one or more of them through the respective adaptation actions. All alternatives that are not explicitly selected are automatically removed from the document.

The specification of alternatives adheres to the following simple rules:

- Alternatives are encapsulated under an appropriate container, which is assigned a distinctive *id* or *class*.

- Each alternative is also assigned a distinctive *id* or *class*, so that it can be distinguished and individually referred to by adaptation actions.
- No other information is specified about the alternative fragments within the document, to allow for maximum flexibility as to how they are selected and combined – the respective, semantic-level information is assumed to be maintained externally to the document.

The selection of alternatives can happen in one of the following ways:

- Through the *select-alternative* adaptation action, which selects one of the alternatives and discards the rest. The parameters include the identification of the alternatives' container and the identification of the sole alternative to be retained.
- Through the *multi-select-alternative* adaptation action, which selects one or more of the alternatives and discards the rest. The parameters include the identification of the alternatives' container and the identification of each of the alternatives to be retained.

Note that the *select-alternative* adaptation action is, in effect, a special case of the multi-select-alternative adaptation action. It is provided as a separate action solely for reasons of authoring convenience.

Example

In the document:

```
<adapt:alternatives class="modality">
  <adapt:alternative class="video">
    <object ... />
  </adapt:alternative>
  <adapt:alternative class="image">
    <img ... />
  </adapt:alternative>
  <adapt:alternative class="text">
    <p> text ... </p>
  </adapt:alternative>
</adapt:alternatives>
```

Single selection action:

```
<adapt:select-alternative
  among="modality"
  select="text" />
```

Multiple selections action:

```
<adapt:muli-select-alternative among="modality">
  <adapt:select select="image" />
  <adapt:select select="text" />
</adapt:muli-select-alternative>
```

Sorting document elements / fragments

This adaptation action is intended to specifically facilitate a common (but difficult to define and implement with basic building blocks) technique in AHS – the sorting of document fragments, links, etc. As in the case of the definition and selection of alternatives, sorting is currently "intrusive" to the document structure, although this can be expected to change in subsequent versions of the framework. Along these lines, sorting adaptations require two steps of specification on the part of the author: firstly, the list of document elements / fragments that are to be sorted needs to be defined, within the document; secondly, a respective sorting adaptation action needs to be authored.

The specification of elements / fragments to be sorted adheres to the following simple rules:

- There exists a generic container named *list*, which holds the items (elements / fragments) to be sorted. Note that the container has been called "list" only to ensure familiarity of authors with the container concept and not because the container provides any facilities for traversing its contents in a list-like fashion.
- The container holds items, each of which must provide a *name* attribute. The latter is used to relate the item, directly or indirectly, to a value that will be used subsequently for sorting the items.
- The container may also hold additional supporting content that accompanies the items to be sorted.

In more detail, the specification of the container involves:

- The assignment of a unique or shared identifier to the list through the *id* or *class* attribute respectively. This step is only necessary, if the corresponding methods for reference element identification are employed.
- The specification of whether the container's non-item contents are to be removed or retained, if all the items are removed from the container as part of the sorting process (more on this below).

The specification of each of the items to be sorted is done through the *item* element and requires only the identification of a *name* for the item, which will be used during sorting to associate the item with a concrete value. As far as the actual sorting adaptation is concerned, this is performed through the *sort* action and can follow two alternative approaches, varying in the way in which item names are associated with values:

- In the first case, all items derive their values from the same data source, and, even more specifically, from the same "path" in the data source (i.e., the item names can be converted to actual variable identifiers by prepending to all of them the same path string).

- In the second case, the items derive their values from different data sources, or from the same data source but from different "paths" therein. As a result, each item needs to be associated with its respective value separately.

The former approach is implemented through the use of one set of *value-src*, *value-path* elements. The latter is implemented through the use of the *relate-item* element, which encompasses individual *value-src*, *value-path* elements for each item. In the latter case, the author has to additionally define what happens to items that are not explicitly associated with a value. The options are obviously to either keep or discard them.

The specification of sorting behavior includes two additional, optional components:

- The identification of the *maximum number* of items to be retained. This has the effect that, after sorting, only the specified number of items is kept in the document, while the rest are automatically removed.

- The identification of a *threshold value* that acts as a cut-off point for items. Specifically, items whose values are below the threshold are automatically excluded from the sorting and removed from the document. The semantics and domain of the threshold are not strictly specified, enabling authors to use them in any way they find appropriate.

The last constraint (threshold value) may actually result, under specific circumstances, in a sorting container that does not include any items (because all are below the threshold). This is the reason why containers have to explicitly qualify their miscellaneous (non-item) contents for retention, in the case of absence of items after the sorting adaptation. A good rule of thumb to use in this case is to include in the container all elements that are required to "support" the items when these are present. Thus, when no items are left, the container with all its contents can safely be removed from the document.

Last but not least, the *sorting order* to be applied to the items is defined as an attribute of the *sort* adaptation action.

The following is an example of sorting the entries of a HTML drop-down selection box, containing possible "destinations" from the current page, with each page being thematically related to the one currently viewed by the user.

Example

In the document:

```
<adapt:list
  class="related-pages"
  if-empty="remove">
```

```
    <select name="destination">
      <adapt:item name="accommodation">
        <option
          value="{page URL}">Accommodation</option>
      </adapt:item>
      <adapt:item name="food-and-drink">
        <option
          value="{page URL}">Restaurants and
          bars</option>
      </adapt:item>
      <adapt:item name="sightseeing">
        <option
          value="{page URL}"> Museums and
          monuments</option>
      </adapt:item>
    </select>
</adapt:list>
```

Sorting action – first variant (item "names" are directly associated to value sources (by defining the source and prepending the common "path" to the value):

```
<adapt:sort
  list-class="related-pages" order="descending">
  <adapt:limit-items-to> 2
  </adapt:limit-items-to>
  <adapt:value-threshold> 0.7
  </adapt:value-threshold>
  <adapt:value-src> DPS </adapt:value-src>
  <adapt:value-path>
    interests.ontology.activites
  </adapt:value-path>
</adapt:sort>
```

Sorting action – second variant (item "names" cannot be directly associated to value sources, either because they are derived from multiple sources, or because there exists no straightforward one-to-one mapping between items and their intended values):

```
<adapt:sort
  list-class="related-pages" order="descending">
  <adapt:limit-items-to> 2
  </adapt:limit-items-to>
  <adapt:value-threshold> 0.7
  </adapt:value-threshold>

  <adapt:relate-item item-name="accommodation">
    <adapt:value-src> DPS </adapt:value-src>
    <adapt:value-path>
       interests.ontology.housing.accomodation
    </adapt:value-path>
  </adapt:relate-item>
```

```
  <adapt:relate-item item-name="food-and-drink">
    <adapt:value-src> DPS </adapt:value-src>
    <adapt:value-path>
       interests.ontology.food-and-drink
    </adapt:value-path>
  </adapt:relate-item>

  <adapt:relate-item item-name="sightseeing">
    <adapt:value-src> DPS </adapt:value-src>
    <adapt:value-path>
       interests.ontology.activities.sightseeing
    </adapt:value-path>
  </adapt:relate-item>

  <adapt:unrelated-items action="keep" />

</adapt:sort>
```

Applying arbitrary XSL transformations

This adaptation action is intended to cover any complex cases that cannot be addressed through the use of the basic adaptation actions listed above. Specifically, this action only defines that an external XSLT file is to be applied to the document (fragment), providing the URL of that file as an attribute. The capability to apply such arbitrary transformations is possible as a result of the framework's explicit representation of document / fragment adaptations as pure XSLT templates. In the case of the action under discussion, the framework simply adds the defined templates to the ones that resulted from the interpretation of the adaptation actions.

To support the framework's employment in the PALIO information systems (see section B.2.3), a number of XSLT files were made available to authors, which address some non-trivial, presentation-level adaptations, such as the transformation of nested lists to tables and vice versa, etc.

Synthesizing Adaptation Techniques

To enable the creation of complex adaptive behaviors in various application domains, a considerable number of "hypermedia techniques" have been developed and reported in the literature in relation to AHS. As already mentioned, the term "adaptive hypermedia techniques" is used to refer to modifications which can be adaptively applied to hypermedia documents and which can be synthesized to arrive at higher-level adaptation *method*, such as "additional explanations" and "global guidance" [Brusilovsky, 1996] (see also Figure 26 which presents a partial classification of the "server-side" adaptation techniques).

Table 4: Synthesizing Adaptive Hypermedia Techniques [Paramythis et al., 2003b]

	Atomic Adaptation Actions in the Adaptation Framework / Adaptive Hypermedia Techniques	Inserting document elements / fragments	Removing document elements / fragments	Replacing document elements / fragments	Manipulating element attributes	Selecting among alternative document elements / fragments	Sorting document elements / fragments
Adaptive Navigation	**Link annotation**	✓			✓		
	Link hiding		✓		✓		
	Link disabling		✓	✓	✓		
	Link removal		✓	✓			
	Link sorting						✓
	Link generation	✓		✓			
	Direct Guidance	✓	✓	✓		✓	
Adaptive Presentation	**Adaptation of modality**					✓	
	Inserting fragments	✓					
	Removing fragments		✓				
	Altering fragments	✓	✓	✓	✓		
	Sorting fragments						✓
	Dimming fragments				✓		

One of the desiderata for the framework under discussion was to provide adequate support for all the techniques in Figure 26 in a way that allows for future refinements and extensions. As already mentioned, adaptation actions are intended as lower-level building blocks that can be used in isolation or in combination to synthesize higher-level adaptation techniques. Table 4 presents the adaptive hypermedia techniques that are directly supported by the framework, associating them with the adaptation actions that can be used (or are required) to implement them [Paramythis et al., 2003b]. Note that, although there is a distinction in the domain of the techniques (i.e., whether they address presentation or navigation), as per the classification depicted in Figure 26, such a distinction is not present in the adaptation actions themselves. As the latter are at a lower level of abstraction, they are necessary in both domains.

Model-update actions

Model-update actions are defined, in the framework, as model variable manipulations. In more detail, the framework supports the concept of model *variables*, variable *namespaces*, and variable *manipulation*. A "variable" can be any piece of information that resides in an external modeling component (e.g., the probability that the user is interested in a particular tourist venue is a user model variable that can be accessed from the user modeling server). The concept of variables has been apparently borrowed from programming constructs. Our goal has been to provide adaptation designers / authors with a familiar metaphor that would enable them to think uniformly about, while abstracting over, information interchange with modeling components.

We considered variable manipulation an appropriate metaphor, as it has explicit and intuitively clear affordances for: (a) storage of values; (b) retrieval of values; (c) application of operations on values; and (d) composition of types. Integrated modeling components are required to support at least the storage and retrieval operations. The semantics of each may vary depending on the model and modeling approach and are not strictly defined in the context of the framework.

Variable namespaces are an optional extension to the concept of variables. They are intended, on the one hand, to ensure uniqueness of variable references where potential problems might exist, and, on the other hand, to structure access to the "global" variable space that would be created if all modeling components pooled their variables together. As namespaces are optional, the framework does not assume their presence.

B.2.2.4 Adaptation Decisions

Support for Alternative Modeling and Decision Making Components

As already discussed, one of the main prerequisites for the adaptation framework has been the possibility to employ alternative approaches to various aspects of the adaptation process. Such flexibility should explicitly be achieved: (a) for the modeling process (i.e., the derivation of dynamic user-, context-, or other models from monitoring data), and (b) the process of deciding adaptations on the basis of static and dynamic models available in the system. As is often the case in software engineering, the aforementioned requirements were addressed by strictly specifying input and output communication protocols for the respective components, and, otherwise, making as few assumptions as possible about their inner workings.

Starting from the modeling part, the framework defines the concept of a "model adapter". Such adapters are, in practice, implementations of programmatic interfaces, and are responsible for mediating between the modeling component / server and the framework itself. The adapters undertake a small number of tasks, which have been iteratively refined to enable maximum flexibility without increasing the cost of adapter implementation. The first task of model adapters is to propagate evidence derived from monitoring users' interaction with the system (or changing context parameters, etc.) to the modeling component. Evidence may follow the concept of variables and variable namespaces (see next section for an explanation), although this is not required by the framework. In fact, different implementations are free to employ whatever scheme is appropriate for their needs, by transforming the evidence collected by the system into whatever types of constructs are required.

The framework makes absolutely no assumptions about the approach used to derive, or infer model attributes from the evidence communicated. Nevertheless, it does assume that models can be queried, and it does expect the return values of such queries to be either among a number of "basic" types, or to be composites which contain other composites or basic types. This limits somewhat the spectrum of modeling approaches that can be employed, precluding specifically ones that follow a "one-step" approach to adaptation (e.g., solutions that employ neural networks-based intelligence). It is argued, however, that this is not a significant limitation of the system, as modern approaches to adaptation require that models are "transparent" [Höök et al., 1996], or "scrutable" [Kay, 1995], which also leaves out such "opaque" modeling approaches.

As mentioned above, the contents of models can be "queried" and accessed using the predefined programmatic interfaces of model adapters. The assumed querying capabilities are quite simple, and typically consist of applying common set operators and functions on sets of data. The reason for keeping requirements at such low levels has been to allow for the possibility of supporting them in the model adapter implementation, if they are not inherently supported by the modeling component / server. As in the case of sending evidence, variables (and namespaces) can be used in accessing model attributes, although their use is not required.

As far as decision making is concerned, the approach taken by the framework is quite similar to that of modeling. In more detail, the framework communicates with the decision making component through well-defined programmatic interfaces. Information sent by the framework includes the document currently being considered for adaptation, as well as a number of run-time, adaptation context-specific parameters that may be utilized in the decision-making process. The framework expects the process to result in a set of adaptation actions to be applied to the document at hand (the set may, of course, be empty).

Using adaptation actions as the "output language" for adaptation decisions involving XML-based documents, becomes, thus, one of the most important facets of the framework since it is one of the few "fixed" parts of the proposed approach. To make a real-world analogy, we consider adaptation actions to be the *Lingua Franca* of sub-systems that need to communicate information about *what* should be adapted and *how*. It is argued that, on the one hand, adaptation actions are both sufficient for describing practically all types of server-side adaptations that have been reported in the literature, and, on the other hand, are a flexible abstraction to be employed in the context of adaptation decisions.

The possibility to interchange modeling and decision-making components has been used extensively in the development of the PALIO information systems. In several occasions, "dummy" implementations have been employed to facilitate testing, simulate specific operational conditions, and explore alternative approaches to decision making. Regarding the latter, in particular, we have been able to experiment with a Bayesian networks-based approach to decision making, as an alternative to the default rule-based one (see below). This required no modifications to the system, other than the actual switching of the respective modules.

Accessing and Manipulating Information in Models and Ontologies

The concept of model variables has already been referred to quite a few times. Although readers may have already formed an overview of what these are, we would like to more precisely qualify and rationalize their use in this section. To start with, the concept of variables has been obviously borrowed from programming constructs. Our goal has been to provide adaptation designers / authors with a familiar metaphor that would enable them to think uniformly about, while, at the same time, abstracting over, information interchange with modeling components. We considered variable manipulation an appropriate metaphor, as it has explicit and intuitively clear affordances for: (a) storage of values; (b) retrieval of values; (c) application of operations on values; and (d) composition of types.

Model adapters are required to support at least the storage and retrieval operations. The semantics of each may vary, depending on the model and modeling approach and are not further defined in the context of the framework. Furthermore, to lift possible restrictions to the underlying modeling operations, the framework does not even make the assumption that a variable the value of which has just been set, will return that same value of immediately queried (although authors that are familiar with the modeling approach used can make such assumptions themselves).

Variable namespaces are an optional extension to the concept of variables. They are intended, on the one hand, to ensure uniqueness of variable references where potential problems might exist, and, on the other hand, to structure access to the "global" variable space that would be created if all modeling components pooled their variables together. Namespaces, as mentioned earlier in this chapter, are not required or assumed by the framework. It is in fact, an aspect that may be decided on a per-case basis. For example, it is possible to use namespaces for the user modeling component, and use a "flat" space for the context modeling one.

A special case in the system-maintained models is the domain model. For the needs of this chapter, we will informally define the domain model to be a representation of the domain-specific information space. We will further assume that the domain model can be used in at least two complementary ways: to examine aspects of the domain at hand; and, to model user behavior, interests, etc., with respect to the domain. The latter use renders the domain model a potential, implicit part of other models. The former use gives rise to new possibilities in terms of information retrieval.

Specifically, if one makes the further assumption that the domain model is structured as an information ontology, then information retrieval and manipulation in adaptive hypermedia systems can employ tools and techniques from the respective field. In PALIO, for example, it was possible to define an information ontology targeted on tourism, and, based on that: (a) create "information adapters" which enabled us to abstract over different static and dynamic information sources; (b) create a dedicated, SQL-like ontology query language for the retrieval of information from information adapters; and, (c) introduce adaptation steps that enabled the adaptive modification of the system's information retrieval behavior.

Rule-based Decision Making

The primary decision engine implemented for PALIO (see "B.2.3 Applying the Framework in PALIO") is rule-based. Although first order logic rule engines can easily be plugged into the framework (the latter having been specifically designed to allow for that), it was decided that, to facilitate the wide adoption of the framework, a simpler and more accessible approach was in order. Along these lines, a new rule language was created, borrowing from control structures that are commonly supported in functional programming languages. The premise was that such structures were much more familiar to designers of adaptive systems, while at the same time, they afforded lower degrees of complexity when it came to understanding the interrelations and dependencies between distinct pieces of adaptation logic. A similar approach was first applied in the AVANTI project with very good results (see also previous Chapter).

An XML binding was developed for the aforementioned rule language, while a rule interpreter and a corresponding rule engine supported the run-time operation of the system. Adaptation rules expressed in such a rule language may be defined either in external files or embedded in the document to be adapted. Rules embedded in documents are constrained in that they can only be applied within the specific document in which they reside, and therefore are not reusable.

Rules external to documents are organized into rule-sets; typically, each rule-set resides in a different file, but this is a matter of organization, rather than a constraint imposed by the framework. Rule-sets have a specific *name* and *scope*. The name may be used to refer to the rule-set within configuration files and can be used, for example, to enable/disable a whole set of rules, by referring to it via the name of the enclosing rule-set. The possible values of the scope attribute are *global*, *service* and *local*. These denote that the rules defined in a rule-set may apply to all documents, to documents belonging to a specific service (basic units of functionality organization in PALIO), or to specific documents indicated through configuration files, respectively.

Every rule, whether internal or external to the document, has the following attributes: (a) the rule *name*, which is an identifier for the rule; (b) the rule *class*, which is optional and may be used as an alternative way for grouping rules; (c) the rule *stage*, which defines whether the rule should be applied during the first, or the second adaptation phase; and (d) the rule *priority*, which provides a partial ordering scheme for rule evaluation and application and may take the values of *high*, *medium* or *low*. The *stage* property defines whether the adaptation rule should be applied before or after querying the IS (if this is required to process the document). Adaptations performed during the first phase (before querying the IS) are either unconcerned with IS-ontology data, or adapt (i.e. apply modifications to) IS queries. Rules applied during the second phase (after querying the IS) are concerned with IS-ontology data.

The framework currently supports three types of rule constructs: *if-then-else* rules, *switch-case-default* rules, and *prologue-actions-epilogue* rules. If-then-else rules are the simplest type of conditional rules; they bind sets of adaptation actions with the truth-value of a conditional expression. Following the example of functional languages, an if-then-else rule is composed of a *condition* part (containing an expression based on adaptation determinants), a *then* part (containing the adaptation actions to be taken whenever the condition is satisfied), and an optional *else* part (containing the actions to be taken when the condition fails).

Switch-case-default rules can be used to relate multiple values (outcomes of run-time expression evaluation) to sets of adaptation actions. In this case, adaptation actions are executed if the value of a variant (expression) is equal to a value, or within the range of values, specified as a selection case. The switch-case-default rule construct supports the definition of multiple cases and requires the additional definition of an (optionally empty) set of default adaptation actions to be performed if none of the defined cases apply.

The prologue-actions-epilogue construct is intended mainly for the definition of unconditional rules. In other words, the specific rule construct is provided to support the definition of (sets of) actions to be performed at a particular stage in the sequence of adaptations. Although the construct can be simulated with an if-then-else construct (where the condition is always true), the prologue-actions-epilogue structure uses the concepts of *action-sets* to provide an explicit separation between "preparatory" actions, the adaptations themselves, and "clean-up" actions. This separation allows for better rule structuring and improves the maintainability of the rule definition. A very common use of the prologue and epilogue parts is the creation / assignment and retraction of variables that are used by the adaptation actions (e.g., in order to determine the point of application of an action). A simple case of the construct is included in the example presented later in this section.

The conditional parts of first two rule constructs presented above, as well as the definition of variants and variant ranges in the case of the switch-case-default construct, are composed of expressions. These consist, as one would expect, of operands and operators applied on the operands. The main supported operand types in the framework are: *string*, *number*, *Boolean*, *date*, *location* and *null*. The *string* type is used for character sequences. The *number* type is used for all kinds of numerals including integers, floating point numbers, etc. The *Boolean* type is used to express truth values and can only assume the values *true* or *false*. The *date* type is used for absolute, relative and recurring dates and temporal ranges. The *location* type is used to express absolute and relative geographical locations. Finally, the *null* type is a utility type with restricted use; authors may use it to determine whether a variable exists in the scope of a particular data source, and whether its type has been set. The operators supported by the PALIO framework can be classified into the following main categories: comparison operators (>, >=, <, <=, <>), mathematical operators (+, -, *, /, %), logical operators (*and*, *or*, *not*), string operators (*concatenate*, *contains*, and *substring*), date-specific operators (*get-year*, *get-month*, etc.), and location-specific operators (*near* and *distance*).

The following is a simplified example taken from PALIO, and specifically from an adaptation rule-set that controlled adaptations related to the detailed presentation of accommodation venues in an accommodation service. The example demonstrates two subsequent steps:

(a) First, the most interesting accommodation facility for the current user is identified. This is done by retrieving from the user modeling server the probabilities associated with the user's interest in any of the facility categories found under "*accommodation.facilities. establishment.public*", and selecting among them the one with the highest probability (note also that a threshold value is set of 0.6 to ensure that the rule is applied only when there is sufficient evidence for a user's interests).

(b) Occurrences of the identified facility (if any) are emphasized for easier recognition and faster access by the user.

```
<adapt:ruleset
  name="accomodation-details" scope="service">

  <adapt:rule
    name="get-most-interestint-facility"
    stage="second" priority="medium">

    <adapt:action-set>
      <adapt:actions>
```

```
        <adapt:bind
          name="user.most_interesting_facility"
          src="Temporary" type="string">
          <adapt:get-max
            threshold="0.6" return="@name">
            <adapt:variables
              from="user.interests.accommodation.
                 facilities.establishment.public"
              src="DPS"/>
          </adapt:get-max>
        </adapt:bind>

      </adapt:actions>
    </adapt:action-set>

   </adapt:rule>

 <adapt:rule
  name="emphasize-most-interesting-facility"
  stage="second" priority="medium">

  <adapt:if>
     <adapt:condition>
       <adapt:ne>
         <adapt:variable
           name="user.most_interesting_facility"
           src="Temporary" type="string"/>
         <adapt:constant
           type="null" value="null"/>
       </adapt:ne>
     </adapt:condition>

     <adapt:then>
       <adapt:set-attribute-parameterised>
         <adapt:select-element>
            <adapt:select-element-type>
              //accomodation/facility
            </adapt:select-element-type>
            <adapt:select-element-attribute>
              type
            </adapt:select-element-attribute>
            <adapt:select-element-attribute-value>
              <adapt:variable
                 name="user.most_interesting_facility"
                 src="Temporary" type="string"/>
            </adapt:select-element-attribute-value>
         </adapt:select-element>
         <adapt:attribute-name>
            style
         </adapt:attribute-name>
         <adapt:attribute-value>
            emphasized
         </adapt:attribute-value>
       </adapt:set-attribute-parameterised>
```

```
      </adapt:then>
    </adapt:if>
  </adapt:rule>

 </adapt:ruleset>
```

B.2.3 Applying the Framework in PALIO

B.2.3.1 The PALIO System Architecture

The PALIO project (see section "A.2.2 Work Context and Research Projects") proposed a new approach to the provision of tourist services in an integrated, open structure, based on the concurrent adoption of the following concepts: (a) integration of wireless and wired telecommunication technologies to offer services through both fixed terminals in public places and mobile personal terminals (e.g., mobile phones, PDAs, laptops); (b) location awareness to allow the dynamic modification of information presented (according to user position); (c) adaptation of the contents to automatically provide different presentations depending on user requirements, needs and preferences; (d) scalability of the information to different communication technologies and terminals; (e) interoperability between different service providers in the envisaged wireless network and the web.

The main components and communication channels PALIO are depicted in Figure 28. The *Service Control Centre* (SCC) is the central component of the PALIO system. It serves as the access point and the runtime platform for the system's information services. The SCC is the framework upon which other services are built. It provides the generic building blocks required to compose services. Seen from a different perspective, the SCC acts as a central server that supports multi-user access to integrated, primary information and services, appropriately adapted to the user, the context of use, the access terminal and the telecommunications infrastructure.

The *User Communication Layer* (CL)[24] encapsulates the individual communication servers (Web gateway, WAP gateway, SMS gateway, etc.) and provides transparent communication independent of the server characteristics. This component unifies and abstracts the different communication protocols (e.g., WAP, HTTP) and terminal platforms (e.g., mobile phone, PC, Internet kiosk). Specifically, the CL transforms incoming communication from the user into a common format, so that the rest of the system does not need to handle the peculiarities of the underlying communication networks and protocols.

---

24 The term "layer" is used in the PALIO project for historical reasons; the CL is not a layer in the sense of layered software architecture, but rather a component.

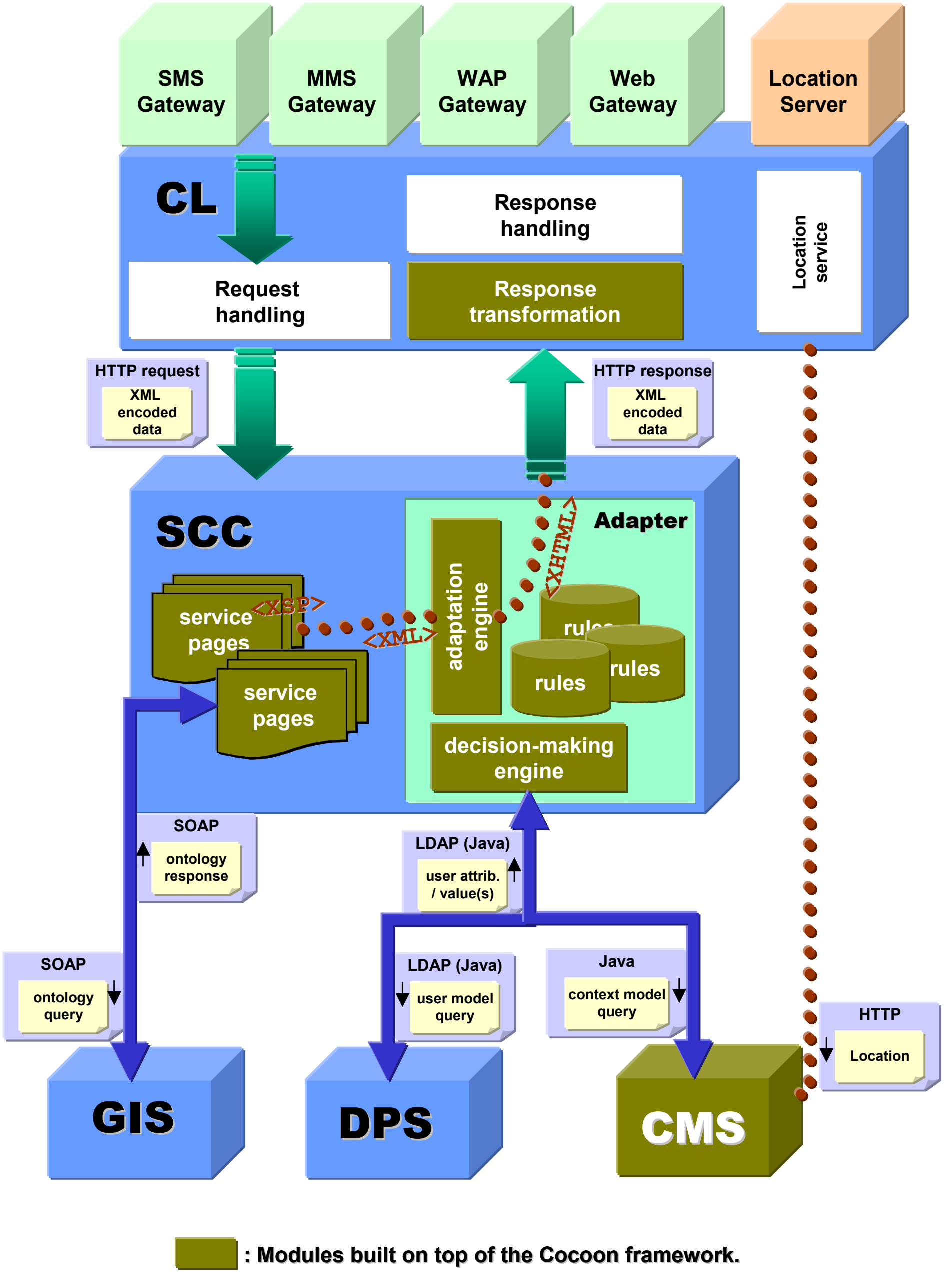


Figure 28: Components and communication channels in the PALIO framework. [Stephanidis, Paramythis et al., 2004]

Symmetrically, the CL transforms information expressed in the aforementioned common format into a format appropriate for transmission and presentation on the user's terminal. In addition to the above, information regarding the capabilities and characteristics of the access terminal propagates across the PALIO system. This information is used to adapt the content and presentation of data transmitted to the user, so that it is appropriate for the user's terminal (e.g., in terms of media, modalities and bandwidth consumption).

The *Generic Information Server* (IS) integrates and manages existing information and services (which are distributed over the network). In this respect, the IS acts as a two-way facilitator. Firstly, it assembles appropriate content and data models (in the form of an information ontology and its associated metadata), upon which it acts as a mediator for the retrieval of information and the utilization of existing services by the SCC. Secondly, it communicates directly with the distributed servers that contain the respective data, or realize the services.

Adaptation support comes, apparently, from the framework described thus far in this chapter. The user modeling server used in PALIO was *humanIt's*[25] *Dynamic Personalisation Server* (DPS). The DPS maintains four models: a *user* model, a *usage* model, a *system* model, and a *service* model. In general, user models consist of a part dedicated to users' interests and preferences, as well as a demographic part. In PALIO, the principal part of a user model was devoted to representing users' interests and preferences. This part's structure was compliant with the information ontology, providing PALIO with a domain taxonomy. This domain taxonomy was mirrored in the DPS-hosted system model.

Usage context modeling in PALIO was undertaken by a purposely developed *Context Modeling Server* (CMS). A usage context was defined, in PALIO, to include all information relating to an interactive episode that is not directly related to an individual user. Along these lines, a context model may contain information such as: characteristics of the access terminal (including capabilities, supported mark-up language, etc.), characteristics of the network connection, current date and time, etc. In addition to these, the PALIO CMS also maintained information about: (a) the user's current location, and (b) information related to *push* services that users are subscribed to.

The PALIO system was implemented on top of the Cocoon[26] publishing framework. The latter was used as the ground platform in the implementation of the SCC, to generate information pages that were delivered to the users in a format supported by their terminal devices. In brief, services in PALIO comprised: (a) Pages containing: static content expressed in XHTML, dynamic content expressed in the PALIO content language, information retrieval queries expressed in the PALIO query and ontology languages, and embedded adaptation rules. (b) External files containing adaptation logic and actions (including files that express arbitrary document transformations in

[25] http://www.humanit.de/

[26] Apache Cocoon is an XML publishing framework that facilitates the usage of XML and XSLT technologies for server applications. Designed around pipelined SAX processing, to benefit performance and scalability, Cocoon offers a flexible environment based on the separation of concerns between content, logic and style–the so-called *pyramid model of web contracts*. More information can be obtained from the project's homepage, at: http://xml.apache.org/cocoon/index.html.

XSLT format). (c) Configuration files specifying the mappings between adaptation logic and service pages. (d) Other service configuration files, including the *site map* (a term used by Cocoon to refer to mappings between request patterns and actual service pages).

The adaptation framework was used in PALIO to support several types of adaptation at the service level [Paramythis et al., 2003a]. Examples include: (a) Users can be assisted by the system in retrieving information in accordance to their requirements and preferences, through: adaptive form simplification and pre-filling; adaptive query augmentation; adaptive filtering and sorting of query results; adaptive selection of "levels of detail"; adaptive selection of pieces of information to include in "pages"; etc. (b) Users can receive recommendations on items that might be of interest to them, on the basis of their individual profiles / models, but also drawing from collective group experience and "opinions". (c) User requests for information or services can be automatically augmented with location, through the employment of intuitive geographical concepts employed ("here", "near" and "far"), with automatic "scaling" possible based on service semantics (e.g., "near" is different when a person on foot is looking for a restaurant nearby, and different when a person driving a car is looking for a parking lot nearby). (d) Specialized, added value, location-based and user-aware services and service components are possible (e.g., recommending places that fall within the users' interests and are in their current vicinity – "what interesting things are around?"). (e) A wide variety of access devices and modalities are supported through the automatic adaptation of presentation and content to account for device capabilities and network characteristics, coupled with the framework's support for transforming to/from many formats and markup languages.

The power of the presented adaptation framework is perhaps better illustrated by the way in which the existing PALIO information systems address the issue of accessibility [Emiliani et al., 2003]. Specifically, accessibility is addressed at two complementary levels: presentation and content. Presentation-oriented adaptations are targeted at ensuring that the interactive front-end of the services is accessible and usable by different categories of disabled people. In this respect, PALIO explicitly accounts for blindness, color blindness, low vision, and motor impairments that may affect a user's interaction with the system. At the content level, services can automatically provide users with seldom sought after information that is, however, of particular relevance to their disability (e.g., the accessibility of a venue to wheelchair-bound persons). Such information is also utilized internally by the system to tailor the services themselves to individual user requirements.

B.2.3.2 A Brief Example

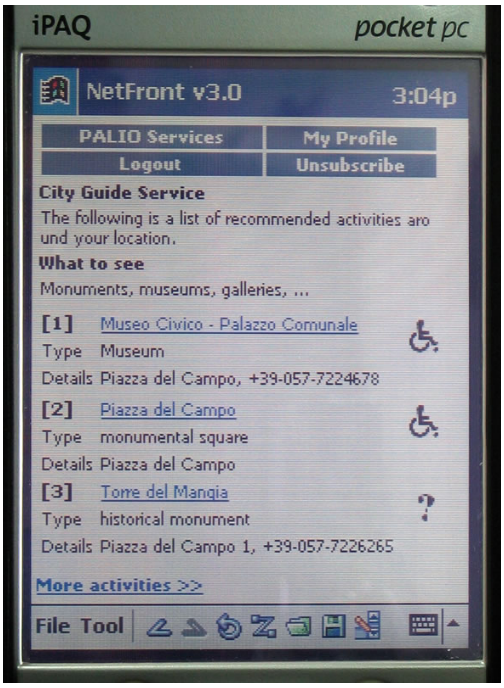
To better illustrate the capabilities of the framework, this section presents an example from the PALIO information system installed for demonstration and testing purposes in the city of Siena, Italy. For brevity, the example will not delve into technical details; instead, it will focus on the results of applying the PALIO framework on the following two fictional interaction scenarios.

> **1st scenario:** An English-speaking wheelchair-bound user is in Piazza del Campo (Siena, Italy) in the morning. She is interested in sightseeing and prefers visiting monuments and museums. She is accessing the system from her palmtop, which she rented from the Siena Tourist Bureau and which is fitted with a GPS unit. She accesses the **City Guide** service and asks for recommendations about what she could see or do next.

> **2nd scenario:** An Italian-speaking able-bodied user is also in Piazza del Campo (Siena, Italy) around noon. He hasn't used the system before and, therefore, there is no information about what he might prefer. He is accessing the system from his mobile through WAP. He also accesses the **City Guide** service and asks for recommendations of what he could do next.

The result of the first user's interaction with the PALIO system is shown in Figure 29a. Relevant characteristics include: (a) the presentation language is English; (b) the front-end is tailored for a small-screen terminal capable of color and graphics; (c) the system's recommendations are in accordance with the user's preferences; (d) recommended sites are in the immediate vicinity of the user; and (e) accessibility information is provided immediately (at the first level), since this information will impact on whether the user can visit the place or not.

The result of the second user's interaction with the PALIO system is shown in Figure 29b. Relevant characteristics include: (a) the presentation language is Italian; (b) the front-end is tailored for tiny-screen terminals without assumptions made about color and graphics; (c) the system's recommendations are derived from preferences for the user's group–asterisks next to each recommendation indicate other users' collective assessments of the venue recommended; (d) recommendations include a wider range of activities (e.g., sightseeing and eating); (e) one type of activity (eating) is relevant to temporal context (it's noon); (f) recommended sites are in the general vicinity of the user; and (g) accessibility information is not provided immediately (at the first level).



(a) GPS-enabled PDA

(b) WAP terminal

Figure 29: Output on different devices from the PALIO Information System in Siena, Italy.

B.2.4 Discussion

This chapter has presented an adaptation framework for hypermedia systems, which is generic in nature, allows for alternative approaches to various aspects of adaptation and is orthogonal to traditional web-publishing approaches, thus making it possible to easily integrate it with them.

The employment of the adaptation framework in PALIO has provided valuable feedback in at least three directions: (a) orthogonality of the adaptation framework to the underlying web "serving" architecture; (b) authoring support for designing / defining adaptations, and (c) user reactions to the adaptive facilities of the system.

Regarding the first of the above aspects, namely orthogonality of the framework to the web "serving" architecture, we found that our goal was largely achieved, as we were able to easily integrate with the Cocoon-based, service-oriented approach at the core of PALIO. The easiest part of the integration turned out to be the link between the publishing cycle and the adaptation engine (passing in XML documents, receiving their adapted counterparts, and introducing them again into the cycle). Somewhat more challenging was the propagation of monitoring information to the appropriate modeling servers. The main encountered difficulties were: (a) the need to synchronize between the application domain model (i.e., information ontology), the user interests' representation and the "evidence" sent by the system for monitoring; (b) the limited guidance from previous results published in the literature as to how monitoring should approach potentially "mixed initiative" requests (i.e., requests that may have the explicit or implicit result of adaptations introduced in a document as part of a prior processing cycle).

Although the former of the two issues is probably self-evident, the second may require some additional explanation. Specifically, an issue that was repeatedly encountered was differentiation between evidence resulting from direct user behavior (e.g., user visiting the page of a hotel because they found its description interesting), versus evidence indirectly resulting from adaptations that were introduced by the system (e.g., hotel summary and overview automatically included in a page in the form of a recommendation). It was specifically found that differentiating between the two was possible, but the framework did not provide adequate support for generalizing the relevant behavior, resulting in repetitive work on the part of the adaptation designers / authors.

Another important category of findings, as already mentioned, concerned the level of authoring support required for designing / defining adaptations. In PALIO, adaptations were designed and implemented by a multidisciplinary group bringing together expertise in Human-Computer Interaction, Adaptation Theory, web-based development, XML- and XSL- based document publishing, etc. Furthermore, the adaptation rules and actions were all written "by hand", using only the support offered by XML schema aware editors.

The synthesis of the team, as well as the shifting focus from design towards implementation, made apparent the need for multi-level design and authoring support tools for adaptation. Informal inquiries between team members identified the following as the activities most demanding of authoring support: (a) design of adaptation logic (whether rule-based or otherwise); (b) automated dependency checking between adaptation actions (i.e., whether modifying the sequence between actions modifies the end result and how); and, (c) authoring abstractions at the level of hypermedia techniques, in addition to the level of actions. Furthermore, an often cited feature pertaining to rule-based adaptation logic, was the visual building of expressions, with visual support extending to the retrieval of variables from the different modeling components, and, perhaps more importantly, to the automatic representation and encapsulation of the domain ontology.

Finally, a few words are in order in terms of user feedback. The evaluation activities carried out in the context of PALIO did not employ an evaluation framework that can provide direct and explicit results in relation to adaptation (see Unit C, "Adaptive System Evaluation" for an in-depth discussion). However, the structure of the evaluation, as well as the explicit usage issues that it set out to explore, made it possible to indirectly draw some adaptation-related conclusions. These, in summary, were that users (who were not aware of the adaptive facilities of the system): appreciated very much the system's capability to present disability-related information and do so in an accessible way; found that the system's recommendations were very relevant to their interests in most occasions; and, had a very positive attitude towards location-orientation and location-sensitivity in the provided tourist services [PALIO Consortium, 2003]. Although these results do not suffice to judge the comparative merits of the adaptation framework with respect to alternative approaches, they are definitely strong indicators of the fact that useful adaptation behavior can be implemented using it.

Looking forward, it is our goal that the framework spearheads a new perspective in the design and implementation of adaptive hypermedia systems, characterized by:

- The support for XML-based, declarative specification of adaptive behavior, as opposed to more programmatic approaches that are common in the literature.
- The concept of adaptation actions as a common and well-defined "vocabulary" in the specification of adaptive behavior, alongside with the concept of synthesizing higher-level adaptive behavior from simpler building blocks.

- The separation of adaptation logic from the hypermedia elements to which it refers.
- The possibility to easily integrate and interchange alternative adaptation technologies and techniques.

Adoption of (parts of) the framework has the potential to "standardize" portions of the implementation of adaptive systems. Such standardization would result in increased levels of reuse, both at the level of dedicated software components, and at the level of common adaptive behaviors across systems (for example, presentation-level adaptation that transform XHTML output to ensure accessibility by sight-impaired users).

Furthermore, we consider the framework to be a step in the direction of enabling the development of sophisticated authoring tools for adaptation, which will enable non-specialists to design and implement adaptive behaviors. Such authoring tools are currently encountered only in specialized domains (e.g., authoring of adaptive on-line courses, see [Brusilovsky, 2003a]), while they are lacking at more general levels. They are, in our opinion, one of the most important prerequisites for bringing adaptation to the mainstream.

B.2.5 A Comparison between the AVANTI and PALIO Adaptation Frameworks

Having discussed two different adaptation frameworks, one intended for use in AUIs, and one for AHS, we will now briefly look at the most important similarities and differences between the two. For the sake of brevity, and to avoid confusion, the two frameworks will be referred to by the name of the system in which they were employed, i.e., the "AVANTI framework" and the "PALIO framework" respectively.

To start with an obvious similarity, both frameworks employ rule-based adaptation logic. The types of supported rule operators and operands are more extensive in the case of the PALIO framework, however, due to the fact that temporal and spatial reasoning also needed to be supported. In both cases, the decision making component of the framework is capable of identifying the rules that are applicable to a given adaptation constituent – in the case of AVANTI, the style being instantiated or modified, in the case of PALIO the document (fragment) going through the adaptation pipeline.

The AVANTI framework distinguishes between syntactic- and lexical- level rules, primarily due to the different adaptation requirements at these levels – when adapting at the syntactic level, the decision refers to the selection of one or more styles to activate, but when adapting at the lexical level, the decision refers to spatial-, presentational-, and behavioral- attributes of components. The PALIO framework does not distinguish between levels of interaction, but instead introduces a different distinction related to information retrieval.

Specifically, the framework supports "stages" at which adaptations are to be executed, so that it is possible to personalize the information retrieval queries prior to their being issued to the information sources, and, in a subsequent stage, personalize the results of the said queries.

Both decision mechanisms can be triggered by modifications in the dynamic models maintained by the system, and effect adaptations as a result. In the AVANTI framework, the activation of styles taking place as a consequence of this form of triggering needs to take care of "timing" and "scheduling" issues (e.g., dialogs should not be modified while active, notifications should not appear at the same time on the screen, nor in immediate succession to one another, etc.) The PALIO framework deals with a related type of adaptation "scheduling" – for information that is dynamically pushed to the user's device (e.g., as a result of interesting artifacts in the vicinity of the user's changing location), the framework applies constraints to ensure that adaptations are neither obtrusive nor "continuous" in nature, allowing the user to consume the provided information before updating it.

In terms of dynamic models, both frameworks maintain a local, vector-based user model. This model is in neither case a full-blown user modeling service, and both frameworks provide explicit support for delegating user modeling tasks to such a backing service / component. Nevertheless, basic functionalities, such as forward chaining of modifications / updates, are supported by both frameworks. The PALIO framework also maintains a per-user interaction context model, and thus has an enriched space of adaptation determinants (covering device and connectivity characteristics, temporal abstractions, the user's geographic location, as well as historical elements of the context itself).

An area where the frameworks largely diverge is that of adaptation actions. The AVANTI framework was explicitly designed to support the development of UUIs. As a result, the framework's adaptation constituents include interaction tasks and styles, and the lexical level characteristics of interaction components. The PALIO framework on the other hand was designed with generic AH support as a goal. Therefore, the framework's adaptation constituents are the elements and structure of hypermedia documents.

A further implication of the AVANTI framework's close ties to UUID, is that aspects of the syntactic-level adaptation rules can be validated automatically. Specifically, using the task decomposition specifications, it is possible to identify through static, pre-runtime analysis whether there exist legitimate model states that lead to inconsistent or conflicting combinations of styles. Such types of validations are not possible in the PALIO framework. The only kind of validation possible in this case would be to identify rules that result in "co-modification" of elements or attributes of a hypermedia document (fragment); determining whether these are indeed problematic, though, would be the task of a human operator, as it is not possible to decide this automatically without a significant amount of semantic information being available for each of these elements and attributes.

In terms of efficiency, the PALIO framework is considerably more optimized than the AVANTI one. This is of course necessary if one considers that the PALIO framework, being a server-side one, needs to support multiple concurrent users, and do so in a tight processing cycle so as not to impose a forbiddingly heavy burden on the normal processing that the underlying information system performs.

In conclusion, and keeping in mind the important differences between the frameworks, one can argue that the PALIO one is the more mature of the two, which should not come as a surprise since it has benefited from the experience gained from design and implementation of its AVANTI predecessor.

B.2.6 Reported Work against the State-of-the-Art

The progress of the area of AHS in the years since the development of the PALIO framework has been explosive. This has been due both to the increasing popularity of the web, and the growing interest in supporting personalization in practically all application domains of computer-supported human activity. A systematic overview of progress in the field in the last decade can be found in [Knutov, De Bra, & Pechenizkiy, 2009], with more extensive treatment of individual topics available in [Brusilovsky, Kobsa & Nejdl, 2007]. In the rest of this section we will focus selectively on topics specifically related to the PALIO framework.

Perhaps the most well known AHS framework in the literature that is still in active development, and also available in open source form, is AHA! (most recently described in [De Bra, Smits & Stash, 2006]). The AHA! framework has evolved significantly since its inception, and recent (some, as of yet, unpublished[27]) additions have introduced features that are on par with those of the PALIO framework (e.g., forward chaining of rules, a more extensive set of operand types and operations that can be used in rule expressions). Compared to AHA! and other systems in the literature, the PALIO framework retains, today, several of its advantages, the most important among which include:

- It has been specifically designed to be easy to combine with other web-publishing frameworks, providing adaptation functionality "on top" of them (but see below for limitations in this respect), making the adaptation infrastructure more easily utilizable in new systems.

[27] Some of the recent additions to AHA! are known to the author as they were made within the ALS project, in which both the FIM Institute and TU/e were partners. Please refer to the section "A.2.2 Work Context and Research Projects" on page 32 for additional information.

- It offers support for creating high-level declarative adaptation rules, separated entirely from the documents / fragments on which these are applied. To achieve the same level of functionality with AHA!, for example, one would have to use the later in combination with the MOT authoring tool [Cristea & de Mooij, 2003], which is capable of transforming high-level rules to a format compatible with AHA!.
- It features an explicit representation of basic adaptation actions that can be synthesized into most of the adaptation techniques in the literature (with the exception of some of the multimedia-oriented techniques, and ones involving Natural Language Generation).

The shortcomings listed in section "B.2.4 Discussion", however, are also still relevant, and some of them, such as the lack of authoring tools, are magnified by the growing attention to facilitating the authoring of adaptation. Further to these shortcomings, here we will focus on additional areas of work in the PALIO framework that would have benefited from recent progress.

One such area is the use of ontologies. Although the framework did support the ontological representation and querying of domain data, the adopted approach was: (a) custom, and (b) not extended to the rest of the framework's modeling needs. In the time since the development of the PALIO framework a significant amount of standardization activity has taken place in the area of semantically meaningful representations of data, spearheaded by the World Wide Web Consortium under the umbrella of the "Semantic Web" [Berners-Lee, Hendler & Lassila, 2001]. Specifications such as the Resource Description Framework (RDF)[28], the SPARQL Protocol and RDF Query Language (SPARQL)[29], and the OWL Web Ontology Language (OWL)[30] have provided the means for standardizing approaches to, and implementations of, representation and reasoning over semantic data. As one might expect, this has had a substantial effect in the area of AHS, with the aforementioned technologies being actively used for user modeling, domain modeling, model-based inferencing, etc. (see, e.g., [Henze, Dolog & Nejdl, 2004], [Heckmann et al., 2005], [Aroyo et al., 2007]). Furthermore, recent efforts have specifically addressed the challenge of interoperability of ontological data in adaptation models, to allow for their sharing and reuse [Balík & Jelínek, 2008]. An in-depth overview of the use of Semantic Web technologies in closed- and open- corpus AHS can be found in [Dolog & Nejdl, 2007], and several articles on recent developments in [Angelides, Mylonas & Wallace, 2009]. Against this background, if one were to revisit the PALIO framework today, the custom modeling solutions employed would

28 http://www.w3.org/RDF/

29 http://www.w3.org/TR/rdf-sparql-query/ (query language), http://www.w3.org/TR/rdf-sparql-protocol/ (protocol)

30 http://www.w3.org/TR/owl-ref/

likely be replaced with standards-based ones, with a view to interoperability and extended generality of the framework.

Recent trends have also put into question one of the basic premises of the PALIO framework, namely its orientation towards adaptation in the context of a XML-based document "pipeline". These days, a lot of hypermedia systems (and especially web-based ones) are built using component-based presentation / interaction frameworks, with hypermedia documents being generated directly from structured representations of data (possibly using approaches such as the ones described in the previous paragraph). This makes it difficult to apply "first stage" adaptations using the approach employed in the framework, which assumes that the retrieval and composition of information and document fragments is also expressed in XML – and can thus be adapted using the same techniques used for the retrieved or assembled fragments. To cater for this shortcoming in a generic way, the framework would need to be updated to support adaptation actions expressed in arbitrary, domain-specific languages. An update would also be necessary for the mechanism that assembles the adapted fragments into a document; a possible approach here would be to create specialized versions of the mechanism for the target web application development frameworks.

The PALIO framework also does not provide explicit support for newer paradigms of client-server communication on the web, such as the use of Asynchronous JavaScript and XML (AJAX)[31]. This would be, in principle, an easier issue to address, since these paradigms necessarily still follow the request-response cycle for the communication between the client and the server. Consequently, the only modifications that would be potentially necessary in this direction would be: (a) transformation of request data from the incoming formats to XML, and (b) transformation of response data from XML to the required output formats. Note that these additions would only be necessary is the incoming or outgoing data need to be expressed in "non-standard" formats, such as Javascript Object Notation (JSON)[32].

A distinctive trend in recent AHS research is adaptation in open corpus hypermedia [Brusilovsky & Henze, 2007]. Work in this direction aims to extend AHS with the possibility to operate on an open corpus of documents, which is not known at design time and in addition to this can be constantly changing and expanding. Although the PALIO framework can certainly be applied in open corpora, it suffers from the same shortcomings as other frameworks in its category: it provides neither mechanisms for concept and content alignment, nor mechanisms for accessing and manipulating an expanding domain model.

A different case of adaptation "in the wild" (i.e., on arbitrary documents), where the PALIO framework would be more readily applicable is exemplified

[31] See http://en.wikipedia.org/wiki/Ajax_(programming)

[32] See http://en.wikipedia.org/wiki/Json

by the "Glue" system[33]. The later comprises a plug-in for the Firefox browser, and a corresponding service, which, combined, augment web pages that the user visits with links to information related to the page's content, and of potential interest to the user. According to the system's documentation, part of these recommendations are of a social nature (based, e.g., on the browsing history, or past ratings of the user's social network), and part are based on analysis of the page contents and their association with a number of categorical information sources, on the basis of the user's profile. In other words, the second type of recommendation consists in identifying "entities" and their type in a page (e.g., a musician, a restaurant, a book), associating these entities with their semantic counterparts in the system's domain ontology, consulting the user model to determine whether the said entities are of interest to the user, and, if so, augmenting the entities with links to additional related information and pages. With additions of the type discussed earlier with respect to the introduction of support for AJAX, the PALIO framework could be easily tailored to play the role of the server-side component of the described system.

In summarizing the preceding brief overview, one could argue that the PALIO framework is still relevant in today's AHS landscape, but would need to be updated to take advantage of, and exhibit better compatibility with, current technologies and approaches.

---

33 http://www.getglue.com/

U n i t C

ADAPTIVE SYSTEM EVALUATION

Towards an Evaluation Framework for Adaptive Systems

The work presented in this Unit is concerned with one of the critical issues in the development of adaptable and adaptive user interfaces, namely the lack of appropriate evaluation methods and techniques. Approaches that existed prior to the publication of the work presented herein could not be used to assess the way and extent to which the adaptation facilities of the interface affect interaction qualities such as accessibility and usability. Publication of the corresponding articles, in particular the one on which the second Chapter is based, has been instrumental in establishing an evaluation framework specifically intended for the formative and summative assessment of adaptive systems.

The first Chapter, "C.1 Evaluating Adaptable and Adaptive User Interfaces: Lessons Learned from the Development of the AVANTI Web Browser", represents the first steps towards an evaluation framework and reports on the practices developed and employed for the evaluation of adaptivity in the AVANTI web browser. The practical experience gained, gave rise to a number of requirements that guided the development of generic methods and techniques for evaluating adaptation-capable systems.

The second stage of the work, presented in the second Chapter "C.2 A Modular Approach to the Evaluation of Adaptive User Interfaces" introduced the idea of treating adaptivity as a multi-dimensional process, and proposed a new, modular approach to the evaluation of adaptive systems, which is specifically intended to cater for the principled assessment of adaptation-related and adaptation-oriented elements of interaction.

The approach presented in the second Chapter has been combined with the one proposed in [Weibelzahl, 2001], and which follows the same principles, to create a framework for the "layered" evaluation of adaptive systems. This framework, introduced in [Paramythis & Weibelzahl, 2005] represents the state-of-the-art in the area.

C.1 Evaluating Adaptable and Adaptive User Interfaces: Lessons Learned from the Development of the AVANTI Web Browser[34]

As already discussed in previous chapters (see "B.1 The AVANTI Adaptive Web Browser", page 39), the design and development of the AVANTI web browser followed the Unified User Interface Development approach [Stephanidis, Savidis & Akoumianakis, 1997], which led to the construction of a unified browser interface, capable of adapting itself to suit the requirements of different user categories: able-bodied, people with light or severe motor disabilities, and blind people [Stephanidis, Paramythis et al., 1998a].

As a reminder, Unified User Interfaces employ adaptation techniques to automatically tailor themselves to different sets of user and usage context characteristics [Stephanidis, Savidis & Akoumianakis, 1997]. Specifically, the design phase involves the construction of a *polymorphic task hierarchy*, within which different tasks may have alternative instantiations in the user interface (called *instantiation styles*, or simply *styles*). The task decomposition thus proceeds in a polymorphic fashion, defining alternative styles and task hierarchies, according to requirements and preferences of different user categories. In other words, different styles define alternative ways in which a specific task can be realized.

Following Unified User Interface Design, the resulting single design artifact may have multiple instantiations during initiation of interaction (adaptability), in order to ensure accessibility for a wide range of users. Moreover, each interface instance is continuously enhanced at run-time (adaptivity), in order to provide high-quality of interaction to all potential users. As already discussed, in AVANTI, both adaptability and adaptivity were supported through a rule-based framework, which necessitated the transformation of the design rationale as this was captured in the polymorphic task hierarchy, into corresponding adaptation rules.

One of the key problems in the development of self-adapting user interfaces is the inadequacy of traditional evaluation methods and techniques to be used for the evaluation of adaptable and adaptive interfaces. Specifically, existing evaluation methods are appropriate for assessing "static" user interfaces, but not the way and extent to which the dynamic adaptation facilities of the interface affect interaction qualities, such as accessibility, usability, acceptability, etc. Although there have been several attempts in the past to

---

[34] This chapter is based on [Stephanidis, Paramythis & Sfyrakis, 1999]. The R&D work described here has been carried out while the author was employed by the HCI-Lab of ICS-FORTH (Heraklion, Greece), and in the context of the ACTS AC042 AVANTI project (please refer to section "A.2.2 Work Context and Research Projects" on page 32 for additional information).

construct both objective and subjective expert- and user-based evaluation methods in the area of interface adaptation (e.g., [Totterdell & Boyle, 1990; Grüniger & Van Treeck, 1993; Höök, 1997]), the lack of understanding of the dynamic dimensions of adaptive user interfaces (as well as of the differences introduced by alternative approaches to achieving and "driving" adaptive behavior), compromises the applicability of solutions that have been suggested to date. The main deficiency of most of the aforementioned approaches is that they fail to identify the adaptation-oriented characteristics of the user interface which have detrimental, or, adversely, beneficial effects on interaction.

Due to these shortcomings, the approach taken in the evaluation of the adaptable and adaptive user interface of the AVANTI web browser has been the introduction of a two-fold assessment process, which involved:

(a) Iterative, expert-based assessment cycles in the design of appropriate interaction styles, the definition of adaptation rules, and the development of the decision mechanism for materializing the required adaptation behavior; expert-based assessment has been intended to compensate for the lack of appropriate evaluation techniques for adaptation-capable user interfaces and the lack of empirical evidence upon which to base the design of adaptations.

(b) End-user based evaluation activities (using questionnaires, observations and interviews), intended to assess the overall usability and accessibility of the user interface.

C.1.1 Expert Evaluation

Expert evaluation activities within the development of the AVANTI web browser aimed to employ accumulated knowledge and experience in the areas of user interface design, usability, and assistive technology, for: (a) the design of alternative interaction styles that cater for the different user and usage-context requirements, as well as (b) the design of appropriate adaptation behavior to be built in the resulting interface.

C.1.1.1 Evaluating the Design of Interaction Styles

Early evaluation activities were intended to assess the appropriateness of the designed interaction styles for the specific interaction context and the particular user characteristics for which they were intended. Particular emphasis was placed in the evaluation of the accessibility features provided by the designed interaction patterns to the target disabled user categories (i.e., blind and motor-impaired).

The experts reviewed each interaction style separately, based on established accessibility and usability guidelines and heuristics (e.g., [Vanderheiden &

Kaine-Krolak, 1995; ISO 9241-11, 1997; Story, 1998]), and were asked to identify potential accessibility, usability, or other problems of each style, as well as to propose possible improvements in the design, based on their experience. The outcome of these inspection activities was collected and analyzed, and the results were fed back into the design process, where they have led to three types of intervention to designed interaction styles: (a) re-design of styles, based on identified problems, or on contributed ideas towards their enhancement; (b) elimination of redundant (due to the similarity in the characteristics of the end-users or the usage contexts they were intended to cater for) interaction styles; (c) introduction of new interaction styles, to cover user characteristics and contexts of use that were not addressed adequately by existing styles.

C.1.1.2 Validating the Adaptation Rules

The development of the adaptation rules took place in two steps. Firstly, the rules were defined by a group of experts, through several iterations following each task-decomposition phase, as well as each stage in the definition / selection of alternative interaction styles. Secondly, a process was defined, to assess the design of adaptations by validating the resulting adaptation rules. A detailed description of the adaptations rules is available in [Stephanidis et al., 1997], while a brief overview can be found in Chapter "B.1 The AVANTI Adaptive Web Browser".

The validation of the adaptation rules has itself taken place in three consecutive phases: evaluation of the rules by experts; verification of the adaptation mechanism on a per-rule basis; and, verification of the adaptation mechanism across sets of rules. The results of the experts' assessment phase have led to four types of intervention to adaptation rules: (a) elimination of rules that were deemed inappropriate, or not sufficiently supported; (b) introduction of new rules (based on the recommendations of the experts); (c) modification of the rules' triggering conditions (e.g., adding, or removing a particular user characteristic from the description of the triggering situation); (d) modification in the rules' decisions (e.g., addressing a particular situation only through guidance, instead of through guidance and extensive interim feedback).

The validation of adaptation rules was followed by the verification of the adaptation mechanism on a per-rule basis, and verification of the system's adaptation behavior across sets of rules. The mechanism used for valuating rules and carrying out the respective adaptation decisions was tested for consistent behavior, by: (i) testing that the triggering conditions for each individual rule led to the desired (adaptation) behavior on the part of the user interface; (ii) testing sets of rules in combination, to assess the degree to which they affect each other from a functional, as well as from the user's point of view. The former procedure (i.e., testing rules individually) was performed by examining the defined rules one-by-one and verifying system behavior, when the activation parameters were set, or changed. A "wizard of oz" technique was used to simulate the functionality of the User Modeling Server[35]. The later procedure (i.e., testing combinations of rules), was performed through the development of representative scenaria, where multiple activation parameters were set or changed simultaneously.

The verification procedure resulted in the identification of conflicts in the activation of specific styles and inappropriate activation of certain rules in specific tasks. The main problem arose from the redundant activation of styles under certain conditions. The outcomes of the validation procedure initiated specific modifications in the pre-defined rules, as well as the adaptation mechanism itself.

C.1.2 User-based Evaluation

Formal usability evaluation studies of the AVANTI web browser have been carried out in the context of the experimental and field evaluations of the AVANTI system in the three user trial sites in Kuusamo, Siena and Rome [Andreadis et. al., 1998]. These experiments evaluated the overall usability of the AVANTI information systems following a common evaluation framework. The trials were performed on distributed heterogeneous network environments supporting different access points, including: public information kiosks, home computers, travel agency offices, and laboratory sites. The subjects that took part in the experiments included citizens and tourists at the trial sites, as well as travel agency staff (in the case of the Siena information system). In terms of physical abilities, subjects were drawn from all three categories supported by the project, i.e., able-bodied, blind and motor-impaired. In total, there were 175 subjects in all three experiments, exposed to more than one instance of the user interface, sometimes through iterative evaluation sessions.

The usability goals set up by the common evaluation framework, and assessed during the experiments were: learnability, efficiency of use, memorability, errors, satisfaction, user attitude, adaptability and adaptivity. The experiments adopted a task-based evaluation approach, utilizing both qualitative and quantitative evaluation methods. The tools which were utilized include observations and interviews (qualitative) as well as attitude scale questionnaires and log-files (quantitative). The functionalities of the AVANTI web Browser as well as the supported adaptability and adaptivity features were addressed in the observation sessions, the interviews and the subjective evaluation (attitude scale) questionnaires.

---

35 For this purpose, a software module that simulates the functionality of a user model server has been developed and was used for "manually" generating the dynamic user situations.

The results of the evaluation were encouraging, both in terms of user acceptance of the characteristics of the interface, and in terms of the fulfillment of the initial goals that led to the employment of adaptation techniques in the user interface. In particular, adaptability addressing accessibility issues for the various end-user groups proved quite successful, as each user category conceived the interface as having been specifically developed to cater for their particular requirements. The results were similar for the non-disability related categories in which users were classified (e.g., according to their computer expertise).

Adaptivity was assessed to a lesser degree than adaptability during the evaluation, mainly due to the following reasons: (a) adaptivity requires that interactive sessions are rather lengthy, so that adequate information about the user and the context of use is collected, before any practically useful inferences can be made, and (b) existing user interface evaluation techniques do not offer themselves for the evaluation of dynamically changing, non-deterministic (from a user's perspective) systems. As far as the first of the above issues is concerned, the typical duration of the interaction sessions performed during the experimental activities was not adequate for the extraction of dependable inferences that dynamic adaptation could be based on. As a result, users were aware of only a minimal set of adaptive features in the interface; however, their reaction to the features they did observe was positive. As far as the availability of empirical methods and techniques for the evaluation of adaptivity is concerned, it has already been argued that existing knowledge in the area of user interface evaluation is inadequate for the derivation of appropriate techniques and instruments to measure the effects of adaptive system behavior on interaction.

C.1.3 Towards an Evaluation Framework for Adaptation

Evaluation of adaptation-capable user interfaces should aim to identify those aspects of the interface that have beneficial / detrimental effects on the accessibility and interaction quality offered by the interface for different categories of users and in different contexts of use. Two coarse evaluation dimensions can be derived from the above goal. The first concerns the appropriateness of the different instantiation styles for the purpose they were developed. This entails the assessment of the styles themselves as individual interactive artifacts and as components of the overall interface, as well as the assessment of the design rationale / decision logic that activates (or deactivates) these styles, based on user and usage characteristics. The second dimension concerns the evaluation of the dynamic adaptation (adaptivity) in the interface. This is in fact the most difficult part to evaluate, as there are multiple factors that determine the various qualities of the interface. For example, an adaptation may be conceived as entirely dissatisfactory by the user if: (a) the adaptation logic itself is flawed; (b) the "triggers" of the adaptation were wrongly inferred by the user modeling component; (c) the adaptation was not "timely" (e.g., it came "too late" from the user's perspective); or (d) the adaptation policy is not satisfactory (e.g., because the user is not given enough control over it).

To counterbalance the inherent difficulties in evaluating dynamic adaptation in the interface, evaluators should plan the evaluation process carefully from the early design phases, and should actually base the evaluation plan on the overall design process. Thus, evaluation should not be restricted to summative activities; rather, it should proceed in parallel to the design of the user interface and should strive to identify deficiencies and possible problems as early as possible, informing and guiding the development process.

The evaluation activities of the AVANTI web browser can be considered as preliminary steps towards generic methods and techniques for the evaluation of adaptation-capable user interfaces. However, a lot more research and practical experience are required in this direction, before we can derive valuable results that will be reusable across application domains, user categories and contexts of use. The next chapter presents work in this direction, and specifically towards a framework specifically intended for the evaluation of adaptive systems, aiming to provide guidance for both formative and summative evaluation activities in this area.

C.2 A Modular Approach to the Evaluation of Adaptive User Interfaces[36]

The challenge of developing adaptive systems for various application domains, and on different interaction platforms, has been addressed extensively in the literature, from an engineering perspective. The same, however, is not true from the perspective of designing and evaluating adaptive systems. To date there is limited knowledge on which adaptation methods and techniques are appropriate for different users and for different interaction contexts. This, in turn, is due to the lack of reusable empirical findings coming from the evaluation of adaptive interactive systems.

This lack of empirical findings can be traced back to the way in which the evaluation of adaptive user interfaces is approached today. To start with, adaptation is not sufficiently addressed by existing standardized evaluation frameworks (although, in some cases, it is a concern) [Stary & Totter, 1997]. As a result, researchers have had to employ more "basic" evaluation tools to approach the assessment of adaptive systems. In doing so, one inevitably comes across the second stumbling block: separating adaptation from the rest of the factors that may influence interaction, so as to assess its relevant merits in isolation.

The latter gave rise to the very popular "with and without" adaptivity evaluation design, in which an adaptive instance of the system is compared with a non-adaptive one. This evaluation design has been used in several studies in the field, including, for example, [Kaplan, Fenwick & Chen, 1993; Meyer, 1994; Boyle & Encarnacion, 1994; Weber & Specht, 1997; Brusilovsky & Pesin, 1998; Brusilovsky & Eklund, 1998]. A major criticism of this evaluation approach has been that the non-adaptive instance cannot be "optimal" in any way, if adaptation is properly "designed into" the system [Höök, 2000]. Another equally important problem is that, in this type of study, the reasons behind the "success", or "failure" of adaptation can only be traced back to the initial hypotheses of the adaptive system design. In other words, it is not possible to ascertain why, and under what conditions, a particular type of adaptation may be employed towards a specific goal. This situation is exemplified in the several studies that have addressed adaptive link annotation, often arriving at contradictory conclusions [Eklund & Brusilovsky, 1998].

A different perspective on the study of adaptive systems has been put forward by Oppermann [Oppermann, 1994], in the assessment of adaptation in

---

[36] This chapter is based on [Paramythis, Totter & Stephanidis, 2001]. The R&D work described here has been carried out while the author was employed by the HCI-Lab of ICS-FORTH (Heraklion, Greece). The overview of evaluation methods in section C.2.1 has been primarily authored by Alexandra Totter, but the section was not redacted for completeness.

Flexcel II [Krogsæter, Oppermann & Thomas, 1994]. Following this perspective, adaptation is treated as an integral part of the system and evaluation is not based on the presence of a non-adaptive counterpart. However, this approach was also limited with respect to assessing the degree to which the different factors influencing adaptation had contributed to its "success" or "failure".

In light of the above, there is an acknowledged need for a renewed look at the evaluation of adaptation, in which the employment of traditional HCI evaluation methods and techniques will be placed on a new basis, acknowledging the particular characteristics that differentiate adaptive systems from their "static" counterparts [Höök & Svensson, 1999; Höök, 2000].

The main idea behind the approach put forward here is that the evaluation of adaptive systems, should not treat adaptation as a "monolithic" / singular process happening behind the scenes; rather, adaptation should be "broken down" into its constituents, and each of these constituents should be evaluated separately where necessary and feasible. The seeds of this idea can be traced back to [Totterdell & Boyle, 1990], who propose that a number of adaptation metrics be related to different components of a logical model of adaptive user interfaces, to provide what amounts to adaptation-oriented design feedback. Furthermore, [Totterdell & Boyle, 1990] present two types of assessment performed to validate what is termed "success of the user model" (note that, in their case, the "user model" is also responsible for adaptation decision making):

> "Two types of assessment were made of the user model: an assessment of the accuracy of the model's inferences about user difficulties; and an assessment of the effectiveness of the changes made at the interface."

The contribution made herein, along these lines, is the introduction of a modular approach, which offers a detailed view into the "decomposability" of adaptation, from the perspective of HCI-oriented evaluation. The main strength of this approach, which builds extensively on previous work in the field, lies with the potential it offers towards deriving detailed evaluation results that can be analyzed, extended and reused across user interfaces and application domains.

The proposed approach is presented in this chapter in relation to a particular class of adaptive systems, namely AUIs. Although there exist numerous definitions and classifications of adaptive user interfaces (see, e.g., [Totterdell, 1990; Totterdell & Rautenbach, 1990; Dieterich et al., 1993]; and section "A.1.2 Definitions, Models and Taxonomies of Adaptation" on page 6), at this point of the discussion is suffices to employ a minimal definition, whereby *an adaptive user interface is characterized by its ability to detect and perceive characteristics of its environment, and modify itself accordingly*. Furthermore, the discussion will, in most cases, constrain the environment to the user; in other words, we are interested in user-adaptive user interfaces, which automatically tailor interaction to fit the individual aptitudes, skills, knowledge, etc., of their users. Nevertheless, it should be noted that the material presented is not exclusively relevant to AUIs; rather, the proposed approach is expected to be easily extensible to other classes / categories of adaptive systems.

A related approach to the one presented here can be found in [Weibelzahl & Lauer, 2001], where the authors introduce an evaluation framework for the assessment of interactive systems that employ Case-Based Reasoning techniques to support adaptation in their interaction with the user. Their framework bears many similarities to the approach postulated herein, especially in terms of how adaptation is "decomposed" and evaluated in a series of steps.

The rest of this chapter is structured as follows. The next section presents a contextual perspective on the evaluation of interactive software systems in general, and the evaluation of AUIs in particular. It also sets the basis for subsequent discussions of different evaluation methods and their suitability for evaluating specific aspects of the AUI. The following section presents the proposed evaluation approach in two steps: in the first step, a high-level model for adaptation in AUIs is introduced; in a second step, the model is broken into (sometimes overlapping) modules and the evaluation of each module is discussed in detail. The final section compares the work presented here with the State-of-the-Art in the area.

C.2.1 A Contextual Perspective on Evaluation

[Karat, 1997] defines evaluation as "the result of a process with a purpose in a context focused on an object". This definition is based on the following common features of evaluations [Karat, 1997]: *objects* being evaluated; *process* through which one or more attributes (of the objects) are judged or given a value; and, *purpose* of the evaluation. Additionally, the "grand total" of the setting within which evaluations take place (and within which the objects evaluated are encountered) constitutes the *context* of the evaluation.

When evaluating a software system one has to select from various processes (or methods) that might be employed for the purposes of judging how that system might be used to accomplish goals in a certain environment (context). The selection of an evaluation method is not a context-free activity. In the field of HCI there exists today a large variety of techniques available, which serve different evaluation purposes (see, e.g., [Karat, 1997; Wixon & Wilson, 1997; Nielsen, 1994]). As a result, to select appropriate methods for the evaluation of AUIs, one has to relate their features and strengths with the primary characteristics and requirements of AUI evaluation.

AUIs are traditionally indented to facilitate and enhance user interaction with the underlying domain; it follows that the primary "goals" of an AUI are to achieve higher levels of user satisfaction (with respect to the application domain, but also the interaction itself), while ensuring user acceptance (a much debated issue in the area of AUIs). Based on the above, methods and techniques already in use for the assessment of user interfaces against well-established evaluation criteria (with usability [Nielsen, 1994] being the most prominent among them), constitute a promising starting point for deriving empirical approaches to the evaluation of AUIs.

[Jordan, 1998] identifies thirteen empirical evaluation methods[37] commonly used for the investigation of usability in HCI (see Table 5, first column). To support the selection of an appropriate evaluation method, several classification schemes have been developed. For example, [Wixon & Wilson, 1997] classify usability evaluation methods along a number of different dimensions, including: formative vs. summative methods; discovery vs. decision methods; formalized vs. informal methods; user involvement vs. user exclusion; component evaluation vs. complete evaluation; etc. (for alternative classifications of evaluation methods, refer to, e.g., [Karat, 1997]). In the following, we will introduce a preliminary classification of these evaluation methods, based on two dimensions, which will assist in the selection of those best suited for the evaluation of (aspects of) AUIs: (a) the types of evaluation measures that are supported by each method, and (b) the stage of the development life cycle that each method is best suited for.

Regarding the first dimension, i.e., evaluation measures, McGrath [McGrath, 1995] identifies three main types of measures[38], used extensively in the social and behavioral sciences (see Table 5, top row):

- *Self reports* of participants, which are always done under conditions in which the respondents know that their behavior is being recorded for research purposes (e.g., questionnaire responses, interview protocols, rating scales, etc.)
- *Observations*[39]: This term refers to records of behavior made directly by the investigator, or made by someone substituting for the investigator, or made by some physical instrument that is serving the investigator (an automatic electronic counter, stopwatch, etc.)
- *Trace measures*: These are records of behavior laid down by the behavior itself, but without the actors being aware that they are making such a record.

---

37 Please note that the term "evaluation method" is used in [Jordan, 1998] to refer collectively to (combinations of) research strategies (e.g., laboratory experiment, field experiment, sample survey) and data collection methods (e.g., questionnaires). Although this view may be considered rather restrictive (a more comprehensive -albeit more complex- approach would allow the two to vary independently), it suffices for the analysis requirements of this paper.

38 [McGrath, 1995] identifies an additional type of measures, namely *archival records* (further distinguishing between *public* and *private* ones), which, however, is not relevant to this discussion.

39 A further distinction made by [McGrath, 1995], between *visible* or *invisible* observers, is not directly relevant to our discussion.

Table 5: Classification of empirical usability evaluation methods.

Empirical Usability Evaluation Methods \ Types of Measures	self reports	observations	trace measures	Suitability for employment at different development stages
Focus groups	●			at any stage of the design process
Interviews	●			
Questionnaires	●			
Private camera conversation	●			better suited for early design stages
Valuation methods	●			
User workshops	●			
Co-discovery	●	○*		require at least an interactive prototype
Think aloud protocols	●	○*		
Logging use			●	
Controlled experiments		●		
Incident diaries	●			better suited for "finished" products
Feature checklist	●			
Field observation		●		

* When applied in HCI, Co-discovery and Think aloud protocols need to be combined with some form of observation in order to obtain a meaningful record of the interaction circumstances, so as to enable the contextual interpretation of the users' comments.

Regarding the second dimension of our classification, i.e., stage of the development life cycle that each method is best suited for, a broad categorization is employed, which distinguishes between methods that: (a) are best suited for the first (exploratory) stages of design, (b) require the existence of at least an interactive prototype, (c) are targeted towards complete ("finished") products, and (d) can be used (in variations) at any stage of development.

The rationale for classifying the methods along the aforementioned dimensions is directly related to the requirements of the proposed evaluation approach. To start with, the proposed approach is explicitly intended to facilitate the derivation of empirical data that will be fed back into the design of adaptation, as well as reused across application (domain) boundaries. To this end, the employment of both *formative* and *summative* evaluation is postulated (although at different stages in the development cycle), where, according to [Karat, 1997]:

- formative refers to evaluation used as a means of gathering information to inform iterative design, at any stage of the development cycle, and is focused on improving design; while,
- summative refers to (mainly experimental) evaluation designs focusing on hypothesis testing and statistical analysis, typically conducted late in the development cycle, with the aim of measuring how "good" or "bad" something is.

Furthermore, the type of measure that each method supports is vital, on the one hand, in deciding whether a method is appropriate for measuring something, and, on the other hand, on making educated decisions as to how to combine methods to derive the desired results.

In addition to the above evaluation methods, the proposed AUI evaluation approach will also consider expert-based ones (i.e., evaluation methods that require the participation of experts, but not of end users). Following the broad categorization of [Jordan, 1998], such methods can be classified as *expert appraisals* and *cognitive walkthroughs*. Although, the two categories are similar in that they require an expert to evaluate a product on behalf of the user, they differ in their perspective on the evaluation [Jordan, 1998]: expert appraisals typically require the expert to judge the product against known principles, guidelines, rules, standards, etc.; cognitive walkthroughs, on the other hand, call upon the expert to approach the evaluation from the point of view of a typical user performing a particular task.

Finally, it should be mentioned that, following the norm in HCI, user testing in a "usability laboratory" (for hypothesis testing, or performance measurements such as error rate, task completion time, task frequency, etc.) is classified under controlled experiments (although controlled experiments are not restricted to this type of user testing).

C.2.2 Modular Evaluation of AUIs

As mentioned earlier, the proposed approach is based on the premise that the evaluation of individual stages (referred to as "modules") involved in the AUI adaptation cycles, enables the derivation of detailed findings, which, in turn, provide ample feedback back into the AUI design process.

Following the definition of evaluation introduced in the previous section, an evaluation approach is proposed which:

- identifies "modules" of AUIs that can, and should, be evaluated both separately and in combination (i.e., the evaluation objects);
- presents the evaluation rationale underlying the decomposition of AUIs into modules and the subsequent individual assessment of these modules, based on specific criteria (i.e., the evaluation purpose);

- circumscribes the methods and techniques that can be employed for the evaluation of the different "modules", in the different stages of the AUI development life-cycle (i.e., the evaluation process).

To that effect, the rest of this section will: (a) present a high-level model for adaptation in AUIs; (b) identify the individual stages of adaptation that can be targeted as evaluation modules; and, (c) propose specific evaluation methods and techniques that can be employed for each module.

C.2.2.1 A High-level Model of Adaptation in AUIs

Although relatively young, the field of AUIs is abundant with conceptual, architectural, and functional models of adaptation, spanning a large range of platforms, component technologies, theoretical approaches to adaptation, and types of adaptation supported. This pluralism is further compounded by the existence of highly relevant models in the related field of Intelligent User Interfaces. In the context of the proposed approach, a base model for adaptation is required, which will reveal some important high-level architectural components of AUIs, as well as explicitly represent the fundamental stages involved in deciding upon and effecting adaptation in HCI.

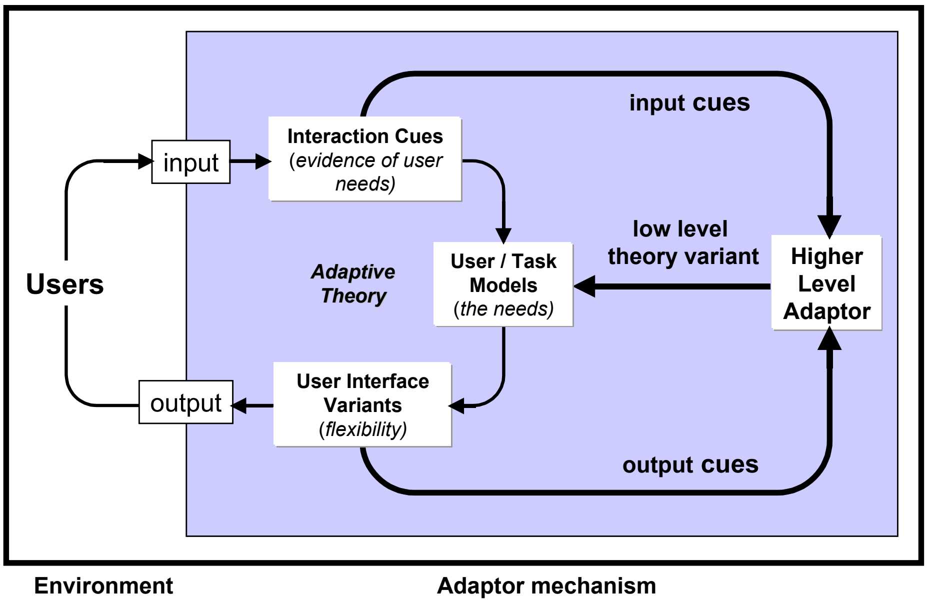


Figure 30: Logical two-level architecture of adaptation, adapted from [Totterdell & Rautenbach, 1990]

The model presented in Figure 31 is based on, and extends the logical two-level architecture of adaptation in [Totterdell & Rautenbach, 1990], presented in Figure 30. Our goals in deriving this model have been to: (a) make the individual stages of adaptation as concrete as possible, without, however, delving into technical issues, or implementation-oriented details, and (b) introduce details related to different approaches to AUI adaptation, which impact on the evaluation choices (affecting both the *objects* of evaluation, as well as the *process* for evaluating them). A number of points that should be noted regarding the model are: no assumptions are made as to the employed technologies and the targeted platforms; no assumptions are made as to the physical distribution of user interface components (e.g., over the network); although depicted separately at the conceptual level, some of the components may actually be combined in an implemented AUI.

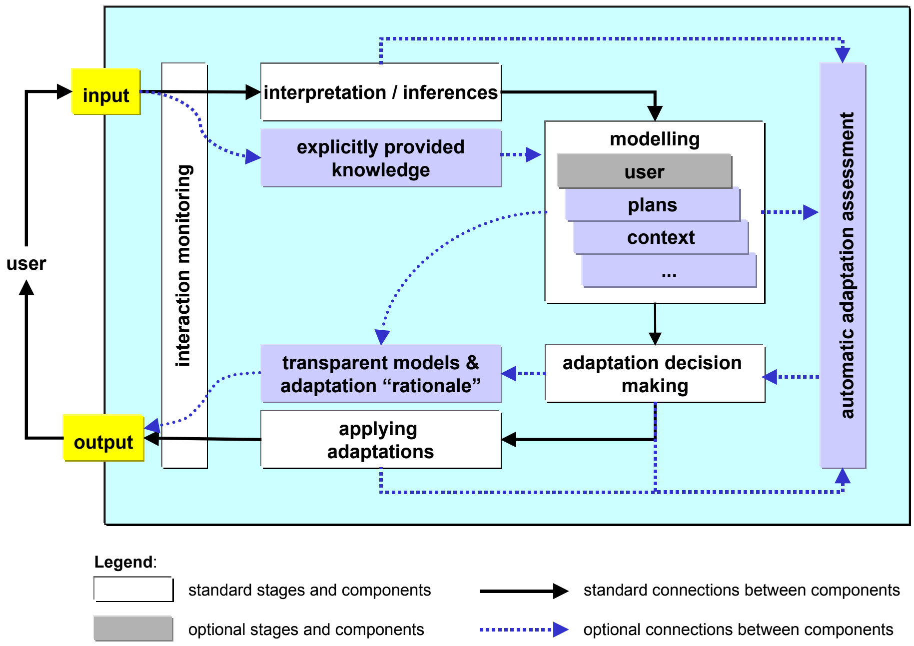


Figure 31: High-level model of adaptation in AUIs.

The model encompasses the following components / stages of adaptation:

- *Interaction monitoring*: Refers to facilities that are intended to capture the exchanges between the user and the user interface, at different levels of the interaction (i.e., physical, syntactic, semantic occurs [Hoppe, Tauber & Ziegler, 1986]).
- *Interpretation / inferences*: Refers to the part (or parts) of the AUI that is responsible for interpreting information made available through interaction monitoring, in order to update the models maintained by the system (e.g., user model).

- *Explicitly provided knowledge*: Refers to information about the users' characteristics, plans, tasks, etc., which is explicitly provided to the system (as opposed to indirectly inferred from interaction data), typically by users themselves.
- *Modeling*: Refers to explicit or implicit representations of the users (including, for example, their abilities, skills, requirements, preferences), their plans with respect to a particular (portion of an) interactive session, the tasks that can be performed with the system, etc. Of particular interest in the context of the present discussion are those models that are dynamically updated during interaction, based on knowledge acquired at run-time (the user model being a typical such case).
- *Adaptation decision making*: Refers to the part (or parts) of the AUI that is responsible for deciding upon the necessity of, as well as the required type of, adaptations, given a particular interaction state. Seen at an abstract level, decisions made at this stage match between information found in the various models maintained by the AUI, and the alternative interactions designed to cater for variations therein.
- *Applying adaptations*: Refers to the actual introduction of adaptations in the user-system interaction, on the basis of the related decisions. Although typically subsumed by adaptation decision making in the literature, this adaptation component may be varied independently of the decision making process, e.g., to account for different adaptation strategies.
- *Transparent models & adaptation "rationale"*: Refers to the particular case of AUIs that enable users to review the models maintained by the AUI (at different levels of "transparency" – see [Höök et al., 1996] for a detailed discussion), or the rationale that underlies the adaptation decisions made by the system. In the case of transparent modeling, users may also be offered the capability to modify these models, so that the latter better reflect their individual or other characteristics.
- *Automatic adaptation assessment*: Refers to the run-time assessment of the effects of decided upon and effected adaptations, with the indent of evaluating their "success" (i.e., whether the goals underlying their introduction have been met). This stage is referred to as "second-level adaptation" in [Totterdell & Rautenbach, 1990] and may further involve the modification of aspects of the lower-level adaptation cycle (e.g., by enabling or disabling rules in rule-based adaptation, or by altering the "weight" of alternatives, in decision theory-based adaptation).

It should be noted that this high-level model is not claimed to capture the characteristics of all AUIs reported in the literature. Indeed, the model does not explicitly address, for example, AUIs which adapt on the basis of knowledge derived from interactions with a community of users, rather than with one user (although, generalization of the model in this respect should be straightforward). On the other hand, there do not exist to date AUIs that comprise all of the identified components. However, the modular nature of the evaluation approach proposed allows one to selectively apply it, or extend it to suit the particular needs of the AUI at hand.

C.2.2.2 Modular Evaluation

In this section we will identify adaptation "modules" (comprising one or more of the adaptation stages / components in the previous section), which can be evaluated individually and in combinations. Before proceeding to the presentation of the modules and their evaluation, we would like to make the following clarifications, which hold true throughout the presentation of the approach:

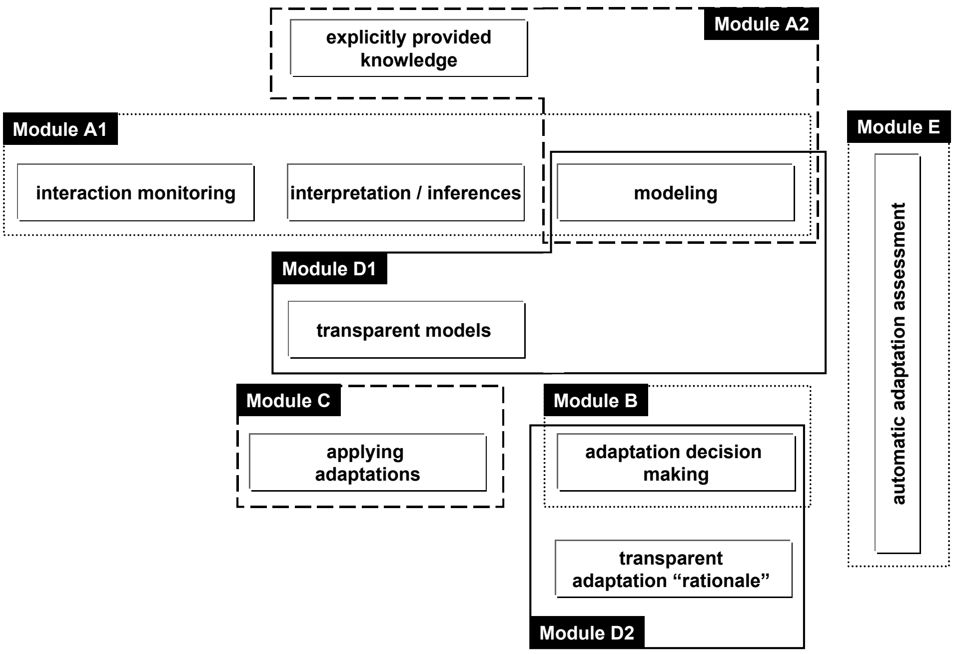


Figure 32: The correspondence between evaluation modules and AUI model components.

- In some cases, evaluation methods that do not involve the users directly, assume that the evaluator / expert takes into account the characteristics (abilities, skills, knowledge, etc.) of the "typical" user of a system. Since the concept of a "typical" user is contrary to the very notion of AUIs, this assumption cannot be applied in AUI evaluation.

Thus, an express requirement that permeates the proposed approach is that, in all cases where users are not directly involved in the evaluation, each and every individual evaluation task takes into account a particular user (conveyed through relevant characteristics, which are encoded in some type of user profile), in a particular context of use (conveyed in a way analogous to the user).

- Expert-based evaluations in HCI are, in general, assumed to be conducted by usability experts. In the description of the proposed evaluation approach we will occasionally refer to expert-based evaluation tasks which are foreseen to be undertaken by individuals that posses expertise relevant to the application domain, the target user group(s), etc., but do not necessarily have a background in usability evaluation.

Let us now move on to the presentation of the modules, wherein, for each identified module, the following information is provided: components comprising the module; evaluation goal(s) and potential evaluation criteria; and, proposed evaluation methods, and prerequisites for supporting these methods.

Module A1

Comprises: *interaction monitoring*, *interpretations / inferences*, and *modeling*. The goal of evaluation in this module is to ensure that the models derived by the system through dynamic interaction assessment are "optimal". Optimality in this context may be related to the following evaluation criteria[40]: *correctness* of the interpretations / inferences (i.e., do the inferences / interpretations reflect that actual state of the entity being modeled?); *comprehensiveness* of the model (i.e., can the model represent in its entirety the inferred / interpreted information about the entity being modeled?); *redundancy* of the model (i.e., does the model contain "attributes" of the entity being modeled, which cannot be inferred from interaction?); *precision* of the model (i.e., how accurately does the model reflect the entity being modeled?); *sensitivity* of the modeling process (i.e., how fast does the modeling process converge to a comprehensive and accurate representation of the entity being modeled?); etc.

In the case of models that directly or indirectly involve the user (e.g., user modeling, plan recognition), one would need to employ a combination of evaluation methods to assess the degree to which the above criteria are met. Specifically, due to the fact that both observations and trace measures can only be used on overt behavior, not on thoughts or feelings or expectations [McGrath, 1995], methods in the *self report* category have to be used. Additionally, methods which allow the users to offer feedback *during*

---

[40] The proposed criteria always refer to what is subjectively perceived by the user, as opposed to what could be "objectively" measured or proven.

interaction are to be favored (to avoid remembering effects), although care should be taken that these methods are not too obtrusive with respect to the interaction itself.

Eliciting user feedback regarding the modeling process requires that at least a prototype of the system exists, with functional *interaction monitoring* and *inferencing / interpretation* components (the *modeling* component could be simulated). Furthermore, users should have some representation of the modeling process itself, which, in this case, can be constrained to the results of the process (i.e., the resulting model or models). If the AUI under evaluation also comprises a functional version of a *transparent models* component, then the latter can be used to that effect (although this might also necessitate a working *modeling* component). If such a component is not foreseen in the AUI (or not available at the time of the evaluation), then an alternative ad-hoc approach to the representation of the model should be sought (e.g., with an observer simulating the model, in a "wizard of oz" type of study).

Expert-based evaluation might also be of use in the early design and evaluation stages for Module A1. In particular, experts may be able to contribute towards the evaluation of *correctness* of inferencing / interpretations, and *comprehensiveness* and *redundancy* of the model. Such involvement of experts would be potentially beneficial, for instance: if part of the user model is related to the application's domain model (e.g., in student models); if the inferencing / modeling process seeks to capture some special user characteristics (e.g., user's ability to interact through a particular input device); etc.

Module A2

Comprises: *explicitly provided knowledge*, and *modeling*. This module is very similar to the preceding one, with the following exceptions:

- Since there is no automatic assessment of the interaction, nor any attempt to elicit / infer information based on such assessment, any related evaluation criteria (including, for example *correctness*) are not relevant.
- Additional criteria that may be considered include: the *transparency* of the process (i.e., whether, and to what extent, the users can understand and / or predict how the information they provide affects the models maintained by the AUI); the *overhead* that may be imposed on the main interaction tasks by the explicit provision of knowledge; etc.
- The involvement of experts in the evaluation of this module might not yield as valuable results as in the case of Module A1. This is due to the fact that the direct "manipulation" of the model(s) is tightly

coupled to the users' mental model of what is being modeled and how, which may be quite hard to simulate or predict.

Module B

Comprises: *adaptation decision making*. The goal of evaluation in this module is to ensure that the adaptation decisions made by the respective component are "correct". Correctness in this context may be related to the following evaluation criteria: *necessity* of adaptation (i.e., is an adaptation indeed required in the current interaction context?); *appropriateness* of adaptation (i.e., is the adaptation decided upon one that can cater for the requirements posed by the current interaction context?); *acceptance* of adaptation (i.e., does the user think that the adaptation is both necessary and appropriate?); etc.

A fundamental difference between this module and previous ones is that it does not (initially) require that any parts of the adaptation infrastructure have been implemented (although it does require that the alternative interaction artifacts have been designed). This is due to the fact that the adaptation logic relates interaction states (as these are depicted in the maintained models) to specific adaptations; thus, if such states can be reproduced or even simulated, it is possible to evaluate the related decisions "in context". The decomposability of adaptation logic is of course constrained by the degree to which adaptation decisions affect each other (e.g., two decisions may be mutually exclusive, if they affect the same facets of interaction, but in different ways).

In practical terms, in a typical adaptation design cycle, a theory, a set of hypotheses, or past empirical findings, will serve as input to the initial corpus of adaptation logic. This corpus can then be validated, in a first stage, using the aforementioned most formative evaluation methods to assess the *necessity* and *appropriateness* of adaptations.

Contrary to the above, it may be difficult (or even impossible) to extrapolate the overall *acceptance* of an adaptation decision in the same manner. This, combined with the requirement to further explore the other two criteria, when the entire corpus of adaptation logic is "active", points to the necessity of a second stage of evaluation in this module, in which users will experience adaptation decisions in "real time".

In either stage, to enable the participation of users in the evaluation, there needs to exist an explicit representation of the decisions made. This is actually a non-trivial requirement, especially in the case that the components that undertake decision making and adaptation application are indeed separate (because, then, users would have to attain an understanding of a decision, without detailed knowledge of how it would be applied in practice). If the AUI comprises a *transparent adaptation* rationale component, then this could be utilized to offer the users the required representation. Otherwise, like in the case of Module A1, a different (probably ad-hoc) approach to the representation of decision triggering (presumably on the basis of dynamic modifications in the maintained models) and decision making would be required. If none of the preceding were feasible, an alternative approach would be to treat modules B and C jointly, in terms of evaluation, as described in the section entitled "Evaluating across modules".

Expert-based evaluation can also play a fundamental role in this module. This is true especially for the first of the two stages described above, and for the first of the two criteria proposed (i.e., *necessity* and *appropriateness*). The involvement of experts could lift the requirement for functional prototypes (or simulations) of the involved adaptation modules, if a structured approach was followed, within which adequate documentation and instruments were provided to the experts in order for them to be able to: (a) fully associate the interaction context (including user characteristics) that triggers a decision, with all the facets of interaction that the decision affects (and, of course, the ways in which it affects them), (b) assess the interplay of decisions on interaction, i.e., assess the possible / potential combined effects of sets of decisions triggered in the same (or similar) interaction contexts. This approach was followed quite successfully in the evaluation of the adaptation rule base of the adaptive user interface of the AVANTI web browser [Paramythis, Totter & Stephanidis, 2001].

Finally, it is interesting to note that the evaluation of the current module as well as of Module C (described next), is both the most challenging and most interesting part of the evaluation of AUIs, as these two modules "contain", in effect, the theory underlying adaptation in the user interface. This theory (typically expressed in the form of higher-level hypotheses) is what researchers have actually sought to evaluate in previous work. However, evaluating the AUI as a "black box" is what has restricted the scope and validity of efforts in the past, since not "separating" the module from the rest of the AUI, results in the concurrent assessment of possibly numerous influencing factors.

Module C

Comprises: *applying adaptations*. The goal of evaluation in this module is complementary to the one for Module B above, and can be expressed through criteria such as: *timeliness* of adaptation (i.e., is the decided upon adaptation applied in a timely manner - e.g., not too late?); *obtrusiveness* of the adaptation (i.e., how obtrusive, or obstructive is the application of an adaptation, with respect to the users' main interaction tasks); *user control* over the adaptation (i.e., can the user disallow, retract, or even disregard an adaptation?); etc. Furthermore, these criteria can be thought of as directly contributing towards the criterion of *acceptance* in Module B.

The evaluation of this module should be treated very carefully, and any related evaluation activity should be designed very carefully to measure only the criteria relevant to this module. The difficulty in doing so arises from the fact that the users "experience" the grand total of the system's adaptive behavior through the adaptations that are effected (and of which they are aware). In other words, users will, in all but the most trivial cases, not be able to (and should not be asked to) distinguish between modeling, decision making and adaptation application. It is the task of the evaluator to factor out any "interference" from the preceding adaptation stages. One feasible approach to achieving this would be to evaluate this module only after Modules Ax and B have been evaluated, and any necessary modifications to the respective AUI components have been made.

The evaluation of adaptation application with the involvement of end users, requires that the AUI "feels" like a complete interactive system, i.e., functional prototypes (or simulations) of all AUI components should be present. This requirement stems from the need to enable users to situate themselves in the actual interaction context in which adaptations would take place. If this requirement is not met, then the only criterion that may be assessable is the degree of control that users feel they have over adaptations. *Timeliness* and *obtrusiveness* cannot be evaluated by end users unless they are actually "immersed" in realistic situations.

The assessment of this module by end users also points to the direction of summative evaluation methods, and especially ones where factors external to the user-system interaction are kept to a minimum (such as, for example, controlled experiments). Expert-based evaluation is not likely to render any significant results in this module, with the exception of initial design cycles (targeted at identifying and treating any major flaws in how the system applies adaptations).

Module D1

Comprises: *modeling*, and *transparent models*. The goal of evaluation in this module is to ensure that the users' perception of the maintained models matches the actual state of the models. This translates into evaluation criteria such as: *completeness* of the presentation (i.e., does the user have a full –perhaps abstracted– view of what is modeled and the current contents of the model?); *coherence* of the presentation (i.e., how well can the user understand the individual attributes of the model); *rationality* of the presentation (i.e., does the user understand why the model is in its current state?); etc.

The evaluation of this module can follow a two-staged approach. In the first stage, end users and experts can be involved in the assessment of the actual representation of the model, addressing issues such as the level of detail to be employed, the level of transparency that is necessary to provide the user with adequate information, but without exposing unnecessary internal modeling details, etc. This stage, which may not even necessarily require the presence of interactive prototypes, can thus explicitly target the *completeness* and *coherence* criteria.

The second stage of the evaluation would have to address how users experience and perceive the model(s) *during* interaction. This stage is complementary to the previous one and is mainly intended to address the *rationality* criterion. The assess the latter, a user would actually need to have a full understanding of the interactions that have led to a particular situation, in order to be able to judge whether the presented model "makes sense". During this stage, it would be preferable to elicit user feedback during interaction, so as to avoid any rationalization effects that may interfere with post-interaction evaluation. A possible compromise might be to structure interaction into small tasks and request users to provide their feedback between tasks (e.g., by answering short, targeted questionnaires).

Module D2

Comprises: *adaptation decision making*, and *transparent adaptation rationale*. This module is similar to the preceding one, with the main difference being that what is presented to the user is not a model, but rather the rationale underlying a particular adaptation (which could also have the form of a recommendation made by the system). Thus, evaluation criteria that may be relevant include: *coherence* of the adaptation rationale (i.e., how well can the user understand what the rationale refers to – e.g., what is / will be adapted and in what way?); *causality* of the rationale (i.e., does the user understand what triggered a particular adaptation?); etc.

Although there exist subtle differences between this module and Module D1, the evaluation can follow, in general, a similar approach. A notable difference is that the second stage of the evaluation (as described for Module D1) would, in this case, require that the AUI is functional (or simulated) almost in its entirety.

Module E

Comprises: *automatic adaptation assessment*. The goal in this module is to ensure that the system shares the same views as the users with regards to the "success", or "failure" of adaptations. This goal differs significantly from the ones expressed in the previous modules, in that the users' feedback regarding specific adaptations and their effects on interaction needs to be captured and subsequently compared to the system's view of the same adaptations. Seen from a different perspective, if this AUI component assesses and modifies the lower-level adaptation "strategies", then what needs to be evaluated is whether any such modifications are optimal from the perspective of the user.

Although, from an engineering perspective, the AUI component(s) involved in "adapting the adapter" operate at a meta-level with respect to the rest of the AUI components, this distinction may not be relevant from the perspective of evaluation. Specifically, it may be possible to treat these "meta-adaptations" as just another type of adaptations taking place in the interface.

This would mean that meta-level adaptations are amenable to the same treatment as first-level adaptations, and can thus be included in the modular evaluation as this has been described so far.

Evaluating across modules

Evaluating each of the above modules in isolation may deliver significant results, but there remain a number of questions that cannot be answered unless modules are evaluated in combination. Additionally, some of the components may be tightly coupled in an implemented AUI, which might render it impossible to evaluate some of the modules in isolation. The rest of this section briefly presents some tentative module combinations and their significance in terms of AUI evaluation.

Modules Ax and D1: These modules capture the entire process of constructing models (automatically, or through explicitly provided information) and presenting these models to the user. Evaluating these modules in combination may offer a more global perspective on how users perceive modeling in the AUI, and allows one to investigate other relevant aspects, such as whether users are comfortable with whatever private information is included in the models, whether they "trust" the system with such information, etc.

Modules B and C: These modules capture the process of deciding upon and applying adaptations in the AUI. Evaluating them in tandem may be inevitable if the AUI does not distinguish between the respective components in a way that allows for treating them separately. On the other hand, even if the AUI does distinguish between the components, it is possible to treat them jointly by: (a) enumerating all the possible methods in which an adaptation decision can be applied, and (b) treating each decision-method pair as a distinct decision.

Modules C and D2: By evaluating these modules in combination, one could, for example, address the questions of how "predictable" and how "controllable" users perceive the AUI to be. Whereas the perception of predictability might result from the user's ability to understand the circumstances under which adaptation decisions are made, the perception of controllability might result from the users' ability to control both the circumstances that lead to an adaptation decision, and the application of the decision as such.

Modules Ax, B and C: The combination of these modules captures the entire "traditional" adaptation cycle in an AUI, and can thus be thought of as evaluating the AUI as a "whole". Although it is argued that the evaluation of this combination should not commence until the modules have been addressed individually, there are questions regarding the adaptive theory employed in the AUI, which can only be posed at this level (such as, for example, "does adaptive task guidance improve the ability of novice users to complete complex tasks in the user interface?")

C.2.3 Reported Work against the State-of-the-Art

This chapter has proposed a new approach to the evaluation of AUIs, on a basis that can facilitate and guide the modular assessment of various components / stages of the adaptation cycle. At the same time as the original paper from which this chapter was derived was published, two highly related proposals were also put forward. Brusilovsky, Karagiannidis & Sampson [2001] reported on the successful application of "layered" evaluation for revisiting a previously inconclusive study reported in [Brusilovsky & Eklund, 1998]. Their approach involved the separation of the adaptation process into two "layers", one encapsulating the acquisition of a user model, and one dealing with the application of the user model for adaptation. By revisiting data collected during the past evaluation, the authors were able to demonstrate the practical value of splitting the adaptation process into steps that are then distinctly addressed in evaluation.

A similar process-based approach, also under the moniker of layered evaluation was proposed by Weibelzahl (2001). The proposed framework discerns four layers that refer to the information processing steps within the adaptation process: evaluation of input data, evaluation of the inference mechanism, evaluation of the adaptation decision, and evaluation of the total interaction. The framework has a very clear focus on the empirical evaluation of IAS and has been applied in practice to different adaptive learning courses, including several studies with thousands of users.

Compared to these two frameworks, the one described in this chapter addresses the issue of formative vs. summative evaluation and, overall, adopts a more "engineering" perspective in the identification of layers, focusing in more detail on the different components involved in the adaptation process. It also addresses the question of which methods and tools might be appropriate for the evaluation of different adaptation layers, in order to elicit input for the development process, which is lacking in the other approaches.

Although there are obvious differences in the approaches advocated by the frameworks discussed thus far, there is inarguably also a lot of common ground. This prompted the initiation of a cooperative effort to merge the common themes of these frameworks, focusing primarily on the one presented in this chapter and the one in [Weibelzahl, 2001].

This effort resulted in the introduction of a new, unified framework, presented in [Paramythis & Weibelzahl, 2005]. The unification addressed first and foremost the establishment of a decomposition model for adaptation, which identifies five main "stages" of adaptation: (a) collection of input data, (b) interpretation of the collected data, (c) modeling of the current state of the "world", (d) deciding upon adaptation, and (e) applying adaptation (i.e., effecting adaptation decisions). The position taken is that these stages are to be treated as interdependent layers that must be explicitly addressed when evaluating adaptation. This framework is currently regarded as the main methodological approach to the evaluation of IAS, and has been presented most recently in a tutorial held in conjunction with the User Modeling 2007 conference [Weibelzahl, Paramythis & Masthoff, 2007].

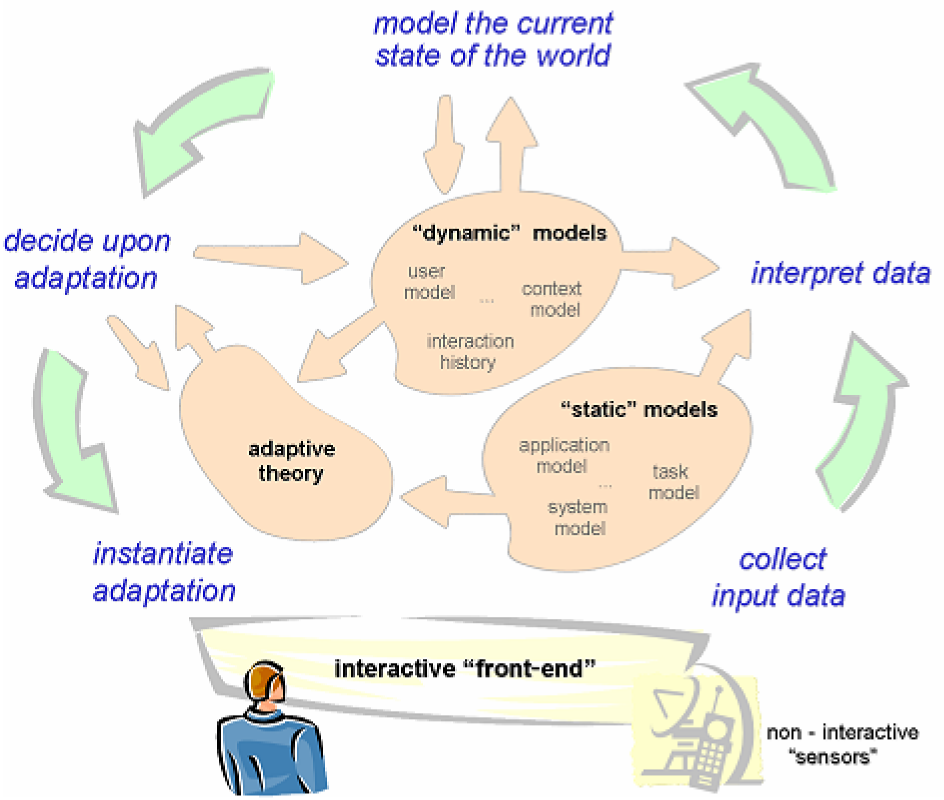


Figure 33: Decomposition model for the layered evaluation of adaptive systems [Paramythis & Weibelzahl, 2005]

Complementary, as well as alternative approaches to the evaluation of adaptation that have been proposed in the literature include the ones presented in [Magoulas, Chen & Papanikolaou, 2003], [Herder, 2003], [Tobar, 2003], and [Tarpin-Bernard, Marfisi-Schottman & Habieb-Mammar, 2009].

Magoulas, Chen, and Papanikolaou [2003] argue about the need to develop an educational-evaluation model and a methodology that include usability testing as standard procedure capable to determine the impact of adaptation on learners' behavior in an educational environment. To this end, they introduce modifications to the standard heuristic evaluation approach and augment it with criteria that diagnose potential usability problems related to adaptation, subsequently integrating it into the layered evaluation approach.

Herder [2003] proposed a utility-based approach to the layered evaluation process. The basic idea is that the added value of an adaptive system can be expressed by a utility function that maps selected, measurable criteria with respect to the performance of the adaptive system to a quantitative representation. If one would compare an adaptive system with its non-adaptive counterpart, the value of adaptation is the difference in utility between the two systems. Herder [2003] argues that the main advantage of the layered evaluation approach in this context is that it separates the utility function in several functions in a principled manner.

Tobar [2003] proposes a different approach based on a so-called map which integrates different design perspectives to facilitate the understanding of adaptation assessment and design. Tobar's proposed framework is more targeted towards the identification of specific adaptation features that need to be assessed, the establishment of criteria for the assessment, and the generation of evaluation plans on this basis.

The approach proposed by Tarpin-Bernard, Marfisi-Schottman, and Habieb-Mammar [2009] is somewhat related to the one proposed by Tobar, but has important differentiations as well. Instead of prescribing the procedural means for identifying adaptation features for assessment, the authors provide a relatively exhaustive enumeration of potential adaptation constituents and determinants in an IAS in a tabular form. Evaluators can use the resulting table to determine exactly what needs to be assessed, and are facilitated in establishing potential conflicts and correlations (e.g., where the same determinant affects several constituents). This framework is also unique in that it attempts to summarize and quantify the "degree" of adaptation in a system, and in that it is supported by a web-based tool that enables evaluators to interactively manage the tabular description of the system at hand. Although this framework is still at the early stages of its development, it appears to bear promise in structuring the adaptation space in an easy to understand way. It would also be interesting to see future work examining the extent to which this approach can be used in conjunction with modular / layered evaluation.

U n i t D

META-ADAPTIVE SYSTEMS

Theoretical and Practical Issues on the Road to Meta-Adaptivity

As already pointed out in section "A.1.2.2 Adaptation Taxonomies", with reference to the taxonomy of Totterdell & Rautenbach [1990] (see also Table 1), the current generation of adaptive systems falls in the area of adaptivity, with several more levels theoretically attainable. This Unit discusses work towards meta-adaptive systems (i.e., systems that can themselves assess and adapt their own adaptive behavior), and specifically targets the level of self-regulation.

Specifically, the first Chapter, "D.1 Towards Self-Regulating Adaptive Systems", presents preliminary work towards a theoretical basis intended to facilitate the development of self-regulating adaptive systems. Self-regulation refers to the capacity of the system to assess the effects of, and modify, its own adaptive behavior in prescribed ways at run-time. Although not new, the concept of self-regulation is largely missing from existing adaptive systems, arguably due to the perceived complexity involved in its theoretical grounding and practical implementation. This Chapter addresses in particular the following two questions: What are the operational requirements of self-regulating adaptive systems? What implications does self-regulation impose on the modeling- and decision making- approaches used? The theoretical benefits of "clusters" of self-regulating systems, and the role of human experts in the self-regulation process, are also briefly discussed. Finally, the work reported here is contrasted with other related efforts in the recent literature.

The second Chapter, "D.2 Self-regulated Adaptivity as a Design and Authoring Support Tool", discusses the changes self-regulated adaptivity will potentially bring about in the culture of designing and authoring adaptive systems. Two dimensions are explored in parallel: the opportunities arising from the employment of self-regulation itself as a tool that can facilitate the design of adaptation; and, the additional requirements self-regulation places upon the traditional authoring process and on the adaptive system itself. The discussion is structured around a specific example tackling an often-discussed problem in the domain of adaptive course delivery systems: adaptive annotation of concept-based hypermedia links. This example is built upon to argue that self-regulation is a viable solution to the "ground-up" design of adaptive systems, and may be used even in (or, for that matter, best suit) cases where there is little empirically validated evidence to support design decisions.

D.1 Towards Self-Regulating Adaptive Systems[41]

The concept of "self-regulating" adaptive systems was proposed in the late eighties by [Trevellyan & Browne, 1986] and was introduced in a taxonomy of adaptive systems by [Totterdell & Rautenbach, 1990] (see Table 1 and Table 2). The rest of this section will provide an informal definition of self-regulation in adaptive systems, and discuss its potential benefits, as well as the factors that have had detrimental effects in its employment in adaptive systems.

To start with, along the lines set out in [Trevellyan & Browne, 1986] and [Totterdell & Rautenbach, 1990], it is argued that effecting self-regulation in adaptive systems requires that the later be capable of learning. This specifically entails that the adaptive system be capable of (incrementally) modifying the "knowledge" [42] it uses for deciding upon adaptation. Such learning would basically result in adapting the system's own adaptive behavior, to better accommodate different users, situations, environments, etc. This, in turn, would necessitate the capability, on the part of the adaptive system, to assess its own adaptive behavior and determine whether it has met its goals (or, in other words, whether it has had the desired effects) and act accordingly.

To understand the motivation behind self-regulation in adaptive systems, let us first consider the typical operation of "traditional" adaptive systems. Firstly, adaptive systems create a model of their environment, which involves at the very least the system's user, and may also incorporate other dynamic and static information affecting interaction (e.g., the context of use). Secondly, the system's adaptation "logic" [42] (embodied in rules, Bayesian decision networks, neural networks, etc.) correlates the model(s) of the system's environment with a range of adaptive behaviors that the system is capable of. An important point to note is that what we referred to as the system's adaptation "logic" is never updated dynamically / automatically (or, at least, not without human intervention). An obvious benefit of this approach is that the system's adaptive behavior is predictable. This also constitutes, however, the weakest point of traditional adaptive systems: adaptation logic is never "questioned", and is applied "blindly" (i.e., irrespectively of whether it actually achieves the desired effects or not).

---

[41] This chapter is based on [Paramythis, 2004]. The work reported here has been supported in part by the project "Integrating Agents into Teleteaching Webportals" sponsored by the Austrian Fund for the Support of Scientific Research (FWF; project P15947-N04). Please refer to section "A.2.2 Work Context and Research Projects" on page 32 for additional information.

[42] The term "knowledge" is used here to generically refer to the whatever combination of modeling- and decision making- approaches are employed by the adaptive system to achieve adaptive behavior, and does not imply that a "knowledge-based" approach is used to that extent. Similarly, the term "logic" is used to refer generically to the decision making approach, rather than the employment of logic-based reasoning, etc.

The main goal behind self-regulation, then, is to enable adaptive systems to progressively validate, and where necessary, modify their own adaptive behavior in prescribed ways. Totterdell and Rautenbach [1990] also argue that the levels of adaptivity reflect a change of intention moving from a designer specifying and testing the mechanisms in a (simple) adaptive system, to the system itself dealing with the design and evaluation of its mechanisms in a self-modifying system. The most obvious, and perhaps simplest, modification a self-regulating system can apply to itself is to "demote" (the use of) adaptation logic that does not have the desired effects.

Since self-regulation bears such great promise, why is it then that it has not yet proliferated in adaptive systems? The answer is two-fold: On the one hand, self-regulation *is* part of some adaptive systems in wide use today in different guises (e.g., recommender systems which use implicit and explicit user feedback to modify their recommendation strategies), albeit in rather restricted forms. On the other hand, as Benyon [1993] points out, moving up the levels of adaptivity incurs an increasing cost, which may not be justified. The most prominent cost in employing complete approaches to self-regulation is the inherent requirement for self-evaluation. Furthermore, there do not exist, to date, proposals on how self-regulation can be formalized and applied across the wide range of approaches to modeling and decision-making, common in adaptive systems today.

The rest of this chapter discusses the main premises of a theoretical basis intended to facilitate the development of self-regulating adaptive systems. Due to lack of space the discussion is informal, does not go into detail, and necessarily leaves out topics that some might consider pivotal to self-regulation. The topics that are discussed include the operational requirements of self-regulating adaptive systems, and the implications of self-regulation in relation to modeling and decision-making. The chapter is concluded with a brief discussion of the theoretical benefits of "clustering" self-regulating systems, and the role of human experts in the self-regulation process.

D.1.1 Dissecting Self-Regulation – Operational Requirements

In general, self-regulation in adaptive systems requires what has been termed a "two-level adaptation architecture" [Totterdell & Rautenbach, 1990], depicted schematically in Figure 34. This type of architecture requires that the system have two adaptation foci: one for adapting its interactive behavior, and one for adapting its adaptive behavior. The input to the "first-level adaptor" comprises user interactions and may include information provided explicitly by the user, characteristics of the context of use (as conveyed by appropriate "sensors"), etc. The output of the "first-level adaptor" consists of modifications effected to the system, which directly or indirectly affect user interaction with the system. The input to the "second-level adaptor" comprises that of the "first-level adaptor" and, in addition, the actual modifications applied to the system as a result of first-level adaptation logic. Although not depicted in Figure 34, this input may also encompass (or, actually, be entirely composed of) information from the various system models. The output of the "second-level adaptor" consists of modifications applied to the "first-level adaptor", which effectively alter the system's apparent adaptive behavior.

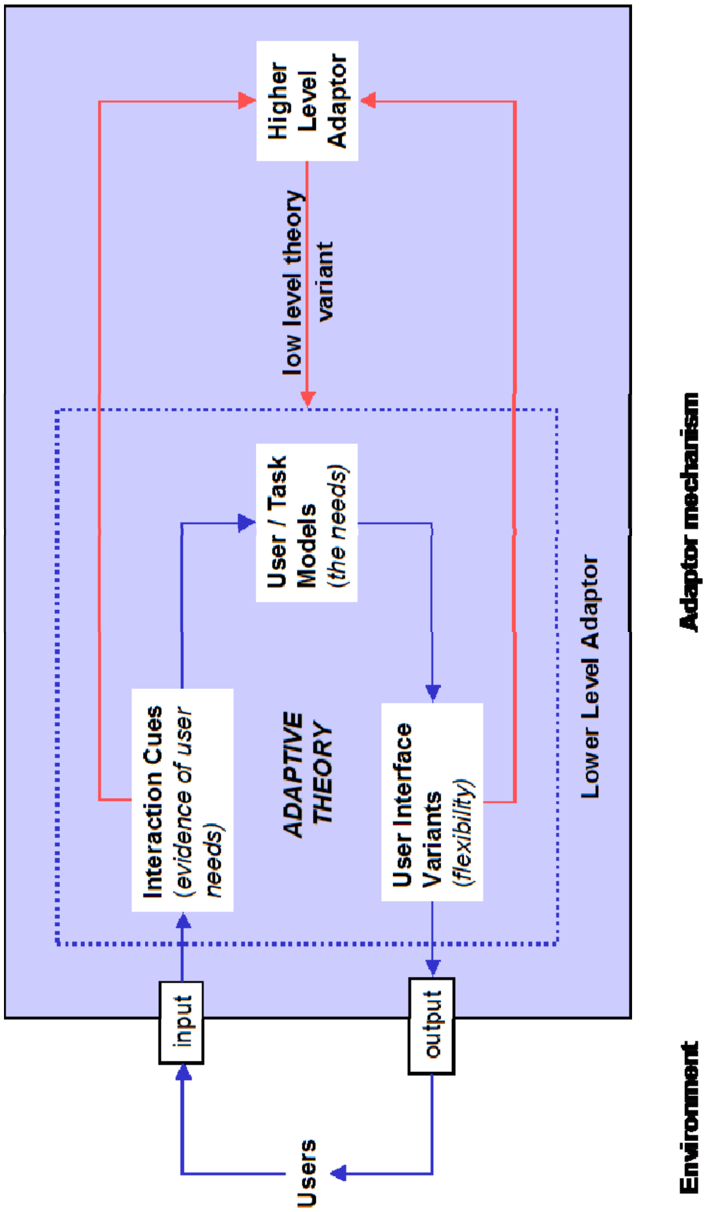


Figure 34: Logical diagram for a two-level adaptation architecture. (adapted from [Totterdell & Rautenbach, 1990])

Let's look now in more detail into the actual operational requirements that the above scenario translates into. In short, the requirements that we will be looking at include: observing interaction; observing adaptive behavior; self-evaluation; and, modifying adaptive behavior.

To start with, self-regulation requires that user interaction with the system be observed and interpreted. This requirement should be trivial to satisfy, as this is an integral part of the operation of any (user-) adaptive system. A far less trivial requirement is that the system's adaptive behavior be observed and modeled. This implies that the system's adaptive behavior must be "broken down" into relatively discrete constituents (the granularity may vary widely) that can be uniquely identified. As we will see in the next section, although it is possible to relax this requirement somewhat, there are implications which constrain the types of adaptive systems in which self-regulation can be applied / implemented.

The third and, perhaps, most demanding requirement of self-regulation is that of self-evaluation. Self-regulating adaptive systems must be capable of assessing the (degree of) success or failure of the system's adaptive behaviors. Although such assessment may take many forms, this work proposes and concentrates on an approach that is, arguably, realistic in terms of implementation costs (given a supporting software framework), and whose overhead in terms of adaptation design are not forbidding. This approach is based on the identification of "expectations" in relation to a system's adaptive behaviors, and has been inspired by the work reported in [Browne, Norman & Adhami, 1990]. The term "expectations" refers to the anticipated benefits that a particular behavior will have on the interaction state. Expectations need to be expressed in quantifiable terms and in relation to the adaptive system's dynamic models (which, presumably, comprise a representation of the current interaction state), or to direct user input. The quantified expectations must then be expressed in computable form, and associated with their corresponding adaptive behaviors. Given the proposed approach, self-evaluation can be defined as the process of assessing adaptive behaviors with the computable expectations acting as metrics used to "measure" (degrees of) success or failure.

The fourth and final requirement concerns the capability of adaptive systems to modify their own adaptive behavior. This implies that the system is capable of either: (a) modifying its first-level adaptation logic at run-time, thus affecting its adaptive behavior, or (b) leaving the adaptation logic unmodified, but overriding the resulting adaptive behaviors (which, in effect, is equivalent to the establishment of second-level adaptation logic). Which of the preceding capabilities are plausible for a given adaptive system depends, mainly, on the way in which the system does its decision making. We will return to this topic in the next section.

The above four requirements are, of course, only a sketchy outline of what is needed for self-regulation. Each of the requirements has several additional implications, some of which will be discussed in the next section. Before doing so, though, we need to elaborate on a number of points.

The first such point is how one can address the modeling and quantification of expectations and their fulfillment, in the context of self-evaluation? After all, evaluation of adaptive systems is known to be a problematic issue in its own accord, even when carried out by humans, so shouldn't self-evaluation be next to impossible?

Starting from the second question, let us delineate the most important differences between self-evaluation in the context of self-regulation, and the "external" (empirical) evaluation of adaptive systems. As recent work has shown [Weibelzahl, 2001; Paramythis, Totter & Stephanidis, 2001; Paramythis & Weibelzahl, 2005], because of the inherently complex nature of adaptive systems, identifying the exact reasons for failure of any given adaptive behavior is quite demanding and requires a structured, methodological approach. Self-evaluation within the self-regulation process, however, need not be concerned with "understanding" *why* an adaptive behavior fails in a given interaction context, but only that it does – the occurrence of failure can then trigger corrective behavior on the part of the system. Some may argue that without knowing the reasons of failure, a system cannot possibly hope to provide a viable alternative. As we will see in later sections, for self-regulating systems, this is a point that can be addressed through human intervention. It should be noted, however, that for systems further up the scale of adaptivity (e.g., self-mediating systems), the reasons for failure are almost equally important as the failure itself. In synthesis then, and in the perspective of this thesis, self-evaluation of an adaptive system can be informed from, but, at the same time, is entirely distinct from, the "external" evaluation of that same system.

Having established the scope of self-evaluation, let's return to the question of how one models or quantifies expectations. The simplest case, in this respect, would be adaptations that are expected to result in individually observable user actions (e.g., user follows a link specifically annotated to encourage selection). Expectations, in this case, could then be codified as the requirement that such actions occur within given temporal or other constraints. This, in turn, necessitates the presence of: (a) "primitives" which can be used to refer to user actions at a semantic level of abstraction, and (b) "constraint languages" that can be used to apply constraints on the aforementioned primitives. For example, consider an adaptive system in which a set of links are reordered according to a specific adaptation strategy. The expectation to be expressed might then be that users select items (primitive action) from the top of the reordered set (first constraint), soon after the reordering has occurred (second constraint). For exemplification, counter-evidence for the success of the adaptation might be that the users do not select any of the items; or, even worse, select items away from the top of the set.

On the other end of the complexity spectrum would be expectations that can be only approximately expressed, include uncertainty, and require an understanding of the user's interactive behavior (as opposed to mere observation of the user's actions). Whereas approximate goal descriptions and uncertainty can be tackled through the employment of appropriate reasoning techniques, acquiring an understanding of the user's behavior is a more involved matter. A pragmatic approach to this requirement would be to express expectations in relation not only to user actions, but also to (changes in) the dynamic models maintained by the system. The premise of this approach is that these models actually embody the system's continuously updated understanding of the "outside world". It is argued that this kind of extension would require few changes in the "constraint languages" discussed above. It would, nevertheless, require a different set of "primitives" capable of capturing the notion of changes in the models, in relation to other dimensions of adaptation (with the temporal dimension playing again a very significant role). To exemplify the concepts discussed, consider the case of an Intelligent Tutoring System, which detects that a user has very limited knowledge of a topic that is a prerequisite for other topics in a delivered course. Adaptations performed at such a stage would be expected to result in the user's knowledge of the topic (primitive) as reflected in the corresponding student model, to be increased (primitive-specific constraint).

A related issue, as mentioned earlier, is that "expectations" need to be associated with specific system behaviors; this, however, is only the design-time part of the picture. At run-time, "expectations" also need to be aware of the context (i.e., the system's current beliefs about the user and the environment, as expressed in the system's dynamic models) within which behaviors were decided upon. This is necessary if the system is to be able to differentiate between contexts in which particular adaptation behaviors have the desired effects, and contexts in which they don't. It is also necessary in order to express "expectations" based on changes in the context over time (e.g., measuring changes in a user characteristic in the system's user model, from the time that a particular adaptation was effected).

Another point to be addressed before continuing to the next section regards what can be considered a marginal case of self-regulating adaptive systems. This is the case whereby the system exposes in some form its adaptation model to the user (much in the same way that user models are exposed in traditional adaptive systems) and allows the user to provide direct and explicit feedback with respect to any specific adaptive behavior. This would eliminate the need for self-evaluation per se (as this is delegated to the user), but would introduce a host of other problems, not the least among which are the enormous overhead imposed on the user, and the challenge of providing a concise yet meaningful representation of the adaptation model for non-expert users in the first place. Given the above constraints, this approach should be considered infeasible at the current stage of evolution in the field of adaptive systems.

D.1.2 Implications on Modeling- and Decision making-Approaches

Let's turn our attention now to the implications of the operational requirements posed in the previous section, on the way in which adaptive systems model users (and the interaction context more generally) and decide upon adaptations. We will start from the additional input required for the second-level adaptor.

D.1.2.1 Observation

As already mentioned, self-regulation requires that the system's adaptive behavior can be observed (and possibly also interpreted). But what exactly *is* observed in this case? The field of adaptive systems is infamous for its lack of standards, or even commonly accepted approaches in this respect. Instead of going into details, which are inevitably bound to specific platforms or architectures (consider, for instance, the differences in adaptive behavior between hypertext and desktop systems), let us focus on how adaptive behavior must be manifested in a system, so that it can be observed:

- Firstly, it must be possible for the second-level adaptor to "learn" when the system's adaptive behavior changes.
- Secondly, it is necessary that the changes incurred be semantically interpretable (i.e., the adaptor must be capable of "understanding" what has changed).

The first of the above characteristics is obviously vital to the operation of self-regulation. The second characteristic, though, is less fundamental than one might originally consider; we will return to this topic shortly. For the time being, it suffices to note that the level and granularity of the interpretation of changes in a system's adaptive behavior can vary widely. Apparently, the more fine-grained an understanding attainable, the more detailed self-evaluation and subsequent interventions can be.

D.1.2.2 Self-evaluation

After collecting its input, the second-level adaptor proceeds to the stage of self-evaluation. As already mentioned, this stage involves the assessment of the (degree of) success or failure of the system's adaptive behavior. This implies the capability on the part of the system to quantify the changes that have occurred in the interaction state as a result of applied adaptations. To facilitate discussion, we will assume that such quantification is done through "functions" applied within the second-level adaptor and we will set out to explore their characteristics:

- *Function inputs*: This may comprise direct user input, current values from the static and dynamic models of the system, "historical" values from the same models, as well as interim results from previous calculations. The term "historical" is used in this context to refer to the values that modeled characteristics had at a given point in time, and they are important when comparisons need to be made, to establish the changes in the models brought about through adaptation. This kind of "memory" is not a typical capability of current modeling components, and may need to be provided by the second-level adaptor itself.
- *Function output*: The value domain of the functions' output can be practically anything (e.g., Boolean, discrete, fuzzy, etc.) and actually depends on the computational approach employed for the implementation of the functions (e.g., a probabilistic approach will have a value domain of [0 ... 1]).
- *Computational approach*: This can be the same as the approach used to implement the first-level adaptation logic, but can also be entirely different. Elaborating on the last point, it is interesting to note that, given a sufficient degree of similarity in how adaptive systems communicate model and adaptation information to the second-level adaptor, it is possible to create a generic computational approach to self-evaluation that can be used "above" several different types of first-level adaptation logic.
- *Association with the adaptation(s) being evaluated*: This has already been briefly discussed in the previous section, and an assertion was made that it is not as fundamental as one might expect at first sight. As we will see in the next section, for some types of intervention (i.e., modification of the first-level adaptive behavior), the only thing necessary is that adaptations can be uniquely identified, so that the two adaptors can "converse" about them (e.g., the second-level adaptor instructing that a specific adaptation be disallowed for the current user). As we will see, even the requirement for *unique* identification can be relaxed, if interventions occur at a sufficiently low level – however, adaptation types would still need to be identifiable.

D.1.2.3 Modifying the system's adaptive behavior

Following self-evaluation, the second-level adaptor may need to intervene and modify the first-level adaptive behavior. This implies that the system is capable of either: (a) modifying adaptation logic at run-time, thus affecting its adaptive behavior, or (b) leaving the adaptation logic unmodified, but overriding the resulting adaptive behaviors. The run-time modification of adaptation logic is a process that is evidently dependent on the type of logic used (e.g., in rule-based systems, it would signify modification of rules; in systems based on Bayesian networks, it would signify changes in the network, etc.). As such, this type of intervention is too broad a subject to address here. Instead, we will focus the second method of intervention, which does not presuppose any modifications to the first-level adaptation logic.

This second method requires that the second-level adaptor can override adaptations decided upon by the first-level one. Overriding, in this context, can take several forms, the most important among which are:

- *Disabling adaptations*: Arguably the simplest form of overriding is to disallow adaptations from occurring when there is evidence that they have detrimental effects on the interaction. If adaptations can be uniquely identified and associated with a context (i.e., user characteristics, interaction state, etc.), then disabling can occur at a quite fine-grained level. Lack of unique identification, and, similarly, lack of context associations, results inevitably in more "global" overriding effects (i.e., *all* instances of a particular adaptation type are disallowed, or a specific adaptation is disallowed in *all* contexts; apparently, this may result in disabling by implication even adaptations that had positive effects on the interaction).
- *Constraining adaptations with weighting functions*: This form of overriding is based on the employment of additional functions that use the results of self-evaluation to "promote", or, more usually, "demote" adaptations. Demotion, in this context, refers to the application of additional constraints on the circumstances under which an adaptation is allowed to take place (with promotion having the converse effect). These constraints might be entirely independent from those used for deciding upon the adaptation at the first level. One plausible approach to such weighting functions would be, for instance, "utility" functions, as described in [Horvitz, 1999], or, from a different perspective, in [Herder, 2003]. Disabling adaptations, as discussed above, can be seen as a special case of constraining, with a Boolean weighting function.
- *Using alternatives*: This form of overriding presupposes the presence of alternatives for given (types of) adaptations. Note that the second-level adaptor does not necessarily need to understand the differences between alternatives. Using a trial-and-error approach, for example, would enable the adaptor to identify the one most suitable for a given context, without knowing how the alternatives actually differ. Disabling adaptations can also be seen as a special case of using alternatives, with two alternatives for each adaptation, one being the "null" or "empty" alternative. Although promising, this approach incurs additional overhead in the design and development of the adaptive system, as the adaptation model will need to have a

representation of the alternatives themselves and of their associations with logic (see, e.g., [Savidis et al., 1997]).

- *"Editing" adaptations*: The most sophisticated form of overriding is to actually modify the adaptation itself. This is a quite demanding endeavor, as it requires that the second-level adaptor: can acquire semantic information about an adaptation's constituent parts[43] and their individual effects on interaction; can modify these constituents to achieve different effects by altering their parameters, or by replacing them with alternatives, or, even, by simply removing them. Arguably, the most challenging part in all this is that the above process presupposes the existence of meta-knowledge that would allow the second-level adaptor to decide what exactly to change and why. A complete solution in this respect falls more within the scope of self-mediating systems, rather than self-regulating ones. A pragmatic approach, however, may be based on the concept of "templates" which could specify ways in which the adaptor can modify specific categories of adaptations. The task then would be reduced to identifying (on the basis of the self-evaluation results, the context associated with the adaptation, and on the nature of the adaptation itself) which template needs to be applied.

The above enumeration of possible forms of intervention is, of course, not exhaustive. Furthermore, the forms discussed are by no means mutually exclusive – although it is unlikely self-regulating systems will support all of them simultaneously. Apart from the overhead involved in designing and developing increasingly sophisticated interventions, there is the fundamental question of what can be achieved, given an existing adaptive system.

D.1.2.4 Overview of Implications

It may seem that given the proliferation of a wide range of modeling- and decision making- approaches in use today, and the fundamental differences between them, the preceding question can only be answered on a per-case basis. It is argued, however, that, at a high level of abstraction, there is one dimension that is by far the most important with respect to self-regulation: the level and granularity at which the internals of the modeling- and decision making- processes are exposed to the rest of the system. We will borrow the terms "white-box" and "black-box" to refer, respectively, to the case of the process internals being fully inspectable by the rest of the system, and the case of having no possibility for inspection at all. Further, "black-box" and "white-box" are to be understood as two fictional endpoints of a continuum, with

[43] For example, one way of decomposing adaptations in this manner is to break them down to primitive adaptation actions (see, e.g., [Paramythis and Stephanidis, 2005]).

increasing levels of inspectability and granularity leading from one to the other.

The factor most likely to determine where in this continuum a particular modeling- or decision making- approach belongs is the computational character of the algorithms that implement it. For instance, consider the case of a system that uses neural networks to associate dynamic model attributes with adaptive behaviors. Since the "internals" of the neural network do not have individual semantic value, even if they were to be exposed they would be of no use to the rest of the system. That system's decision-making approach would then lie at the "black-box" end of the spectrum. Conversely, consider a system that uses rule-based adaptation logic. As rules are distinct and, at least theoretically, possible to manipulate individually, they would result in the system's being classified as having a "white-box" decision making approach.

Figure 35 presents a high-level overview of how the inspectability and granularity of the modeling- and decision making- processes affect the self-regulation capabilities of an adaptive system. Of particular note are the following points:

- Self-evaluation capabilities are mainly dependent on the inspectability and granularity of a system's modeling approach.
- Intervention capabilities are mainly dependent on the inspectability and granularity of a system's decision making approach.
- Implementing self-regulation in systems with "black-box" modeling would effectively necessitate additional modeling performed at the second level adaptor, on the basis of direct user input.
- Implementing self-regulation with "black-box" decision making would require that the second level adaptor can "reverse" (or otherwise modify) the *potential effects* of adaptations, as the adaptations themselves are opaque.
- Implementing self-regulation in a model- and logic- agnostic manner is still possible, but requires that the second level adaptor: (a) can directly interpret direct user input; (b) supports a concept of context, based on that input; (c) supports a generic concept of self-evaluation along the same lines; and (d) applies second-level adaptations independently of the first-level adaptor.

D.1.3 Discussion

The previous sections have attempted to provide an overview of how self-regulation can be understood in the context of, as well as of the prerequisites it imposes on, modern adaptive systems. One of the several important topics that have not been discussed thus far, is how the system can be sure that the changes it observes on the interaction are attributable to a specific adaptation, while performing self-evaluation? This is a fundamental question to which the answer is, perhaps unsurprisingly: it can't! At its core, self-regulation is restricted to assessing whether system behaviors have the expected results, but there are only "extrinsic" ways for the system to ensure that these results were not side effects of other, entirely unrelated behaviors.

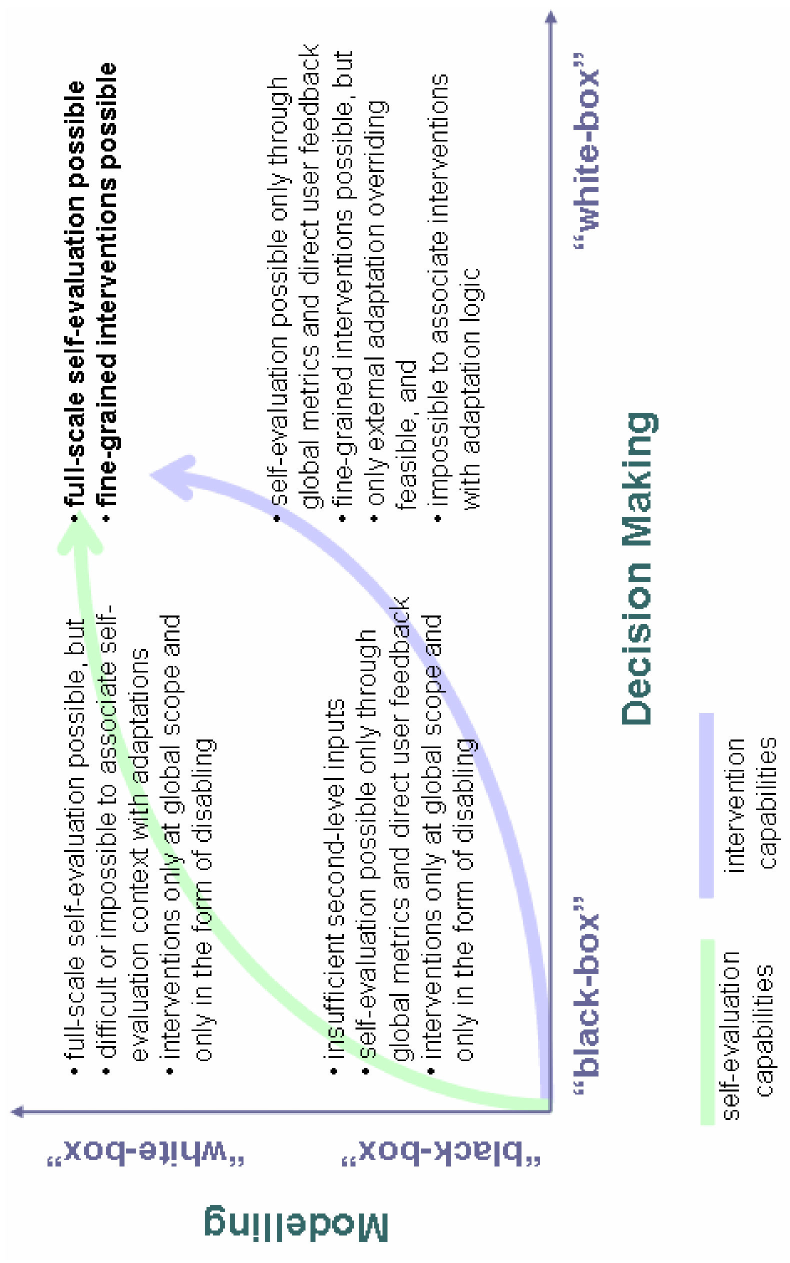


Figure 35: The effects of the exposed granularity of the modeling- and decision making- models on an adaptive system's self-regulation capabilities.

One such way is the establishment of "clusters" of collaborating self-regulating systems. Clustering, here, refers to the establishment of communication and coordination channels between the systems. The subject of collaboration is none other than the systems' aggregated "findings" in the second-level adaptation cycle. For instance, an adaptive learning system could communicate to its cluster that it has observed that a particular type of adaptation (e.g., link annotation) has not had the expected results (e.g., the user did not choose the annotated links over non-annotated ones) in some of the hosted courses; furthermore, the system could attach to that observation the attributes of the models describing the interaction state that were common in all these observations (e.g., that the user has considerable computer expertise, and that the user has prior knowledge of the subject domain). Other systems in the cluster could then refine (or challenge) the asserted observation, with their own. It is argued that the power of this approach lies with the fact that the validation of the first-level adaptive behavior happens in large scale, and is based on the statistical validity of contributed observations. Please note that, although the preceding example is of a negative observation, positive observations would not only be equally interesting, but also vital to the operation of the cluster.

This form of "sharing of experience" within clusters of self-regulating systems can result in a body of meta-knowledge regarding the systems' basic, or first-level adaptive behavior. This meta-knowledge is of value unto itself, as it would, in several cases, suffice to answer questions such as the one posed at the beginning of this section. It would also make it possible for newly-installed same-domain systems to take advantage of the accumulated "experience" of other peer systems, as this is expressed at the level of the cluster. Furthermore, creating clusters that support exchanges between systems operating within different contexts of use would enable us to derive more generalized knowledge, and perhaps even "discover" cross-application or cross-domain interaction patterns with relevance to adaptation. Finally, the knowledge accumulated within clusters could serve as the basis for adaptation models that would enable the development of grounded self-mediating systems.

Another important topic that merits our attention is the redefinition of the role of adaptation designers, as well as the more general role of human experts in relation to self-regulating systems. To start with, self-regulation demands that we revise the way in which we design adaptive systems. To date, the knowledge and rationale behind adaptation design may exist (although the literature indicates that sometimes common sense and intuition are the sole basis of designs), but is definitely not "codified" into the adaptive system. As discussed earlier, this knowledge now needs to be formalized and expressed as measurable "expectations" that the system assesses against. It is argued that although this requirement may imply additional overhead in the design of adaptive systems, it also has the potential of improving designs in the first place.

Apart from changing design practices, self-regulation calls upon human experts to undertake new roles in the adaptation process. Specifically, humans may now need to (occasionally) inspect the modifications effected on the system's behavior by the second-level adaptor and intervene when the system is evidently at fault. More interestingly, humans are called upon to semantically interpret the "findings" of self-regulating systems working in isolation or within clusters, or resolve conflicts in the latter case. Finally, the aforementioned "findings" have the potential to inform, or even serve as input, to the empirical evaluation of adaptive systems, which, in turn, can help improve both the first- and second- level adaptive behavior.

There are several questions of pivotal nature in the theoretical and practical employment of self-regulation that have not been addressed in this chapter. For instance, what is the role of the adaptive system's goal against which the evaluation should take place? Should the goal be fixed and concrete, or should the self-regulating adaptive system be able to deal with less concrete goals or goals that change over time? Are there clear limits between self-regulation and self-mediation? These and other questions need to be brought to the epicenter of discussion, as they are at least equally important for the adoption of self-regulation as the more "technical" issues discussed herein.

The final topic that we would like to touch upon in closing is the steps we need to make as a community to move closer to the establishment of self-regulation as a standard property of adaptive systems. It is argued that realistic path will go through the following milestones: (a) establishment of a comprehensive theoretical basis for self-regulation; (b) development of software frameworks that can provide basic self-regulation capabilities as an "add on" to existing adaptive systems; (c) experimentation and validation on systems with "white-box" modeling- and decision making- approaches; (d) accumulation and synthesis of experiences, towards a more broad dissemination of the involved theory and technologies.

D.1.4 Reported Work against the State-of-the-Art

Recent years have seen an increasing amount of interest in meta-adaptation in IAS. Brusilovsky [2003b] explicitly calls for additional research in the area of educational AHS, and the need for meta-adaptation is proclaimed by researchers in other areas and domains as well (see, e.g., [Alpert et al., 2003], [Ahmad, Basir & Hassanein, 2004]). Yet, despite the acknowledgement that meta-adaptation is important and a logical next step in the evolution of IAS, researchers are often reluctant to address it directly. An example of the attitude exhibited in many cases is [Francisco-Revilla & Shipman, 2003], where a framework for adaptive spatial hypermedia is introduced, and where the importance of meta-adaptivity is acknowledged, but not addressed directly in the framework itself. It is interesting to consider the reasons for this phenomenon, especially against the background of meta-adaptation being employed routinely in non-interactive system domains (e.g., autonomous agents [Tzafestas, 2005], software feedback systems [Cen, 1997], traffic light planning [Gershenson, 2005], self-organizing systems [Prokopenko, 2008], to name just a few)

Inarguably, one of the overarching difficulties in applying meta-adaptation in general, and self-regulation in particular, in the area of IAS is the indeterminism and ambiguity introduced by the human element. In domains other than IAS, meta-adaptation and self-regulation are based on comparatively semantically unambiguous evidence collected from the system's "world", on how the behavior of the system affects and is affected by its environment. This is not a trivial task by any means in most cases, but it pales in comparison to the task of finding ways to quantify the effects of adaptation on the users' mental model and trust of a system, their disposition towards it, their short- and long- term strategies in engaging with it, their affective state, etc. What renders this task especially challenging is that the observable evidence of all these effects is scarce and semantically ambiguous. Humans are the most complex adaptive systems known to us, and (changes in) behavior across individuals, or even in the same individual, cannot be readily and deterministically interpreted and attributed to specific factors. This poses significant hurdles in the design and implementation of self-evaluation, which is further hindered by the complexity of evaluating adaptation in the first place. Despite these obvious difficulties, some promising efforts have started emerging in the past few years.

Balík and Jelínek [2006], for example, iterate the in-principle arguments in favor of meta-adaptation in educational AHS, and propose that a meta-model be used as a basis for meta-adaptive systems. Meta-models, such as the one described in [Seefelder de Assis & Schwabe, 2004] are essential for a system to be able to reason about and modify its own adaptive behavior. Seefelder de Assis, Schwabe and Arraes Nunes [2006] propose an architecture, based on meta-models and ontologies, which they argue can be utilized for effecting meta-adaptation. Although the approach does indeed show potential, the modification of the system's adaptive behavior, as the authors describe it, is based on static rules, and there is no element of learning on the part of the system, nor any form of self-evaluation.

In contrast to the preceding efforts, [Vassileva & Bontchev, 2006] explicitly deals with dynamically altering adaptive behavior on the basis of models "learned" by observing user behavior. Specifically, the authors propose what they call a "self-adaptive" approach to hypermedia navigation in educational AHS, based on learner model characters. The meta-adaptation element in their proposed approach involves the establishment by the system itself of "working paths" through learning material, through the establishment of associations of elements in the learner model (such as learner style, goals/preferences and/or prior knowledge and shown performance) with the said paths. In the proposed approach, the system establishes these associations based on behavior of multiple learners going through the learning material, and, thus, the final models "converge" as increasing evidence from user behavior becomes available.

Ounaies, Jamoussi and Ghezala [2008] focus on the question of how the effects of adaptation can be quantified, so as to enable the self-evaluation aspect of meta-adaptation. The authors propose a specific measurement framework for adaptive web based educational system evaluation, which addresses a co-called "strategic level" of assessment of adaptation goals and benefits. Unfortunately, the measurement framework does not appear to be sufficiently well-defined at the moment. This is evidenced, for example, by the proposed "Adaptation Method Appropriateness Rate" metric, which is specified to be the ratio of "Number of adopted recommendations" by the "Number of total proposed adaptations", which is not readily applicable to adaptive systems other than recommender systems (and, even for them, the semantics of this ratio are not clear and depend on several system-specific factors in the author's opinion). It is hoped that the planned practical application of the measurement framework might bring progress in this otherwise promising line of work.

[Papanikolaou & Grigoriadou, 2004] describe an approach that is closely related to the concept of self-regulation as presented in this chapter. Specifically, the authors propose that a system dynamically adapt its behavior on the basis of a diagnosis process that includes the identification of specific measures of learners' observable behavior which are indicative of learners' learning style preferences. Emphasis in placed on the aforementioned measures (which would form the basis for what has been termed "metrics" in this chapter), and indicators that have been investigated in the literature for several learning style categorizations are listed: navigational indicators (number of hits on educational resources, preferable format of presentation, navigation pattern); temporal indicators (time spent in different types of educational resources proposed); performance indicators (total learner attempts on exercises, assessment tests).

In synthesis, the above brief overview of recent efforts appears to indicate that meta-adaptation in IAS is a research area that is drawing increasing amounts of attention and where formative work is taking place at the moment. Arguably, the work presented in this chapter provides the basic theoretical tools for approaching the subject in a generic, application domain-independent way, and considers not only technical aspects of the problem, but also the changing responsibilities of humans and systems, in their redefined roles as adaptation agents and recipients.

D.2 Self-regulated Adaptivity as a Design and Authoring Support Tool[44]

As already discussed in the previous chapter, the design and implementation of adaptive systems is a challenging process. Among the major historical factors behind this fact are: (a) the unavailability of open platforms / frameworks that can be used as the basis for implementing adaptivity; (b) a lack of implementation support tools; and, (c) the lack of hard, empirical evidence that can be used as input to the design of adaptivity. In recent years, each of the above areas has been addressed by research to a greater or lesser degree. For instance, on the framework side, one can now employ and build upon open-source frameworks such as AHA! [De Bra et al., 2002c]. On the implementation support side, a number of tools have emerged to support the declarative definition / composition of adaptive behavior; there are currently both tools that are delivery framework-specific (e.g., the AHA! authoring support tools [De Bra et al., 2005]) and ones that are largely framework-independent (e.g., MOT [Cristea & de Mooij, 2003]). Last but not least, great attention is being paid recently to the principled evaluation of adaptive systems, which progressively gives rise to a body of knowledge that can be directly applied in adaptation design.

Despite these recent achievements, however, there are still at least two aspects of the development process of adaptive systems, that are directly related to the aforementioned three dimensions, and are in need of additional research attention. Firstly, as adaptivity is becoming widespread, we are more often called upon to design adaptive system behavior in novel interactive contexts and application domains. There are simply cases where the design has to evolve from a basis comprising educated guesses, design decisions that draw upon intuition, and, sometimes, only remotely related empirically derived evidence. Secondly, the current generation of adaptive systems, although a considerable improvement over their static predecessors, still share a deficiency with them as far as interaction with humans is concerned: they have no means of reasoning about, or modifying in any way, their own adaptive behavior. A direct side effect of this is that any evolutions to the adaptation design must be brought about by the designers themselves. A typical lifecycle for such evolutionary steps involves the setting up of experiments with real end users, the analysis of results, the identification of required modifications in the system's adaptive behavior (e.g., changes in the

[44] This chapter is based on [Paramythis, 2006]. The work reported here has been supported in part by the Socrates-Minerva "Adaptive Learning Spaces" project (229714-CP-1-2006-1-MINERVA-M). Please refer to section "A.2.2 Work Context and Research Projects" on page 32 for additional information.

adaptation logic, or in the adaptation patterns), and the reformulation of those modifications for inclusion into the adaptive system.

We herein argue and provide an example of how self-regulated adaptivity (or, self regulation for short) can address, at least to some extent, both of the problems briefly outlined above. The rest of this chapter is structured as follows: The next section presents a worked out example of a system capable of adaptive presentation of hypermedia e-learning materials, and follows through the potential design process, as this would be manifested in the presence of self-regulated adaptivity. The subsequent section then goes on to analyze the presented example, point out the benefits potentially derived from applying the approach, and extract the requirements that such an approach would impose on the system. It further discusses the potential of using self-regulation as a specific form of meta-adaptivity to achieve the desired objectives.

D.2.1 Meta-adaptive System Design

This section presents a worked out example of a case study of the design of a meta-adaptive system. An important point to note is that, for the sake of simplicity, this section does not discuss one very important aspect of the prerequisite infrastructure, namely the approach and steps required to effect self-evaluation, which is an integral part of self-regulated adaptive system behavior. Instead, a "deus ex machina" view of the respective process is adopted, and further discussion is deferred till section 3.

D.2.1.1 The Case Study

The basis of our exemplary system design is a simple, yet popular, adaptive function in adaptive hypermedia systems: the annotation of links within learning content.

For our needs we will assume a system that exhibits characteristics common to a large range of adaptive systems in the field (e.g., AHA! [De Bra et al., 2002c], or NetCoach [Weber et al., 2001]). The system's most important features can be summarized as follows:

- The system's domain model is a small, course-specific ontology comprising learning concepts and semantic relations between these concepts (e.g., "prerequisite-of"). The domain model also has explicit representations of the modules / pages that make up the actual learning content, annotated with the semantic relation between each module and the respective concept (e.g., "explains", "provides-examples-for", "tests-knowledge-of", etc.)
- The system's user model is a simple overlay model (over the domain model), with a small number of discrete (and possibly mutually

exclusive), user-specific "states" with respect to each of the concepts in the domain model (e.g., "has-not-seen", "has-seen", "has-learned", etc.)

- Individual user models are updated on the basis of directly observable user activities (e.g., visiting the module that "explains" a concept, responding to test items)
- Based on the current state of an individual user model, the system can decide on recommendations regarding the future visits of different modules / pages. Once again, these recommendations can be assumed to belong to a small set of discrete, mutually exclusive, module-centric ones (e.g., "ready-to-learn"). It is exactly on the basis of these recommendations that link annotation is being considered in the context of our example.
- Although this is not pertinent to the ongoing discussion, one can assume for completeness that in the adaptive system at hand adaptation logic is expressed through simple adaptation rules, such as in the case of [Stephanidis, Paramythis et al., 2004].

Further to the above, we will assume that the user model contains other user attributes, some of them explicitly provided by the user (e.g., demographic data), and others inferred from user activity. As an example for the second category, consider "expertise", which refers to the user's familiarity with the system and is updated through a function that takes into account factors such as how often the user enters the system, how long the user's sessions are, "coverage" of system facilities in typical usage patterns, etc. (might be over more than one courses).

The design question at hand is whether to present users with the system's recommendations with respect to links present in a page, and, if yes, what is the best way to annotate links to convey the semantics of the system's recommendations. Although there is a considerable body of research on this question, for the purposes of this example, we will assume that the system's designer has no empirical evidence to support the considered design alternatives. The alternatives themselves are encapsulated in five different strategies as far as link annotation is concerned (see also Figure 36):

Strategy A: *No annotation.* This can be considered the base-line strategy, and would simply involve not exposing the user to the system's recommendations. In our very simple example, this might be identical to the non-adaptive version of the system.

Strategy B: Annotation using different link colors. In this strategy different colors are used directly on the links to signify system recommendation. From a human-computer interaction perspective, this strategy can be seen as one that might require some learning on the part of the users, but would add as little clutter to the page as possible. One potential problem is that links colors already have well-defined meanings (e.g., whether a link has been visited) that may clash with the strategy at hand.

Strategy C: Annotation using bullets of different colors. This is very similar to strategy B above, with the exception that the colors are applied externally to the links. Annotations (i.e., the bullets) are dynamically added to the document.

Strategy D: Annotation using custom icons. A variation of strategy C above, with the bullets replaced by icons that are intended to carry more semantic information, in effect embodying the system's recommendation in a pictographic manner. The rationale for their use is that they would presumably be more readily recognizable by novice users. On the other hand, they might add a lot more visual noise to a page.

Strategy E: Link hiding. This strategy involves hiding (although not disabling) links (see [Brusilovsky, 1996]), for which the system's recommendation is that the user is not yet ready to visit them. This strategy is intended as a more direct attempt to guide the user as compared to the previous ones.

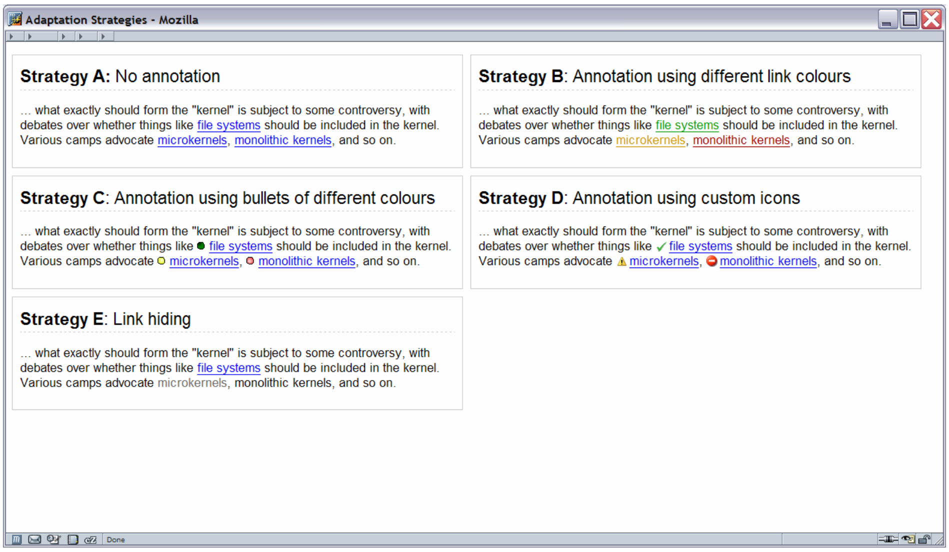


Figure 36: Examples of adaptive link annotation design strategies.

Given the strategies above, the design question at hand is which one(s) to use, and for which users or usage scenarios. Note that the strategies, as formulated herein, are not necessarily mutually exclusive (e.g., C cannot be combined with D, but both C and D can be combined with E). Also note that it would be desirable to identify situations (e.g., increases in the user's "expertise", as recorded in the respective attribute of the user model) that might justify a transition from one strategy to another.

It should finally be explicitly pointed out that the objective of the design process *is not* to arrive at a single alternative that can be used for all users (which, in itself, would defeat the purpose for employing adaptivity in the first place). The primary assumption behind the design is, actually, that no alternative will fit all users under all circumstances, and the desideratum is to identify the cases where a specific alternative would be the best option for a given user and context of use.

D.2.1.2 Evolution of Adaptive Behavior

The designer starts out with no evidence about when and under what conditions to use each strategy, or whether, indeed any one strategy is "better" than all the rest. Each strategy has obvious trade-offs as far as flexibility and user control over the navigation process is concerned. The designer's goal however, is clear: students should encounter concepts that they are not "ready" for as little as possible, and this should be achieved with the least possible restrictions on interaction / navigation.

As already mentioned, this section does not go into a discussion of how self-evaluation might be effected in the example system, but it assumes that self-evaluation does takes place, and that its results drive the subsequent design iterations. Continuing with the example introduced above, we will look at three potential iterations that the design process could have gone through. Please note that these iterations are not "normative", and are meant mainly to facilitate the discussion of how different types and levels of self-evaluation results can influence the design process. Although the proposed approach is by nature iterative, there is neither any ground, nor any motivation to constraint the number of iterations required; this is a design decision that needs to be made on a per case basis.

Iteration 1: "Tabula rasa"

The first step of the design would involve the encoding of the strategies as sets of adaptation actions – something that could be done in "hard-coded" programmatic logic, or, preferably in a declarative fashion (e.g., as in [Paramythis & Stephanidis, 2005]).

Since the designer has no evidence regarding the applicability of the different strategies, however, these cannot be directly assigned to adaptation logic (e.g., one cannot, yet, create adaptation rules that would link strategies to attributes in the user model). The system would then need to be able to recognize these strategies and apply them (separately or combined) in more or less a trial-and-error fashion.

The design information that already exists, and can be conveyed to the system, is which strategies are mutually exclusive, and which ones can be applied in combination. Using the information in Table 6, the following twelve combinations can be identified: (i) A, (ii) B, (iii) B+C, (iv) B+D, (v) B+E, (vi) B+C+E, (vii) B+D+E, (viii) C, (ix) C+E, (x) D, (xi) D+E, (xii) E.

Table 6. Considered adaptation strategies and their compatibility; ● signifies that the strategies are compatible, and **X** signifies that the strategies are mutually exclusive.

Strategies	A. No annotation	B. Colored links	C. Colored bullets	D. Custom icons	E. Link hiding
A. No annotation		X	X	X	X
B. Colored links			●	●	●
C. Colored bullets				X	●
D. Custom icons					●
E. Link hiding					

Given these constraints, and a suitably encoded, computable representation of the design goal stated earlier (i.e., to help guide users to modules / pages that are best suited to their current level of knowledge), the system is then ready to undergo the first round of user testing.

Iteration 2: Selecting, categorizing and prioritizing (combinations of) strategies

We will assume that the results of the first round of testing do not yet suffice for building a comprehensive body of adaptation logic to guide the system's adaptive behavior. They do, however, provide enough evidence for the following:

- Eliminating strategies (and combinations thereof) that do not seem to meet the desired design goal under any circumstances. In the context of the ongoing example this might include, for instance, strategy B and all its combinations (presumably because changing links' colors is confusing for users).
- Categorizing and providing a tentative "ranking" of the remaining combinations, based, respectively, on their design / interaction semantics, and on the rate of success they have exhibited during the first round of testing; of course, other factors (for instance, how restrictive or obtrusive alternatives are) can be used as well.

The aforementioned categorization and ranking process, might result in something like the following in the case of our example:

Category I: Includes only strategy A and corresponds to absolute freedom in navigation, with no system assistance / guidance whatsoever.

Category II: Includes the uncombined strategies C and D and corresponds to absolute freedom in navigation, but this time with explicit system assistance / guidance.

Category III: Includes all combinations of strategy E and corresponds to the application of restrictions on navigation, to enforce a path through the learning material.

The above categorization and ranking is, obviously, only one of several possibilities. It does, nevertheless, serve to demonstrate the following points:

- Although it is a ranking, it is not obvious in which "direction" it should be applied. For example, should the system start with the most "liberal" (in terms of navigation freedom) category and move to the more "restrictive" one when attempting to satisfy the overall adaptation design goal? Or should it apply the categories in reverse to accommodate, for instance, increased user familiarity with the system, or to compensate for inferred user frustration with the navigation constraints?
- Applying such a ranking incorporates two concepts that may need to be extricated and made explicit: the concept of the "default" category of strategies that might be applicable for a new user; and the concept of a "fallback" category that gets applied when none of the available categories / strategies has the desired effect.

For our example, we will assume that the designer has opted to use the ranking in the order presented above (i.e., "liberal" to "restrictive"), and to let the default and fallback categories be the first and last ones respectively. With these additional constraints, the system would be ready for a second round of user testing.

Iteration 3: Binding strategies to concrete adaptation logic

The introduction of additional structure in the adaptation design space effected in the previous iteration, along with more results from user testing based on that structure, can be expected to finally provide detailed enough results to start building more concrete and comprehensive adaptation logic around the alternative strategies.

According to results from related research in the literature, this iteration might result in user model-based adaptation logic along the following lines:

- For novice system users, as well as for users unfamiliar with the knowledge domain of the material, the more restrictive category (III) of strategies would be applicable.

- Within Category III, a ranking between strategies would be possible, such as: (i) strategy E – link hiding, no explicit recommendations by the system, (ii) combination D+E – link hiding and icons to "explain" the rationale between the provided guidance, and (iii) combination C+E – same as previous case, but with less visual clutter on the page (intended for more experienced users).
- Category II (uncombined strategies C or D) would be reserved for users who seem to be sufficiently familiar with the system and the recommendation mechanism. If there is evidence of user confusion or ill effects of the user's increased navigation freedom (e.g., often-occurring negative test results for concepts encountered before their prerequisite concepts have been learned), the system should fall back to using Category III.
- Category I should be reserved for users who already have some familiarity with the knowledge domain, or exhibit behavior indicating intention to circumvent constraints applied on their navigation freedom (e.g., using the course outline to move between modules not interlinked).
- Etc.

The above adaptation logic is only exemplary in nature and might differ significantly from the actual results one might get with a specific system and learning material. It can, however, serve as a basis for the discussion in the forthcoming sections. Also note that, although the example case study is ending here, there is no reason why in real-world settings this would be the last design iteration. In fact, it is quite imaginable that the updated adaptation design might be put to the test again, to achieve even more refined adaptation logic, or more fine-grained adaptation strategies.

D.2.2 Adaptive System Design Revisited

The preceding section has put forward a scenario of how the adaptive behavior of a system could be designed, or "evolved" in a step-wise manner, taking advantage of meta-adaptive facilities. This section will fill in some of the intentional "gaps" in the presented scenario, formulating them as requirements posed by the introduction of meta-adaptivity. Before doing so though, let us first examine these facilities being utilized behind the scenes, and their effects on the design process.

D.2.2.1 New Possibilities

To start with, the basis of the design iterations has been the derivation of new knowledge regarding the suitability of specific adaptive behaviors for different users (or contexts of use) given the overall design goal / self-regulation metrics. This knowledge, in its simplest form, is derived by applying alternative (combinations of) adaptation strategies, and assessing the extent to which the self-regulation metrics are satisfied, always in connection to the current user's model. Knowledge derivation, then, is achieved by analyzing all recorded cases where a particular strategy has had similar results, and identifying common user model attributes among the respective users. This is the core of the "learning" facilities in the context of self-regulated adaptivity, and their output could be expressed in various forms, including for instance as preliminary adaptation logic, intended to be reviewed, verified and incorporated into the systems by the designers. Although rather straightforward, the above step may already suffice to provide valuable input to the design process. For example, it should be capable to identify strategies that are not suitable for any (category of) users, in any context of use. This was assumed to be the case in the elimination of strategy B and all its combinations in the previous section.

A second set of capabilities alluded to in the previous section is the categorization, or "clustering" of adaptation strategies, as well as their "ranking". Categorization can take place mainly along two dimensions: (a) The system can try to identify strategies that have similar effects with respect to the self-regulation metrics, given sufficiently similar user models; the output of this process would be a provisional clustering of strategies, based on their "cause and effect" patterns. (b) The system can try to identify the differentiating subsets of user models that render some strategies more effective than others. These dimensions give, respectively, two semantically rich measures of similarity and differentiation of adaptation strategies. When sufficient meta-data about the user model itself is available, the system can combine that with the measures to provide provisional rankings of alternatives within categories.

Before continuing it is important to note that the example in the previous section, as well as the analysis in this section, only assume three types of analytical assessment capabilities on the part of a self-regulating adaptive system. Although these are by far not the only ones possible, they are already adequate for the type of design support put forward herein.

D.2.2.2 New Requirements

The fact that self-regulated adaptivity introduces new requirements becomes already evident in the first iteration of the design described above. Specifically, the first iteration made the following two fundamental assumptions:

- That adaptation strategies may be represented independently from the adaptation logic that drives them.

- That adaptation strategies, potentially expressed as sets of adaptation actions, be applicable in combination.
- That there may exist a representation of one or more adaptation "goals" that drive both the process of self-evaluation and the selection / application of strategies.

Of the above requirements, it might seem that the first two would be relatively easy to achieve in existing adaptive systems. This could be done, for example, respectively through the introduction of adaptation rules that randomly select one of a set of alternatives, and through the exhaustive representation of all possible combinations. Such an approach, however, would be incomplete in the sense that the adaptive system has no "understanding" of the alternative adaptation strategies, a fact which precludes the satisfaction of the third requirement, namely the employment of self-evaluation.

The second design iteration has also introduced additional requirements with respect to the adaptation engine. Specifically, it was implied that the system is capable of maintaining and employing a type of ranking amongst (combinations of) adaptation strategies, in a way that still does not associate the later with concrete, user model-based adaptation logic. Although perhaps inherent in the capacity to support such a ranking, the concepts of a "default" and a "fallback" strategy also merit individual mention. This is especially true in the case of the fallback strategy, which may often be a mid-ranked alternative, rather than one at either extreme (this depends primarily on the factors taken into account to produce the ranking in the first place).

Finally, implied practically in all iterations, but made more explicit in the third one, is the fundamental requirement of the system being able to perform self-evaluation. As already mentioned, the concept of self-evaluation requires: (a) that the system be "aware" of the existence of alternative adaptive behaviors, and (b) has some way of assessing the extent to which these alternatives, when used, satisfy the design requirements. The first item points to the need for having an explicit representation of meta-data about the alternatives in the system, along with any semantic relations and constraints between them. The second item highlights the requirement that the "intention" behind an adaptive behavior (or set of behaviors) be expressed in a computable / measurable manner, so that the system can undertake the related assessment tasks.

"Armed" with these capabilities, the system can then proceed to analyze the commonalities / differences between the models of users for whom a particular strategy (or category of strategies) has proven successful / unsuccessful. This analysis would most likely be statistical in nature, and could draw upon well-established techniques in data mining. There is, however, a non-conventional requirement in this case: that when analyzing, the system must have access to prior states of a user's model. This is necessary because observations of the system as to the employment of specific (categories of) strategies for a given user, are inextricably linked to that user's individual model *at the time that the strategy was employed*. This is further obviated by the fact that appropriate strategies for users may change as their user model evolves over time. The re-stated requirement, then, is that the system must maintain a "history" of changes in the user models (what the history comprises and how its maintenance can be automated are discussed in the next section).

Finally, the example presented contains a simplification that we need to address. Specifically, in the example, the design iterations are linear and incremental, i.e., each iteration is based on the findings of the previous one, and adds more detail to the system's adaptation meta-data. In the real world, however, this may not always be the case. For instance, it might be necessary to consider entirely new strategies (e.g., adding link annotation through tooltips as an additional possibility) at later stages of the design. What this necessitates is that the system be able to function at two or more levels of granularity simultaneously, where each level represents a different degree of detail in the binding between adaptation logic and alternative adaptive behaviors.

D.2.2.3 Deus Ex Machina – Can Self-regulation Be It?

There are two overarching questions that have still not been addressed: (a) Why introduce self-regulated adaptivity at all? Can't the above design activities be supported through an entirely human-facilitated approach? and, (b) How does one achieve the level of self-regulated adaptivity proposed in the preceding sections?

To start with, let us consider what meta-adaptivity, as implemented through self-regulation, brings to the "design table". Firstly, it allows us to specify and test with end users a potentially large number of alternative adaptive behaviors (or combinations). A human-facilitated approach would require the generation of several instances of the adaptive system (in the worst case, one such instance would be required for each alternative / combination), as well as the overhead of administering tests involving real users for each case. Additionally, it is questionable whether such an approach would allow for the seamless transition between different adaptation strategies within one interaction session. The second benefit we derive from the employment of meta-adaptivity is that the generation of new adaptation knowledge (which, in turn, is subsequently expressed as adaptation logic to drive the system behavior) takes place *within* the system, making use of, and adding to, the adaptation meta-data. Since this process can be automated, it makes little sense to repeat the necessary analysis and encoding steps on a per-case basis. And last but not least, the employment of meta-adaptivity brings forth an entirely novel opportunity: a meta-adaptive system can continue improving itself (e.g., by accumulating adaptation knowledge / evidence, and using it to revisit its own adaptation logic) even without human intervention.

Let us now move over to the second of the questions posed at the beginning of this section, namely: how does one achieve the level of meta-adaptivity described herein. It is argued that self-regulation is sufficient for satisfying the requirements posed thus far. Since a detailed description of self-regulation was provided in the preceding chapter, we will simply focus on the basic premises of this particular type of meta-adaptivity.

The most important facets of self-regulated adaptivity are: (a) it explicitly accounts for alternative adaptive behaviors; (b) self-evaluation in the context of self-regulation is based on metrics that relate adaptive system behavior with changes in the user models (or, potentially, other dynamic models maintained by the system); (c) it entails apparent learning on the part of the system; and (d) it does *not* require that the system modify existing, or devise new behaviors on the fly. Let us address each of the requirements identified in the previous section in more detail to examine the degree to which they are satisfied within the proposed approach.

Explicit representation of alternatives: This is one of the basic premises of self-regulated adaptivity. Specifically, a self-regulating system is capable of maintaining sets of behaviors that are applicable to different users / context of use, and dynamically switch between them on the basis of their "success".

Explicit representation of adaptation goals / objectives: Again, this is one of the fundamental building blocks of self-regulation. Representation in this case is achieved through the formulation of metrics, i.e., computable quantities derived from the current or historical values of attributes in the system's dynamic or static models (the user model being the primary source).

Supporting categorization and ranking of alternatives: Although not an intrinsic requirement for self-regulation, this capability should be relatively easy to implement in a compliant system. Categorization, for instance, can be supported through the application of the primary self-regulating capabilities at different levels of granularity. Ranking on the other hand can be implemented as an algorithm for selecting among alternatives within one level of granularity. Support for "default" and "fallback" strategies can be attained in a similar manner.

Self-evaluation and creation of adaptation knowledge: Again, self-regulation is "at home" with these requirements, since, even in its more basic incarnations, it involves the capability to on-the-fly assess the degree to which the currently employed alternatives satisfy the metrics encapsulating the design objectives, and take corrective actions as necessary. Furthermore, a self-regulating system can infer, over time, the applicability of behaviors to different (states of) user models and contexts of use. What's more, self-regulating systems can even validate the body of adaptation logic present and apply basic improvements over it without any human intervention (e.g., by not using adaptation rules that do not have the desired effects under any circumstances), or provide the evidence required for improvements to be made by humans.

Maintenance of "history" of model changes: As described in detail in the previous chapter, access to historical values of dynamic models is essential in enabling the learning part of a self-regulating system. What is interesting to note is that this requirement can be satisfied: (a) automatically, and (b) without maintaining a full record of all changes made to models. This of course implies that the system can "decide" what to place in the history. This information can be extracted, in the case of self-regulation, from the metrics used for self-evaluation. Specifically, provided that the system can "understand" the metrics used, it can identify all model attributes that are involved in any of the known metrics it handles, and record changes to them. This way, the history can be created in both an entirely automated way, and only cover those parts of a model that will be relevant to later learning processes.

D.2.3 Discussion

This chapter has presented a case for the use of meta-adaptivity as a facilitator in the design of adaptive systems. It has furthermore argued that the requirements introduced by the employment of meta-adaptivity can be met with only moderate complexity through self-regulation.

The applicability of the proposed approach is of course not universal: it requires, for example, that an adaptive system (or infrastructure) is already operational. It is also mainly intended for cases where there exist several alternative adaptive behaviors, with little or no empirical evidence as to their suitability for different categories of users, or different "states" of a single user.

Another interesting question that has not been addressed here concerns the additional overhead that the proposed approach imposes on the designers / authors of adaptivity. Although the space available does not allow for a full treatment of the topic, it may be of value to cursorily outline the trade-offs: On the one hand, this approach requires that authors spend additional time in authoring adaptation strategies and providing the system with metadata for using and managing them; furthermore, it requires that designers review and, where necessary, validate the findings derived by the system through self-evaluation. On the other hand, some of the preceding activities are arguably ones that would need to be undertaken anyway in the context of iterative design cycles aimed at establishing and improving adaptive system behavior; an exception to this would be the metadata describing the adaptation strategies themselves, however the effort associated with them may be well justified by the more active role the system can play in facilitating aspects of the evolutionary process.

In closing, it is important to briefly go over a few additional characteristics and constraints of the proposed approach. To start with, self-regulation is by no means a way to forego user studies, but rather an exploratory yet structured tool to employ in conducting them. Secondly, the approach, as put forward in this chapter, is targeted to the design of adaptivity: abrupt transitions between potentially substantially different strategies might be unacceptable for a deployed system; as a result, in such settings, self-regulation may need to be further constrained (or limited to the accumulation of knowledge, but not permitted to effect changes in the system's adaptive behavior on the basis of the new knowledge). Thirdly, whether used solely in the design stage, or retained in the final system, self-regulation must be applied with care, as, by nature, it poses an even greater "threat" to traditional usability qualities of interactive systems (e.g., predictability) than traditional forms of adaptivity.

Within the confines discussed above, it is argued that self-regulated adaptivity represents not only the next logical step in the evolution of adaptive systems, but also a potentially irreplaceable tool in their design.

BIBLIOGRAPHY

[Ahmad, Basir & Hassanein, 2004] Ahmad, A.-R., Basir, O., & Hassanein, K.: Adaptive User Interfaces for Intelligent E-Learning: Issues and Trends. In Proceedings of the Fourth International Conference on Electronic Business (ICEB2004), Xiyuan Hotel, Beijing, China, December 5-9, 2004, pp. 925-934. Academic Publishers/World Publishing Corporation, 2004.

[Andreadis et. al., 1998] Andreadis, A., Giannetti, L., Marchigiani, E., Rizzo, A., Schiatti, E., Tiberio, M., Pentila, M., Perala, J., Leikas, J., Suihko, T., Emiliani, P.L., Bini, A., Nill, A., Sabbione, A., Sfyrakis, M., Stary, C., & Totter, A.: Global evaluation of the experiments. ACTS AC042 AVANTI Project, Deliverable DE030, August 1998.

[Angelides, Mylonas & Wallace, 2009] Angelides, M.C., Mylonas, P., Wallace, M. (eds.): Advances in Semantic Media Adaptation and Personalization, Volume 2. Auerbach Publications, Boca Raton, FL, USA, 2009.

[Alpert et al., 2003] Alpert, S.R., Karat, J., Karat, C.M., Brodie, C., & Vergo, J.G.: User Attitudes Regarding a User-Adaptive eCommerce Web Site. *User Modeling and User Adapted Interaction*, 13 (4), 373-396.

[Ashman, Brailsford & Brusilovsky, 2009] Ashman, H., Brailsford, T., & Brusilovsky, P.: Personal Services: Debating the Wisdom of Personalisation. In Proceedings of Advances in Web Based Learning – ICWL 2009, 19-21 August 2009, RWTH Aachen University, Aachen, Germany, pp. 1-11. (*Lecture Notes in Computer Science*, Volume 5686/ 2009). Springer, Berlin - Heidelberg, 2009.

[Ardissono & Goy, 1999] Ardissono, L., & Goy, A.: Tailoring the interaction with users of electronic shops. In Kay, J. (ed.), Proceedings of the 7th International Conference on User Modeling (UM99), pp. 35-44. Springer, Wien - New York, 1999.

[Aroyo et al., 2007] Aroyo, L., Stash, N., Wang, Y., Gorgels, P., & Rutledge, L.: CHIP demonstrator: Semantics-driven recommendations and museum tour generation. In Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., & Cudre-Mauroux, P. (eds.), Proceedings of the 6th International Semantic Web Conference, 2007, pp. 879-886. Springer, Berlin – Heidelberg, 2007.

[Balabanovic & Shoham, 1997] Balabanovic, M., & Shoham, Y.: Fab: content-based collaborative recommendation. *Communications of the ACM*, 40(3), 66-72.

[Balík & Jelínek, 2006] Balík, M., & Jelínek, I.: Modelling of Adaptive Hypermedia Systems. Proceedings of the International Conference on

Computer Systems and Technologies (CompSysTech'06), University of Veliko Tarnovo, Bulgaria, 15-16 June 2006. [On-line] Available at: http://ecet.ecs.ru.acad.bg/cst06/Docs/cp/sV/V.8.pdf

[Balík & Jelínek, 2008] Balík, M., & Jelínek, I.: Towards Data Interoperability in Adaptive Environments. In Richards, G. (ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2008, pp. 2480-2485. AACE, Chesapeake, VA., 2008.

[Berners-Lee, Hendler & Lassila, 2001] Berners-Lee, T., Hendler, J., & Lassila, O.: The Semantic Web. *Scientific American*, May, 2001.

[Benyon, 1993] Benyon, D. R.: Adaptive Systems: A Solution to Usability Problems. *User Modeling and User-Adapted Interaction*, 3(1), 65-87.

[Benyon, 1997] Benyon, D. R.: Intelligent Interface Technology to Improve Human-Computer Interaction (Tutorial). In: HCI International '97. San Francisco, USA.

[Boyle & Encarnacion, 1994] Boyle, C., & Encarnacion, A. O.: Metadoc: An Adaptive Hypertext Reading System. *User Modeling and User-Adapted Interaction*, 4(1), 1-19.

[Brown et al., 2006] Brown, E., Brailsford, T., Fisher, T., Moore, A., & Ashman, H.: Reappraising cognitive styles in adaptive web applications. In Proceedings of the 15th international conference on World Wide Web, pp. 327-335. ACM, Edinburgh, Scotland, 2006.

[Browne, Norman & Adhami, 1990] Browne, D., Norman, M., & Adhami, E.: Methods for Building Adaptive Systems. In Browne, D., Totterdell, P., & Norman, M. (eds.): *Adaptive User Interfaces*, pp. 85-129. Academic Press, London, 1990.

[Browne, Norman & Riches, 1990] Browne, D. P., Norman, N., & Riches, D.: Why Build Adaptive Systems? In Browne, D., Totterdell, P., & Norman, M. (eds.): *Adaptive User Interfaces*, pp. 15-57. Academic Press, London, 1990.

[Browne, 1993] Browne, P. D.: Experiences form the AID project. In Schneider-Hufschmidt, M., Kühme, T., & Malinowski, U. (eds.): *Adaptive User Interfaces: Principles and Practice*, pp. 69-78. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1993.

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

[Brusilovsky, 1998] Brusilovsky, P.: Methods and techniques of adaptive hypermedia. In Brusilovsky, P., Kobsa, A., & Vassileva, J. (eds.), *Hypertext and Hypermedia*, pp. 1-43.Kluwer Academic Publishers, Dordrecht, 1998.

[Brusilovsky, 2001] Brusilovsky, P.: Adaptive hypermedia. *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue, 11(1/2), 87-110.

[Brusilovsky, 2003a] Brusilovsky, P.: Developing Adaptive Education Hypermedia Systems: From Design Models to Authoring Tools. In Murray, T., Blessing, S., & Ainsworth, S. (eds.), *Authoring Tools for Advanced Technology Learning Environments*, pp. 377-410. Kluwer Academic Publishers, Dordrecht, 2003.

[Brusilovsky, 2003b] Brusilovsky, P.: Adaptive navigation support in educational hypermedia: the role of student knowledge level and the case for meta-adaptation.*British Journal of Educational Technology*, 34(4), 487–497.

[Brusilovsky & Eklund, 1998] Brusilovsky, P., & Eklund, J.: A Study of User Model Based Link Annotation in Educational Hypermedia. *Journal of Universal Computer Science*, 4(4), 429-448. [On-line] Available at: http://www.jucs.org/jucs_4_4/a_study_of_user

[Brusilovsky & Henze, 2007] Brusilovsky, P., & Henze, N.: Open corpus adaptive educational hypermedia. In: Brusilovsky, P., Kobsa, A., & Nejdl, W. (eds.), *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 671-696. LNCS, vol. 4321. Springer, Berlin - Heidelberg, 2007.

[Brusilovsky, Kobsa & Nejdl, 2007] Brusilovsky, P., Kobsa, A., & Nejdl, W. (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer Verlag, Berlin – Heidelberg, 2007.

[Brusilovsky & Millán, 2007] Brusilovsky, P., & Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., & Nejdl, W. (eds.), *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 3-53. LNCS, vol. 4321. Springer, Berlin - Heidelberg, 2007.

[Brusilovsky & Nijhavan, 2002] Brusilovsky, P., & Nijhavan, H.: A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities. In Proceedings of the World Conference on E-Learning in Corp., Govt., Health., & Higher Ed. 2002(1), pp. 154-161. AACE, 2002. [On-line]. Available at: http://www.aace.org/dl/index.cfm/fuseaction/View/paperID/9357.

[Brusilovsky & Pesin, 1998] Brusilovsky, P., & Pesin, L.: Adaptive navigation support in educational hypermedia: An evaluation of the ISIS-Tutor. *Journal of Computing and Information Technology*, 6(1), 27-38.

[Brusilovsky, Karagiannidis & Sampson, 2001] Brusilovsky, P., Karagiannidis, C., & Sampson, D.: The Benefits of Layered Evaluation of Adaptive Applications and Services. In Weibelzahl, S., Chin, D., & Weber, G. (eds.), Empirical Evaluation of Adaptive Systems. Proceedings of workshop held at the Eighth International Conference on User Modeling in Sonthofen, Germany, July 13th 2001 (pp. 1-8). Pedagogical University of Freiburg, Freiburg, 2001.

[Brusilovsky, Kobsa & Vassileva, 1998] Brusilovsky, P., Kobsa, A. & Vassileva, J. (eds.): *Adaptive Hypertext and Hypermedia*. Kluwer Academic Publishers, Dordrecht, 1998.

[Campbell et al., 1991] Campbell, R.H., Islam, N., Johnson, R., Kougiouris, P., & Madany, P.: Choices, Frameworks and Refinement. In Cabrera, L.-F., Russo, V., & Shapiro, M. (eds.), Proceedings of the 1991 International Workshop on Object Orientation in Operating Systems, Palo Alto, California, USA, pp. 9-15. IEEE Computer Society Press, Los Alamitos (CA, USA), 1991.

[Catledge & Pitkow, 1995] Catledge, L., & Pitkow, J.: Characterizing browsing strategies in the World Wide Web. In: Proceedings of the Third International World Wide Web Conference. April 10-14, 1995, Darmstadt, Germany. [On-line]. Available at: http://www.igd.fhg.de/archive/1995_www95/proceedings/papers/80/userpatterns/UserPatterns.Paper4.formatted.html

[Cen, 1997] Cen, S.: A Software Feedback Toolkit and its Application in Adaptive Multimedia Systems. PhD Dissertation, Oregon Graduate Institute of Science and Technology, 1997.

[Chieko & Lewis, 1998] Chieko, A., & Lewis, C.: Home page reader: IBM's talking web browser. In Proceedings of Closing the Gap Conference 1998.

[Chignell et al., 1989] Chignell, M. H., Hancock, P. A., & Loewenthal, A.: An Introduction to Intelligent Interfaces. In Hancock, P. A., & Chignell, M. H.: *Intelligent User Interfaces: Theory, Research and Design*, pp. 1-26. Elsevier Science Publishers, Amsterdam, 1989.

[Collins & Gentner, 1987] Collins, A., & Gentner, D.: How people construct mental models. In Holland, D., & Quinn, N. (eds.), *Cultural models in language and thought*, pp. 243-268. Cambridge University Press, 1987.

[Cote-Munoz, 1993] Cote-Munoz, A. H.: AIDA: An Adaptive System for Interactive Drafting and CAD Applications. In Schneider-Hufschmidt, M., Kuhme, T., Malinowski, U. (eds.): *Adaptive User Interfaces: Principles and Practice*, pp. 225-240. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1993.

[Cristea & Calvi, 2003] Cristea, A., & Calvi, L.: The Three Layers of Adaptation Granularity. In Brusilovsky, P., Corbett, A., & de Rosis, F. (eds.), Proceedings of the 9th International Conference on User Modeling, UM 2003, Johnstown, PA, USA, June 22-26, 2003, pp. 4-14 (*Lecture Notes in Computer Science*, Volume 2702 / 2003). Springer, Berlin - Heidelberg, 2003.

[Cristea & de Mooij, 2003] Cristea, A.I., & de Mooij, A.: Adaptive Course Authoring: My Online Teacher. ICT'03 (International Conference on

Telecommunications), Papeete, French Polynesia, IEEE, IEE, ISBN: 0-7803-7662-5, pp. 1762-1769.

[De Bra & Stash, 2002] De Bra, P., & Stash, N.: AHA! A General-Purpose Tool for Adaptive Websites. In Proceedings of the 11th International World Wide Web Conference, Honolulu, Hawaii, USA, 7-11 May 2002. [On-line]. Available at: http://www2002.org/CDROM/poster/168/index.html

[De Bra et al., 2002a] De Bra, P., Aerts, A., & Rousseau, B.: Concept Relationship Types for AHA! 2.0. Proceedings of World Conference on E-Learning in Corp., Govt., Health, & Higher Ed. 2002(1), pp. 1386-1389. AACE, 2002. [On-line]. Available at: http://www.aace.org/dl/index.cfm/fuseaction/View/paperID/9568

[De Bra et al., 2002b] De Bra, P., Aerts, A., Smits, D., & Stash, N.: AHA! Version 2.0, More Adaptation Flexibility for Authors. In Proceedings of the AACE ELearn'2002 conference, October 2002, pp. 240-246.

[De Bra et al., 2002c] De Bra, P., Aerts, A., Smits, D., & Stash, N.: AHA! The Next Generation. In Blustein, J. (ed.), Proceedings of the thirteenth ACM conference on Hypertext and Hypermedia, College Park, Maryland, USA May 2002, pp. 21-22. ACM Press, New York, NY, USA, 2002.

[De Bra et al., 2005] De Bra, P., Stash, N., & Smits, D.: Creating Adaptive Web-Based Applications. Tutorial at the 10th International Conference on User Modeling, Edinburgh, Scotland, July 25, 2005.

[De Bra, Smits & Stash, 2006] De Bra, P., Smits, D., & Stash, N.: The design of AHA!. In Proceedings of the 17th Conference on Hypertext and Hypermedia (Hypertext'06) , pp. 133-134. ACM Press, New York, NY, USA, 2006.

[Dieterich et al., 1993] Dieterich, H., Malinowski, U., Kühme, T., & Schneider-Hufschmidt, M.: State of the Art in Adaptive User Interfaces. In Schneider-Hufschmidt, M., Kühme, T., & Malinowski, U. (eds.): *Adaptive User Interfaces: Principles and Practice*, pp. 13-48. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1993.

[Dolog & Nejdl, 2007] Dolog, P., & Nejdl, W.: Semantic Web Technologies for the Adaptive Web. In Brusilovsky, P., Kobsa, A., & Nejdl, W. (eds.), *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 697-719. Springer Verlag, Berlin – Heidelberg, 2007.

[Eklund & Brusilovsky, 1998] Eklund, J., & Brusilovsky, P.: The value of adaptivity in hypermedia learning environments: A short review of empirical evidence. In Brusilovsky, P., & De Bra, P. (eds.), Proceedings of Second Adaptive Hypertext and Hypermedia Workshop (at the Ninth ACM International Hypertext Conference - Hypertext'98) (Computing Science Reports, Report No. 98/12), pp. 13-19. Eindhoven University of Technology, Eindhoven, 1998.

[Emiliani et al., 2003] Emiliani, P.L., Paramythis, A., Burzagli, L., & Stephanidis, C.: PALIO as an enabling platform for disabled and elderly people. In Stephanidis, C. (ed.), Universal Access in HCI: Inclusive Design in the Information Society - Volume 4 of the Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003), Crete, Greece, 22-27 June, pp. 547 - 551.Lawrence Erlbaum Associates, Mahwah - New Jersey, 2003.

[Eynard, 2008] Eynard, D.: Using semantics and user participation to customize personalization. HP Laboratories Technical Report HPL-2008-197. [On-line]. Available at: http://www.hpl.hp.com/techreports/2008/HPL-2008-197.pdf

[Fink & Kobsa, 2000] Fink, J., & Kobsa, A.: A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web. *User Modeling and User-Adapted Interaction*, 10(2-3), 209-249.

[Fink, Kobsa & Nill, 1997] Fink, J., Kobsa, A., & Nill, A.: Adaptable and Adaptive Information Access for All Users, Including the Disabled and Elderly. In Jameson, A., Paris, C., & Tasso, C. (eds.), Proceedings of the 6th International Conference on User Modelling (UM '97), Sardinia, Italy, pp. 171-173. Springer, Berlin - Haidelberg, 1997.

[Fink, Kobsa & Nill, 1999] Fink, J., Kobsa, A., & Nill, A.: Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People. *New Review of Hypermedia and Multimedia*, 4(1), 163-188.

[Finin et al., 1993] Finin, T., W., Weber, J., Widerhold, G., Genesereth, M., Fritzson, R., McKay, D., McGuire, J., Pelavin, R., Shapiro, S., & Beck, C.: Specification of the KQML Agent-Communication Language. [On-line]. Available at: http://www.cs.umbc.edu/kqml/papers/kqmlspec.ps. (1993)

[Francisco-Revilla & Shipman, 2003] Francisco-Revilla, L., & Shipman, F.: MASH: A Framework for Adaptive Spatial Hypermedia. Proceedings of the Third Workshop on Spatial Hypertext, Nottingham, UK, August 26, 2003, pp. 117-126. [On-line] Available at: http://www.csdl.tamu.edu/~shipman/SpatialHypertext/SH3/luis.pdf

[FRIEND21, 1995] Institute for Personalised Information Environment: FRIEND21: Human Interface Architecture Guidelines. Institute for Personalised Information Environment , Tokyo, 1995.

[Gershenson, 2005] Gershenson, C.: Self-organizing traffic lights. *Complex Systems*, 16(1):29–53.

[Grammenos et al., 2007] Grammenos, D., Savidis, A., Georgalis, Y., Bourdenas, T., & Stephanidis, C.: Dual Educational Electronic Textbooks: The Starlight Platform. In Proceedings of the Ninth International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'07), Tempe, Arizona, USA, October 15-17, 2007, pp. 107-114. ACM Press, New York, 2007.

[Gunderson, 1998] Gunderson, J. (ed.): WAI Accessibility User Agent Guidelines: Browser User Interface (W3C Working Draft). Web Accessibility Initiative, World Wide Web Consortium. [On-line]. Available at: http://www.w3.org/WAI/UA/WD-WAI-UA-BROWSER-0312.htm. (March, 1998)

[Grüniger & Van Treeck, 1993] Grüniger, C., & Van Treeck, W.: Contributions of a Social Science Based Evaluation for Adaptive Design Projects. In Schneider-Hufschmidt, M., Kühme, T., & Malinowski, U. (eds.): *Adaptive User Interfaces: Principles and Practice*, pp. 319-330. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1993.

[Heckmann et al., 2005] Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., & Von Wilamowitz-Moellendorff, M.: GUMO - the General User Model Ontology. In Proceedings of the 10th International Conference on User Modeling (UM'2005), Edinburgh, UK, 2005, pp. 428-432 (*Lecture Notes in Artifical Intelligence*, Volume 3538/2005). Springer-Verlag, Berlin - Heidelberg, 2005.

[Henricksen & Indulska, 2001] Henricksen, K., & Indulska, J.: Adapting the Web interface: an adaptive Web browser. In Proceedings of the Second Australasian User Interface Conference, 2001 (AUIC 2001), Gold Coast, Qld., Australia, January 29 – February 1, 2001, pp. 21-28.

[Henze, 2001] Henze, N.: Open Adaptive Hypermedia: An approach to adaptive information presentation on the Web. In Stephanidis, C. (ed.), Universal Access in HCI: Towards an Information Society for All – Volume 3 of the Proceedings of the 9th International Conference on Human-Computer Interaction (HCI International 2001), New Orleans, Louisiana, USA, 5-10 August, pp. 818-821.Lawrence Erlbaum Associates, Mahwah - New Jersey, 2001.

[Henze, Dolog & Nejdl, 2004] Henze, N., Dolog, P., & Nejdl, W.: Towards personalized e-learning in a semantic web. *Educational Technology and Society Journal*, Special Issue on Ontologies and the Semantic Web for E-learning 7(4), 82–97.

[Herder, 2003] Herder, E.: Utility-Based Evaluation of Adaptive Systems. In: Weibelzahl, S., & Paramythis, A. (eds.). Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems (pp. 25-30), held at the 9th International Conference on User Modeling UM2003, Pittsburgh, PA, USA.

[Horvitz et al., 1998] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., & Rommelse, K.: The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 256-265. Morgan Kaufmann, San Fransisco, 1998.

[Hoppe, Tauber & Ziegler, 1986] Hoppe, H., Tauber, M., & Ziegler, J.: A Survey of Models and Formal Description Methods in HCI with Example Applications. ESPRIT Project 385 HUFIT. Report B.3.2.a. 1986.

[Höök, 1997] Höök, K.: Evaluating the Usability of an Adaptive Hypermedia System. In Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI '97), Orlando, USA, pp. 179-186. ACM Press, New York, 1997.

[Höök, 2000] Höök, K.: Steps to take before intelligent user interfaces become real. *Interacting with Computers*, 12(4), 409-426.

[Höök et al., 1996] Höök, K., Karlgren, J., Waern, A., Dahlbaeck, N., Jansson, C. G., Karlgren, K., & Lemaire, B.: A Glass Box Approach to Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3), 157-184.

[Höök & Svensson, 1999] Höök, K., & Svensson, M.: Evaluating Adaptive Navigation Support. In Maybury, M. (ed.), Proceedings of the 1999 International Conference on Intelligent User Interfaces (IUI'99), Redondo Beach, Los Angeles, CA, January 5-8, 1999, pp. 187. ACM Press, Redondo Beach, CA, 1999.

[Herder, 2003] Herder, E.: Utility-Based Evaluation of Adaptive Systems. In Weibelzahl, S., & Paramythis, A. (eds.), Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems, held at the 9th International Conference on User Modeling (UM2003), Johnstown, PA, USA, 22-26 June 2003, pp. 25-30. [On-line]. Available at: http://art.ph-freiburg.de/um2003/proceedings.html

[Hermsdorf, Gappa & Pieper, 1998] Hermsdorf, D., Gappa, H., & Pieper, M.: WebAdapter: A prototype of a WWW-browser with new special needs adaptations. In Stephanidis, C., & Waern, A., (eds.) Proceedings of the 4th ERCIM Workshop on "User Interfaces for All", Stockholm, Sweden, 19-21 October 1998, 15 pages. [On-line] Available at: http://ui4all.ics.forth.gr/UI4ALL-98/hermsdorf.pdf

[Horvitz, 1999] Horvitz, E.: Principles of Mixed-Initiative User Interfaces. In Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '99), Pittsburgh, PA, USA, May 1999, pp. 159-166. ACM Press, New York, 1999.

[ISO 9241-11, 1997] International Standards Organisation: Ergonomic Requirements for office work with visual display terminals (VDTs), Part 11: Guidance on Usability (ISO 9241-11). International Standards Organisation, Geneva, Switzerland, 1997.

[ISO 13407, 1999]. International Standards Organisation: Human-Centred Design Processes for Interactive Systems (ISO 13407). International Standards Organisation, Geneva, Switzerland, 1999.

[Jacobson, Griss & Johnson, 1997] Jacobson, I., Griss, M., & Johnson, P.: Making the Reuse Business Work. *IEEE Computer*, 30(10), 36-42.

[Jameson, 2008] Jameson, A.: Adaptive Interfaces and Agents. In Sears, A., & Jacko, J. (eds.), *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (2nd ed.), pp. 433–458. Erlbaum, Mahwah, NJ, 2008.

[Jordan, 1998] Jordan, P. W.: Methods for Usability Evaluation. *An Introduction to Usability*, pp. 51-80. Taylor & Francis, London – Bristol, 1998.

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

[Karat, 1997] Karat, J.: User-Centered Software Evaluation Methodologies. In Helander, M., Landauer, T. K., & Prabhu, P. V. (eds.), *Handbook of Human-Computer Interaction* (2nd ed.), pp. 689-704.Elsevier Science Publishers B.V (North-Holland), Amsterdam, 1997.

[Kay, 1995] Kay, J.: The um toolkti for cooperative user modelling. *User Modelling and User Adapted Interaction*, 4(3), 149-196.

[Kieras & Bovair, 1984] Kieras, D.E., & Bovair, S.: The role of a mental model in learning to operate a device. *Cognitive science*, 8(3), 255-273.

[King, Evans & Blenkhorn, 2004] King, A., Evans, G., & Blenkhorn, P.: WebbIE: a web browser for visually impaired people. In Proceedings of the The 2nd Cambridge Workshop on Universal Access And Assistive Technology (CWUAAT), 2004.

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

[Kobsa, 1993]. Kobsa, A.: User Modelling: Recent Work, Prospects and Hazards. In Schneider-Hufschmidt, M., Kühme, T., & Malinowski, U. (eds.): *Adaptive User Interfaces: Principles and Practice*, pp. 111-128. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1993.

[Kobsa, 2001] Kobsa, A.: Generic user modeling systems. *User Modeling and User Adapted Interaction*, 11(1-2), 49-63.

[Kobsa, 2007] Kobsa, A.: Privacy-Enhanced Web Personalization. In Brusilovsky, P., Kobsa, A., & Nejdl, W. (eds.), *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 628-670. Springer Verlag, Berlin – Heidelberg - New York, 2007.

[Kobsa & Pohl, 1995] Kobsa, A., & Pohl, W.: The user modelling shell system BGP-MS. *User Modelling and User-adapted Interaction*, 4(2), 59-106.

[Krogsæter & Thomas, 1994]. Krogsæter, M., & Thomas, C. G.: Adaptivity: System-Initiated Individualisation. In Oppermann, R.(ed): *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*, pp. 67-96. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1994.

[Krogsæter, Oppermann & Thomas, 1994] Krogsæter, M., Oppermann, R., & Thomas, C. G.: A User Interface Integrating Adaptability and Adaptivity. In Oppermann, R. (ed.), *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*, pp. 97-125. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1994.

[Magoulas, Chen & Papanikolaou, 2003] Magoulas, G. D., Chen, S. Y., & Papanikolaou, K. A. (2003). Integrating Layered and Heuristic Evaluation for Adaptive Learning Environments. In Weibelzahl, S., & Paramythis, A. (eds.), Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems (pp. 5-14), held at the 9th International Conference on User Modeling UM2003, Pittsburgh.

[McGrath, 1995] McGrath, J. E.: Methodology Matters: Doing Research in the Behavioral and Social Sciences. In Baecker, R.M., Grudin, J., Buxton, W. A. S., & Greenberg, S. (eds.), *Readings in Human Computer Interaction Toward the Year 2000* (2nd ed.), pp. 152-169. Morgan Kaufmann Publishers, San Fransisco, CA, USA, 1995.

[McLoughlin, 1999] McLoughlin, C.: Implications of the research literature on learning styles for the design of instructional material. *Australian journal of educational technology*, 15(3), 222-241. [On-line]. Available at: http://www.ascilite.org.au/ajet/ajet15/mcloughlin.html

[Meyer, 1994] Meyer, B.: Adaptive Performance Support: User Acceptance of a Self-Adapting System. In Goodman, B., Kobsa, A., & Litman, D. (eds.), Fourth International Conference on User Modeling (UM94), Hyanis, Cape Cod, MA, USA, August 15-19, 1994, pp. 65-70. User Modeling Inc., Boston, 1994.

[Mobasher et al., 2007] Mobasher, B., Burke, R., Bhaumik, R., & Williams, C.: Towards trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Transactions on Internet Technology, 7(4), 23:1-38.

[Mühlbacher & Putzinger, 2006] Mühlbacher, J.R., & Putzinger, A. (eds.): Die WeLearn Plattform. Trauner Verlag, Linz, Austria, 2006.

[Narayanan & Shmatikov, 2008] Narayanan, A., & Shmatikov, V.: Robust De-anonymization of Large Sparse Datasets. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP 2008), May 18-21, 2008, Oakland, California, USA, pp. 111-125. IEEE Computer Society, Los Alamitos, CA, USA, 2008.

[Narayanan & Shmatikov, 2009] Narayanan, A., & Shmatikov, V.: De-anonymizing Social Networks. In Proceedings of the 2009 IEEE

Symposium on Security and Privacy (SP 2009), May 17-20, 2008, Oakland, California, USA, pp. 173-187. IEEE Computer Society, Los Alamitos, CA, USA, 2009.

[Nielsen, 1994] Nielsen, J.: Usability Engineering. (2nd ed.). Morgan Kaufmann Publishers, San Fransisco, CA, 1994..

[Nielsen, 1995] Nielsen, J.: *Multimedia and Hypertext: The Internet and Beyond.* AP Professional, Boston, MA, 1995.

[Oppermann, 1994] Oppermann, R.: Adaptively supported adaptability. *International Journal of Human Computer Studies*, 40(3), 455-472.

[Oppermann & Simm, 1994]. Oppermann, R., & Simm, H.: Adaptability: User-Initiated Individualisation. In Oppermann, R.(ed): *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*, pp. 14-66. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1994.

[Oppermann & Specht, 1998] Oppermann, R., & Specht, M.: Adaptive Support for a Mobile Museum Guide. In Proceedings of the Interactive Applications on Mobile Computing Conference (IMC '98), Rostock, Germany, 24-25 November 1998. [On-line]. Available at: http://fit.fraunhofer.de/~oppi/publications/NoWoRost.NomadicComp.pdf

[Ounaies, Jamoussi & Ghezala, 2008] Ounaies, H.Z., Jamoussi, Y., & Ghezala, H.B.: Measurement framework for aligning adaptation methods with business and usage factors in e-learning. In Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 (IMECS 2008), 19-21 March, 2008, Hong Kong, pp. 35-39. [On-line] Available at: http://www.iaeng.org/publication/IMECS2008/IMECS2008_pp29-34.pdf

[PALIO Consortium, 2003] PALIO Consortium: PALIO field-trials - Results of the final evaluation. IST-1999-20656 PALIO Project Deliverable D24.

[Paramythis, 2004] Paramythis, A.: Towards Self-Regulating Adaptive Systems. In Weibelzahl, S., & Henze, N. (eds.), Proceedings of the Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems of the German Informatics Society (ABIS04), Berlin, October 4-5, pp. 57-63.

[Paramythis, 2006] Paramythis, A.: Self-regulated adaptivity as a design and authoring support tool. In Proceedings of the joint 1st International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H) and the 4th International Workshop on Authoring of Adaptive & Adaptable Educational Hypermedia (A3EH), held in conjunction with the Fourth Adaptive Hypermedia and Adaptive Web-Based Systems 2006 Conference (AH 2006), June 21-23, 2006, Dublin, Ireland, pp. 355-366.

[Paramythis & Stephanidis, 2005] Paramythis, A., & Stephanidis, C.: A Generic Adaptation Framework for Hypermedia Systems. In Chen, S. Y., & Magoulas, G. D. (eds.), *Adaptable and Adaptive Hypermedia Systems*, pp. 80-103. Idea Group, Inc. (IRM Press), Hershey, PA, USA, 2005.

[Paramythis & Van Velsen, 2009] Paramythis, A., & Van Velsen, L.: Lessons learned from the evaluations of an adaptive meta-search engine. Paper presented in the Workshop "Users' Preferences Regarding Intelligent User Interfaces: Differences Among Users and Changes Over Time", held in conjunction with the Intelligent User Interfaces 2009 Conference, February 8th, 2009, Sanibel Island, Florida, USA.

[Paramythis & Weibelzahl, 2005] Paramythis, A., & Weibelzahl, S.: A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In Ardissono, L., Brna, P., & Mitrovic, A. (eds.), Proceedings of the 10th International Conference on User Modeling (UM2005), Edinburgh, Scotland, UK, July 24-29, pp. 438-442 (*Lecture Notes in Computer Science* Volume 3538 / 2005). Springer, Berlin - Heidelberg, 2005.

[Paramythis, Totter & Stephanidis, 2001] Paramythis, A., Totter, A. & Stephanidis, C.: A Modular Approach to the Evaluation of Adaptive User Interfaces. In Weibelzahl, S., Chin, D., & Weber, G. (eds.), Proceedings of the Empirical Evaluation of Adaptive Systems workshop held at the Eighth International Conference on User Modeling, Sonthofen, Germany, July 2001, pp. 9-24,. Pedagogical University of Freiburg, Freiburg, 2001.

[Paramythis et al., 2003a] Paramythis, A., Alexandraki, C., Segkos, I., Maou, N., & Stephanidis, C.: Personalisable, context-aware services: the PALIO approach. In Stephanidis, C., (ed.), Universal Access in HCI: Inclusive Design in the Information Society - Volume 4 of the Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003), Crete, Greece, 22-27 June 2003, pp. 582 - 586. Lawrence Erlbaum Associates, Mahwah - New Jersey, 2003.

[Paramythis et al., 2003b] Paramythis, A., Alexandraki, C., Segkos, I., Maou, N., & Stephanidis, C.: The PALIO framework for hypermedia adaptations. In C. Stephanidis, (Ed.), Universal Access in HCI: Inclusive Design in the Information Society - Volume 4 of the Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003), Crete, Greece, 22-27 June 2003, pp. 587 - 591. Lawrence Erlbaum Associates, Mahwah - New Jersey, 2003.

[Paetau, 1994] Paetau, M.: Configurative Technology: Adaptation to Social Systems Dynamism. In Oppermann, R.(ed): *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software*, pp. 194-234. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1994.

[Papanikolaou & Grigoriadou, 2004] Papanikolaou, K.A., & Grigoriadou, M.: Accommodating learning style characteristics in adaptive educational hypermedia systems. In Proceedings of the Workshop "Individual Differences in Adaptive Hypermedia", held in conjunction with the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004), Eindhoven University of Technology, The Netherlands, August 23 – 26 2004. [On-line] Available at: http://hermes.di.uoa.gr/lab/cvs/papers/papanikolaou/AH2004_WS_Papanikolaou.pdf

[Petrie et al., 1997] Petrie, H., Morley, S., McNally, P., O'Neill, A.M., & Majoe, D.: Initial design and evaluation of an interface to hypermedia systems for blind users. In Proceedings of Hypertext '97, Southampton, UK, April 6-11, 1997, pp. 48-56. ACM Press, New York, 1997.

Popescu, 2009] Popescu, E.: Addressing Learning Style Criticism: The Unified Learning Style Model Revisited. In Advances of Web Based Learning – ICWL 2009, pp. 332-342. (*Lecture Notes in Computer Science*, Volume 5686 / 2009). Springer, Berlin - Heidelberg, 2009

[Prokopenko, 2008] Prokopenko, M. (ed.): *Advances in Applied Self-organizing Systems*. Springer, London, UK, 2008.

[Savidis et al., 1997] Savidis, A., Paramythis, A., Akoumianakis, D., & Stephanidis, C.: Designing user-adapted interfaces: the unified design method for transformable interactions. In Proceedings of the ACM Conference on Designing Interactive Systems: Processes, Methods and Techniques (DIS '97), Amsterdam, The Netherlands, August 1997, pp. 323-334. ACM Press, New York, 1997.

[Schmidt-Belz, 2005] Schmidt-Belz, B.: User trust in adaptive systems. Paper presented at the Workshop on Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen (ABIS), Saarbrücken, Germany (2005).

[Seefelder de Assis & Schwabe, 2004] Seefelder de Assis, P., & Schwabe, D.: A General Meta-model for Adaptive Hypermedia Systems. *Lecture Notes in Computer Science*, Volume 3137 / 2004, pp. 433-436. Springer, Berlin, 2004.

[Seefelder de Assis, Schwabe & Arraes Nunes, 2006] Seefelder de Assis, P., Schwabe, D., & Arraes Nunes, D.: ASHDM – Model-Driven Adaptation and Meta-adaptation. In Wade, V., Ashman, H., & Smyth, B. (eds.), Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2006), Dublin, Ireland, June 21-23, 2006 (*Lecture Notes in Computer Science*, Volume 4018 / 2006), pp. 213-222. Springer, Berlin – Heidelberg, 2006.

[Shearin & Lieberman, 2001] Shearin, S., & Lieberman, H.: Intelligent profiling by example. In Lester, J. (ed.), Proceedings of the 2001 International Conference on Intelligent User Interfaces (IUI 2001), pp. 145-151. ACM Press, New York, 2001.

[Shneiderman & Maes, 1997] Shneiderman, B., & Maes, P.: Direct manipulation vs. interface agents. *interactions*, 4(6), 42-61.

[Spalteholz, Li & Livingston, 2007] Spalteholz, L., Li, K. F., & Livingston, N.: Efficient navigation on the world wide web for the physically disabled. In Fillipe, J., & Cordeiro, J. (eds.), Proceedings of the 3rd International Conference on Web Information Systems and Technologies, Barcelona, Spain, March 3-6, 2007, pp. 321-326. Springer, Heidelberg – Berlin, 2008.

[Stary & Totter, 1997] Stary, C., & Totter, A.: How to Integrate Concepts for the Design and the Evaluation of Adaptable and Adaptive User Interfaces. In Stephanidis, C., & Carbonell, N. (eds.), Proceedings of the 3rd ERCIM Workshop on "User Interfaces for All", Obernai, France, 3-4 November 1997, pp. 7.1-7.15. [On-line]. Available at: http://ui4all.ics.forth.gr/UI4ALL-97/stephanidis.pdf

[Stephanidis, 1995] Stephanidis, C.: Towards User Interfaces for All: Some Critical Issues. In: Panel Session on "User Interfaces for All - Everybody, Everywhere, and Anytime", Proceedings of HCI International '95, Tokyo, Japan, 9-14 July 1995, pp. 137-142 Elsevier Science & Technology, Burlington, MA, 1995.

[Stephanidis et al., 1997b]. Stephanidis, C., Savidis, A., & Paramythis, A.: Addressing Cultural Diversity Through Unified Interface Development. In Proceedings of the HCI International '97 Conference, San Francisco, USA, 24-29 August 1997, pp. 165-168. Elsevier Science & Technology, Burlington, MA, 1997.

[Stephanidis, 2001a] Stephanidis, C.: User Interfaces for All: New perspectives into Human-Computer Interaction. In Stephanidis, C. (ed.), *User Interfaces for All - Concepts, Methods, and Tools*, pp. 3-17. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2001.

[Stephanidis, 2001b] Stephanidis, C.: Adaptive techniques for Universal Access. *User Modelling and User Adapted Interaction International Journal*, 10th Anniversary Issue, 11 (1/2), 159-179.

[Stephanidis, Paramythis & Sfyrakis, 1999] Stephanidis, C., Paramythis, A., & Sfyrakis, M.: Evaluating Adaptable and Adaptive User Interfaces: Lessons Learned from the Development of the AVANTI Web Browser. In Kobsa, A., & Stephanidis, C. (eds.), Proceedings of the 5th ERCIM Workshop on "User Interfaces for All", Dagstuhl, Germany, 28th November - 1st December 1999, pp. 22.1-22.6. [On-line]. Available at: http://ui4all.ics.forth.gr/UI4ALL-99/Stephanidis_s.pdf

[Stephanidis & Savidis, 1995] Stephanidis, C., Savidis, A.: Towards Multimedia Interfaces for all: a New Generation of Tools Supporting Integration of Design-time and Run-time Adaptivity Methods. In Proceedings of the workshop on "Adaptive Multimedia Technologies for People with Disabilities", held in conjunction with the Third ACM International Conference on Multimedia '95, November 5-9, 1995, San Francisco, CA, USA.

[Stephanidis, Savidis & Akoumianakis, 1997] Stephanidis, C., Savidis, A., & Akoumianiakis, D.: Unified Interface Development: Tools for Constructing Accessible and Usable User Interfaces. Tutorial 13, HCI International '97 Conference, San Fransisco, USA, August 1997. Available on-line at: http://www.ics.forth.gr/hci/publications/tutorials.html.

[Stephanidis et al., 1997] Stephanidis, C., Karagiannidis, C., Paramythis, A., Sfyrakis, M., Stergiou, A., Savidis, A., Schreck, J., & Fink, J.: Report on Rules. ACTS AC042 AVANTI Project, Deliverable DE012, 82 pages, February 1997.

[Stephanidis, Paramythis et al., 1997] Stephanidis, C., Paramythis, A., Savidis, A., Stergiou, A., Sfyrakis, M., Maou, N., Leventis, A., Paparoulis, G., Vernardos, G., Karagiannidis, C.: Enhancing Web Browsers: Accessibility, Usability, Navigation and Orientation (Internal Technical Report). Assistive Technology and Human-Computer Interaction Laboratory, Institute of Computer Science, Foundation for Research and Technology - Hellas (1997) (Available from the authors)

[Stephanidis, Paramythis et al., 1998a] Stephanidis, C., Paramythis, A., Sfyrakis, M., Stergiou, A., Maou, N., Leventis, A., Paparoulis, G., & Karagiannidis, C.: Adaptable and Adaptive User Interfaces for Disabled Users in AVANTI Project. In Proceedings of the 5th International Conference on Intelligence in Services and Networks (IS&N '98), "Technology for Ubiquitous Telecommunications Services", Antwerp, Belgium, 25-28 May 1998, pp. 153-166 (*Lecture Notes in Computer Science*, Volume 1430 / 1998). Springer, Berlin - Heidelberg, 1998.

[Stephanidis, Paramythis et al., 1998b] Stephanidis, C., Paramythis, A., Akoumianakis, D., & Sfyrakis, M.: Self-Adapting Web-based Systems: Towards Universal Accessibility. In Stephanidis, C., & Waern, A. (eds.), Proceedings of the 4th ERCIM Workshop "User Interfaces for All", Stockholm, Sweden, 19-21 October 1998. [On-line]. Available at: http://ui4all.ics.forth.gr/UI4ALL-98/stephanidis2.pdf

[Stephanidis et al., 1998a]. Stephanidis, C., Akoumianakis, D., Sfyrakis, M., & Paramythis, A.: Universal accessibility in HCI: Process-oriented design guidelines and tool requirements. 4th ERCIM Workshop on "User Interfaces for All", Stockholm, Sweden, 19-21 October 1998. [On-line]. Available at: http://ui4all.ics.forth.gr/UI4ALL-98/stephanidis1.pdf

[Stephanidis et al., 1998b] Stephanidis C. (Ed.), Salvendy, G., Akoumianakis, D., Bevan, N., Brewer, J., Emiliani, P.L., Galetsas, A., Haataja, S., Iakovidis, I., Jacko, J., Jenkins, P., Karshmer, A., Korn, P., Marcus, A., Murphy, H., Stary, C., Vanderheiden, G., Weber, G., & Ziegler, J. (1998). Toward an Information Society for All: An International R&D Agenda. *International Journal of Human-Computer Interaction*, 10(2), 107-134.

[Stephanidis, Paramythis et al., 2001] Stephanidis, C., Paramythis, A., Sfyrakis, M., & Savidis, A.: A Case Study in Unified User Interface Development: The AVANTI Web Browser. In Stephanidis, C. (ed.), *User Interfaces for All - Concepts, Methods, and Tools*, pp. 525-568.Lawrence Erlbaum Associates, Mahwah, NJ, 2001.

[Stephanidis, Paramythis et al., 2004] Stephanidis, C., Paramythis, A., Zarikas, V., & Savidis, A.: The PALIO Framework for Adaptive Information Services. In Seffah, A., & Javahery, H. (eds.), *Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces*, pp. 69-92.John Wiley & Sons, Ltd., Chichester, UK, 2004.

[Sternberg, 1985]. Sternberg, R. J. (ed.): *Human Abilities: An Information-Processing Approach*. W. H. Freeman and Company, 1985.

[Story, 1998] Story, M. F.: Maximising Usability: The Principles of Universal Design. *Assistive Technology Journal*, 10(1), 4-12.

[Sukaviriya & Foley, 1993] Sukaviriya, P., & Foley, J.: Supporting Adaptive Interfaces in a Knowledge-Based User Interface Environment. In: Gray, W. D., Hefley, W. E., Murray, D. (eds.), Proceedings of the 1st International Conference on Intelligent User Interfaces, Orlando, FL, pp. 107-114. ACM Press, New York, NY, USA, 1993.

[Szekely, 1991]. Szekely, P.: Structuring Programs to Support Intelligent Interfaces. In Sullivan, J. W., & Tyler, S. W. (eds.): *Intelligent User Interfaces*, pp. 445-464. ACM Press, New York, NY, USA, 1991.

[Tan, Yu & McAllister, 2006] Tan, C.C., Yu, W., & McAllister, G.: Developing an ENABLED adaptive architecture to enhance internet accessibility for visually impaired people. *International Journal on Disability and Human Development*, 5 (2), 97-104.

[Tarpin-Bernard, Marfisi-Schottman & Habieb-Mammar, 2009] Tarpin-Bernard, F., Marfisi-Schottman, I., & Habieb-Mammar, H.: AnAmeter: The First Steps to Evaluating Adaptation. In Weibelzahl, S., Masthoff, J., Paramythis, A., & Van Velsen, L. (eds.), Proceedings of the Sixth Workshop on User-Centred Design and Evaluation of Adaptive Systems, held in conjunction with the International Conference on User Modeling, Adaptation, and Personalization (UMAP2009), Trento, Italy, June 26th, 2009 (pp. 11-20). CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/Vol-462/paper2.pdf

[Tauscher & Greenberg, 1997] Tauscher, L., & Greenberg, S.: How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems. *International Journal of Human Computer Interaction*, Special issue on World Wide Web Usability, 47(1), 97-138.

[Thevenin, 2002] Thevenin D.: ARTStudio; Tool for Multi-target UI Design. Poster presented at the 15th Annual Symposium on User Interface Software &Technology (UIST 02), Paris, France 27-30 October 2002.

[Thevenin & Coutaz, 1999] Thevenin, D., & Coutaz, J.: Plasticity of User Interfaces: Framework and Research Agenda. In Sasse, A., & Johnson, C. (eds.), Proceeding of the 1999 IFIP Conference on Human Computer Interaction (INTERACT'99), Edinburgh, Scotland, August 30 - September 3, 1999, pp. 110-117. IOS Press, 1999.

[Thevenin, Coutaz & Calvary, 2004] Thevenin, D., Coutaz, J., & Calvary, G.: A Reference Framework for the Development of Plastic User Interfaces. In Seffah, A., & Javahery, H. (eds.), *Multiple User Interfaces*, pp. 27-51. John Wiley & Sons, Ltd., Chichester, UK, 2004.

[Tobar, 2003] Tobar, C. M.: Yet Another Evaluation Framework. In: Weibelzahl, S., & Paramythis, A. (eds.), Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems (pp. 15-24), held at the 9th International Conference on User Modeling UM2003, Pittsburgh.

[Totterdell, 1990] Totterdell, P.: Introduction. In Browne, D., Totterdell, P., & Norman, M. (eds.), *Adaptive User Interfaces*, pp. 1-14. Academic Press, London, 1990.

[Totterdell & Boyle, 1990] Totterdell, P., & Boyle, E.: The Evaluation of Adaptive Systems. In Browne, D., Totterdell, P., & Norman, M. (eds.), *Adaptive User Interfaces*, pp. 161-194. Academic Press, London, 1990.

[Totterdell & Rautenbach, 1990] Totterdell, P., & Rautenbach, P.: Adaptation as a Problem of Design. In Browne, D., Totterdell, P., & Norman, M. (eds.), *Adaptive User Interfaces*, pp. 61-84. Academic Press, London, 1990.

[Totterdell et al., 1990]. Totterdell, P., Rautenbach, P., Wilkinson, A., & Anderson, S. O.: Adaptive Interface Techniques. In Browne, D., Totterdell, P., & Norman, M. (eds.), *Adaptive User Interfaces*, pp. 131-160. Academic Press, London, 1990.

[Trevellyan & Browne, 1986] Trevellyan, R., & Browne, D. P.: A self-regulating adaptive system. In Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface, Toronto, Ontario, Canada, 1986, pp. 103-107.

[Tyler et al., 1991]. Tyler, S. W., Schlosseberg, J. L., Gargan, R. A., Cook, L. K., & Sullivan, J. W.: An Intelligent Interface Architecture for Adaptive Interaction. In Sullivan, J. W., & Tyler, S. W. (eds.): *Intelligent User Interfaces*, pp. 85-109. ACM Press, New York, NY, USA, 1991.

[Tzafestas, 2005] Tzafestas, E.S.: Regulation problems in explorer agents. *Cybernetics and Systems*, 36, 339-377.

[Vanderheiden & Kaine-Krolak, 1995] Vanderheiden, G.C., & Kaine-Krolak, M. Access to Current and Next-Generation Information Systems by People with Disabilities [On-line]. Available at: http://www.tracecenter.org/docs/access_info_sys/full_doc.htm

[Vanderheiden & Chisholm, 1998] Vanderheiden, G., Chisholm, W. (eds.): Page Author, User Agent and Assistive Technology Guidelines and Issues - Version 8, Unified Web Site Accessibility Guidelines (Draft). Trace R and D Center, University of Wisconsin - Madison, Web Accessibility Initiative - Guidelines Working Group. Available on-line at: http://trace.wisc.edu/archive/html_guidelines/author.htm.

[Vassileva & Bontchev, 2006] Vassileva, D., & Bontchev, B.: Self adaptive hypermedia navigation based on learner model characters. In Proceedings of the 3rd International Conference on Education (IADAT-e2006), Barcelona, Spain, July 12-14, 2006. [On-line] Available at: http://www.iadat.org/iadat-e2006/abstracts_web/IADAT-e2006_13.pdf

[Vasilyeva et al., 2007] Vasilyeva, E., Pechenizkiy, M., Gavrilova, T., & Puuronen, S.: Personalization of Immediate Feedback to Learning Styles. In Proceedings of the.Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007), pp. 622-624.

[Weber et al., 2001] Weber, G., Kuhl, H.-C., & Weibelzahl, S.: Developing adaptive internet based courses with the authoring system NetCoach. In Reich, S., Tzagarakis, M., & De Bra, P. (eds.), "Hypermedia: Openness, Structural Awareness, and Adaptivity", Proceedings of the Third Workshop on Adaptive Hypermedia (AH2001) Berlin, 2001, pp.226-238. Springer, Berlin - Heidelberg, 2001.

[Weber & Specht, 1997] Weber, G., & Specht, M.: User modeling and adaptive navigation support in WWW-based tutoring systems. In Jameson, A., Paris, C., & Tasso, C. (eds.), Proceedings of the 6th International Conference on User Modelling (UM '97), Sardinia, Italy, pp. 289-300. Springer, Berlin - Heidelberg, 1997.

[Weibelzahl, 2001] Weibelzahl, S.: Evaluation of adaptive systems. In Bauer, M., Gmytrasiewicz, P., & Vassileva, J. (eds.), User Modeling 2001: Proceedings of the Eighth International Conference, UM2001, pp. 292-294 (*Lecture Notes in Computer Science*, Volume 2109/2009). Springer, Berlin - Heidelberg, 2001.

[Weibelzahl & Lauer, 2001] Weibelzahl, S., & Lauer, C. U.: Framework for the Evaluation of Adaptive CBR-Systems. In Reimer, U., Schmitt, S., & Vollrath, I. (eds.), Proceedings of the 9th German Workshop on Case-Based Reasoning (GWCBR01), pp. 254-263.Shaker, Aachen, 2001.

[Weibelzahl, Paramythis & Masthoff, 2007] Weibelzahl, S., Paramythis, A., & Masthoff, J.: Formative Evaluation Methods for Adaptive Systems. Tutorial no 4 in the 11th International Conference on User Modeling (UM 2007), 25 - 29 June 2007, Corfu, Greece.

[Williams, Mobasher & Burke, 2007] Williams, C.A., Mobasher, B., & Burke, R: Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*, 1(3), 157-170.

[De Witt & Hakkinen, 1998] De Witt, J.C., & Hakkinen, M.T.: Surfing the Web with pwWebSpeak. In Proceedings of the Technology and Persons with Disabilities Conference 1998.

[Wixon & Wilson, 1997] Wixon, D., & Wilson, C.: The Usability Engineering Framework for Product Design and Evaluation. In Helander, M., Landauer, T. K., & Prabhu, P. V. (eds.), *Handbook of Human-Computer Interaction* (2nd ed.), pp. 653-688. Elsevier Science Publishers (North-Holland), Amsterdam, 1997.

APPENDIX A: CURRICULUM VITAE

Personal information

Name	Paramythis Alexandros
Date / place of birth	June 8th, 1972, Orestiada, Greece
Address	Haushoferstr. 25/4, A-4020, Linz, Austria
Telephone	+43 664 5219904
Email	paramythis@gmail.com

Education

Present	**PhD Candidate** *Johannes Kepler University (Linz, Austria)*
September 1998	**M.Sc. in Computer Science** Thesis title: "Software Architecture for Adaptable and Adaptive User Interface Development" Specialization area: Human-Computer Interaction and Information Systems *University of Crete (Heraklion, Greece)*
September 1996	**B.Sc. in Computer Science** Thesis title: "Navigation and Interaction Techniques for 3D Virtual Environments" *University of Crete (Heraklion, Greece)*
February 1988	**Certificate of Proficiency in English** *University of Michigan (Thessaloniki, Greece)*
December 1987	**Certificate of Proficiency in English** *University of Cambridge (Thessaloniki, Greece)*

Affiliations

Apr. 2002 – present	**Project Member / Researcher** Institute for Information Processing and Microprocessor Technology (FIM) *Johannes Kepler University (Linz, Austria)*
May 2002 – Apr. 2003	**External Consultant and Project Coordinator** Human-Computer Interaction and Assistive Technology Laboratory, Institute of Computer Science, Foundation for Research and Technology – Hellas *ICS-FORTH (Heraklion, Greece)*
Apr. 2001 – Mar. 2002	**Military Service** Twelve-month service in the Greek army, under compulsory conscription.

June 1994 – Mar. 2001	**Research Assistant** (Jun. 1994 – Dec. 1998), **Research Staff** (Jan. 1999 – Mar. 2001) Human-Computer Interaction and Assistive Technology Laboratory, Institute of Computer Science, Foundation for Research and Technology – Hellas *ICS-FORTH (Heraklion, Greece)*
Oct. 1992 – June 1994	**English Language Tutor** *Lambraki Language Inst. (Heraklion, Greece)*
Oct. 1991 – July 1992	**English Language Tutor** Saridakis Language Inst. (Heraklion, Greece)
Jan. 1991 – Sep. 1991	**Computer Club Organiser and Administrator** *'INFO' Computer Club (Komotini, Greece)*
Sep. 1990 – Sep. 1991	**Co-editor of the 'Computer News' Bimonthly Newsletter** *'Computer News' Newsletter (Komotini, Greece)*
Sep. 1989 – Sep. 1991	**Independent Programming Contractor** *Independent Contractor (Komotini, Greece)*

Teaching

University courses

- Systemnahe Programmierung
  353.007 / 353.007, Johannes Kepler University, Linz, Austria
  2009 W, 2008 W, 2007 W
- Spezielle Kapitel aus Informatik: Privacy, Security and Trust in Personalized Systems
  353.044, Johannes Kepler University, Linz, Austria
  2009 S, 2008 S
- Spezielle Kapitel aus Informatik: Adaptive Hypermedia Systems
  353.046, Johannes Kepler University, Linz, Austria
  2008 W
- Spezielle Kapitel aus Informatik: Adaptive Web-Based Systems
  (Spezielle Kapitel aus Teleteaching/Telelearning: Adaptive Web-Based Systems)
  353.023 (353.008), Johannes Kepler University, Linz, Austria
  2007 W, 2006 W, 2005 W, 2004 W, 2003 W, 2002 W
- Praktikum: Programmiersprache C
  353.027, Johannes Kepler University, Linz, Austria
  2007 S, 2006 S, 2005 S, 2004 S, 2003 S
- Praktikum: Programmiersprache C++
  353.021, Johannes Kepler University, Linz, Austria
  2007 S, 2006 W, 2005 W
- Informatik 3
  Engineering für Computer-basiertes Lernen (currently: Kommunikation, Wissen, Medien), Upper Austria University of Applied Sciences, Hagenberg, Austria
  2002 W, 2003 W, 2004 W

Other teaching experience

Five years of experience in teaching English as a foreign language, in both classroom and private settings. Levels taught range from intermediate to advanced, including students preparing for the Certificate for Proficiency in English exam.

Tutorials presented at major international conferences:

- Weibelzahl, S., Paramythis, A., & Masthoff, J. (2007). Tutorial on "Formative Evaluation Methods for Adaptive Systems". *Tutorial no 4 in the 11th International Conference on User Modeling (UM 2007)*, Corfu, Greece, 25–29 June, 2007.
- Stephanidis, C., Akoumianakis, D., & Paramythis, A. (1999). Tutorial on "Coping with Diversity in HCI: Techniques for Adaptable and Adaptive Interaction". *Tutorial no 11 in the 8th International Conference on Human-Computer Interaction (HCI International '99)*, Munich, Germany, 22–26 August. [On-line]. Available: http://www.ics.forth.gr/proj/at-hci/html/tutorials.html

Participation in Research & Development Projects

International Projects

- ALS, "Adaptive Learning Spaces" (229714-CP-1-2006-1-MINERVA-M), 2006–2009
- PALIO, "Personalised Access to Local Information and services for tOurists" (IST-1999-20656), 2000–2002
- JUST, "JUST-in-time health emergency interventions – Training of non-professionals by Virtual Reality and advanced IT tools" (IST-1999-12581), 2000–2002
- TACIT, "Theory and Applications of Continuous Interaction Techniques" (Training and Mobility of Researchers – Research Networks, DG XII), 1998–2002
- Accessible version of the "Consolidated Treaties of the European Union" Web site (subcontract in the context of a framework project assigned by OPOCE to European Dynamics), 2000.
- ATTRACT, "Applications in Telemedicine Taking Rapid Advantage of Cable Television Network Evolution" (TELEMATICS, DGXIII), 1998–2000
- TIDE-WAI "TIDE – Web Accessibility Initiative" (DE 4105), 1998–2000
- TEMeTeN, "Towards a European Medical and Teleworking Network (RISI 2, DGXVI), 1997–1999
- AVANTI , "Adaptive and Adaptable Interactions for Multimedia Telecommunications Applications" (ACTS-AC042), 1995–1998
- ACCESS "Development platform for unified Access to enabling environments" (TIDE-TP1001), 1994–1996
- HANDYNET "Development of a European Information System for people with special needs" (HELIOS Programme), 1989–1996

National Projects

- ASCOLLA, "Adaptive Support for Collaborative E-Learning" (FWF P20260-N15), 2008–present
- "Integrating Agents into Teleteaching Webportals" (FWF P15947-N04), 2002–2005
- SKEPSIS, "System for the Distributed Indexing of Lecture Notes and Web Pages" (University of Crete, Greek Ministry for Education and Religious Affairs), 1999–2000

- EC-Tourism, “ELECTRONIC COMMERCE – Provision of Tourism Services over the Internet” (EPET II, General Secretariat for Research & Technology – Greece), 1999–2001
- NAUTILUS, “Universally Accessible Information Kiosks” (EPET II, General Secretariat for Research & Technology – Greece), 1999–2001
- PERIGRAMMA, “Office-support Environment for People with Special Needs” (EPET II, General Secretariat for Research & Technology – Greece), 1999–2001
- HESTIA “Vocational Training of People with Special Needs in Computer-Based Working Environments” (HORIZON Programme), 1996–1997

Publications

Editorships

- Weibelzahl, S., Paramythis, A., & Masthoff, J. (Eds.) (2010) Principled Evaluation of Adaptive Systems. Special Issue of the User Modeling and User Adapted Interaction journal, scheduled for publication in 2010.
- Weibelzahl, S., Masthoff, J., Paramythis, A., & Van Velsen, L. (Eds.) (2009). Proceedings of the Sixth Workshop on User-Centred Design and Evaluation of Adaptive Systems, held in conjunction with the International Conference on User Modeling, Adaptation, and Personalization (UMAP2009), June 26th, 2009, Trento, Italy.
- Paramythis, A., and Weibelzahl, S. (Eds.) (2008). Proceedings of the Workshop on Adaptive Collaboration Support, held in conjunction with the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'08), 29 July - 1 August 2008, Hannover, Germany.
- Weibelzahl, S., Paramythis, A., and Masthoff, J. (Eds.) (2006). Proceedings of the Fifth Workshop on User-Centred Design and Evaluation of Adaptive Systems, held in conjunction with the 4th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH'06), June 20th 2006, Dublin, Ireland.
- Weibelzahl, S., Paramythis, A., & Masthoff, J. (Eds.) (2005). Proceedings of the Fourth Workshop on the Evaluation of Adaptive Systems. Held in conjunction with the 10th International Conference on User Modeling (UM'05), Edinburgh, UK , July 24th to 30th, 2005.
- Weibelzahl, S. & Paramythis, A. (Eds.) (2004). Proceedings of the Third Workshop on Empirical Evaluation of Adaptive Systems. Held in conjunction with the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004), August 23 - 26, Eindhoven, The Netherlands.
- Weibelzahl, S. & Paramythis, A. (Eds.) (2003). Proceedings of the Second Workshop on Empirical Evaluation of Adaptive Systems. Held at the Ninth International Conference on User Modeling (UM2003), June 2003, Johnstown, PA.

Journal articles and Book chapters

- Stephanidis, C., Paramythis, A., & Savidis, A. (2005). Developing Adaptive Interfaces for the Web. In R. Proctor & K. Vu (Eds.), Handbook of Human Factors in Web Design (pp. 251-266). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Paramythis, A., & Stephanidis, C. (2005). A Generic Adaptation Framework for Hypermedia Systems. In Chen, S. Y., & Magoulas, G. D. (Eds.) Adaptable and Adaptive Hypermedia Systems (pp. 80-103). Idea Group, Inc.

- Paramythis, A., & Loidl-Reisinger, S. (2004). Adaptive Learning Environments and e-Learinng Standards. Electronic Journal of e-Learning, 2 (1), 181-194.
- Stephanidis, C., Paramythis, A., Zarikas, V., & Savidis, A. (2004). The PALIO Framework for Adaptive Information Services. In A. Seffah & H. Javahery (Eds.), Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces (pp. 69-92). Chichester, UK: John Wiley & Sons, Ltd.
- Guillen, S., Arredondo, M.T., Traver, V., Valero, M.A., Martin, S., Traganitis, A., Mantzourani, E., Totter, A., Karefilaki, K., Paramythis, A., Stephanidis, C., & Robinson, S. (2002). User satisfaction with home telecare based on broadband communication. Journal of Telemedicine and Telecare, 8 (2), 81-90.
- Stephanidis, C., Paramythis, A., Sfyrakis, M. & Savidis, A. (2001). A Case Study in Unified User Interface Development: The AVANTI Web Browser. In C. Stephanidis (Ed.) User Interfaces for All – Concepts, Methods, and Tools (pp. 525-568). Mahwah, NJ: Lawrence Erlbaum Associates.
- Stephanidis, C., Akoumianakis, D., Paramythis, A., & Nikolaou, C. (2000). User Interaction in Digital Libraries: Coping with Diversity through Adaptation. International Journal on Digital Libraries, special issue "Alexandrian Scholars II", 3 (2), 185-205.

Conference and Workshop papers (full paper reviewed)

- König, F., Van Velsen, L., & Paramythis, A. (2009). Finding *My* Needle in the Haystack: Effective Personalized Re-ranking of Search Results in Prospector. Proceedings of the Tenth International Conference on Electronic Commerce and Web Technologies (EC-Web 2009) held in conjunction with DEXA 2009, 1–4 September 2009, Linz, Austria.
- Van Velsen, L., König, F., & Paramythis, A. (2009). Assessing the Effectiveness and Usability of Personalized Internet Search through a Longitudinal Evaluation. Proceedings of the Sixth Workshop on User-Centred Design and Evaluation of Adaptive Systems (UCDEAS) held in conjunction with the International Conference on User Modeling, Adaptation, and Personalization (UMAP 2009), June 26th 2009, Trento, Italy (pp. 44-53). CEUR Workshop Proceedings, ISSN 1613-0073.
- Paramythis, A., & Van Velsen, L. (2009). Lessons learned from the evaluations of an adaptive meta-search engine. Paper presented in the Workshop "Users' Preferences Regarding Intelligent User Interfaces: Differences Among Users and Changes Over Time", held in conjunction with the Intelligent User Interfaces 2009 Conference, February 8th 2009, Sanibel Island, Florida, USA.
- Paramythis, A., & Mühlbacher, J. R. (2008). Towards New Approaches in Adaptive Support for Collaborative E-Learning. Proceedings of the The Eleventh IASTED International Conference on Computers and Advanced Technology in Education (CATE 2008), September 29 – October 1 2008, Crete, Greece.
- Sonntag, M., & Paramythis, A. (2008). Adaptive Feedback for Legal E-Learning. Proceedings of the special track "Computer-based Knowledge & Skill Assessment and Feedback in Learning Settings" at the International Conference on Computer-Aided Learning, 24-26 September 2008, Villach, Austria.
- Paramythis, A., & Cristea, A. (2008). Towards Adaptation Languages for Adaptive Collaborative Learning Support. Proceedings of the Workshop on "Individual and Group Adaptation in Collaborative Learning Environments", held in conjunction with the Third European Conference on Technology Enhanced Learning Maastricht (ECTEL 2008), The Netherlands, September 17-19 2008. CEUR Workshop Proceedings Vol. 384.
- Paramythis, A. (2008). Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet? Proceedings of the Workshop on Adaptive Collaboration Support, held in conjunction with the 5th International Conference on Adaptive

Hypermedia and Adaptive Web-Based Systems (AH'08), 29 July - 1 August 2008, Hannover, Germany (pp. 17 - 29).

- Paramythis, A., König, F., Schwendtner, C., & van Velsen, L. (2008). Using thematic ontologies for user- and group- based adaptive personalization in web searching. Proceedings of the 6th International Workshop on Adaptive Multimedia Retrieval (AMR'2008), June 26-27 2008, Berlin, Germany.
- Paramythis, A. (2006). Self-regulated adaptivity as a design and authoring support tool. Proceedings of the joint 1st International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H) and the 4th International Workshop on Authoring of Adaptive & Adaptable Educational Hypermedia (A3EH), held in conjunction with the Fourth Adaptive Hypermedia and Adaptive Web-Based Systems 2006 Conference (AH 2006), June 21-23 2006, Dublin, Ireland (pp. 355-366).
- Schwendtner, C., König, F., Paramythis, A. (2006). Prospector: An adaptive front-end to the Google search engine. In Klaus-Dieter Althoff and Martin Schaaf, editors, Proceedings of the 14th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS 2006) held in conjunction with Lernen, Wissensentdeckung und Adaptivität (LWA 2006), volume 1/2006 of Hildesheimer Informatik-Berichte, pp. 56–61. University of Hildesheim, Institute of Computer Science, 2006.
- Paramythis, A. (2006). Can adaptive systems participate in their design? Meta-adaptivity and the evolution of adaptive behaviour. Proceedings of the Fourth Adaptive Hypermedia and Adaptive Web-Based Systems 2006 Conference (AH 2006), June 21-23 2006, Dublin, Ireland.
- Paramythis, A., & Weibelzahl, S. (2005). A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In Ardissono, L., Brna, P., & Mitrovic, A. (Eds.), Proceedings of the 10th International Conference on User Modeling (UM2005) , July 24-29 2005, Edinburgh, Scotland, UK (pp. 438-442) (Lecture Notes in Computer Science LNAI 3538, Springer Verlag). Berlin: Springer.
- Paramythis, A., Loidl, S., Mühlbacher, J.R., & Sonntag, M. (2005). A Framework for Uniformly Visualizing and Interacting with Algorithms. In Montgomerie, T.C., & Parker E-Learning, J.R. (Eds.), Proceedings of the IASTED Conference on Education and Technology (ICET 2005), 2-6 July 2005, Calgary, Alberta, Canada (pp. 28 – 33). Calgary: ACTA Press.
- Paramythis, A. (2004). Towards Self-Regulating Adaptive Systems. In Weibelzahl, S., & Henze, N. (Eds.), Proceedings of the Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems of the German Informatics Society (ABIS'04), October 4-5 2004, Berlin, Germany (pp. 57-63).
- Alexandraki, C., Paramythis, A., Maou, N., & Stephanidis, C. (2004). Web Accessibility through Adaptation. In: K. Miesenberger, J. Klaus, W. Zagler, & D. Burger (Eds.), Proceedings of Computers Helping People with Special Needs: 9th International Conference (ICCHP 2004), July 7-9 2004, Paris, France (pp. 302 - 309). Berlin / Heidelberg: Springer, Lecture Notes in Computer Science, Volume 3118 / 2004.
- Paramythis, A., & Loidl-Reisinger, S. (2003). Adaptive Learning Environments and e-Learning Standards. In R. Williams (Ed.), Proceedings of the 2nd European Conference on e-Learning (ECEL2003), Glasgow, Scotland, 6-7 November (pp. 369-379). Academic Conferences International Reading 2003; ISBN: 0-9544577-4-9 (2003)
- Loidl-Reisinger, S., & Paramythis, A. (2003). Distance Education – A Battlefied for Standards. In A. Szücs, E. Wagner, & C. Tsolakidis (Eds.), The Quality Dialogue. Integrating Quality Cultures in Flexible, Distance and eLearning; Proceedings of the 2003 EDEN Annual Conference, 15-18 June, Rhodes, Greece (pp.89-94).
- Paramythis, A., Totter, A., & Stephanidis, C. (2001). A modular approach to the evaluation of Adaptive User Interfaces. In S. Weibelzahl, D. Chin & G. Weber (Eds.),

Proceedings of the Workshop on Empirical Evaluations of Adaptive Systems, held in the context of the 8th International Conference on User Modeling (UM'2001), 13-17 July 2001, Sonthofen, Germany (pp.9-24). Freiburg: Pedagogical University of Freiburg.

- Grammenos, D., Paramythis, C., & Stephanidis, C. (2000). Designing the User Interface of an Interactive Learning Environment for Children. In C. Stephanidis (Ed.), Proceedings of the ERCIM WG UI4ALL one-day joint workshop with i3 Spring Days 2000 on "Interactive Learning Environments for Children", March 3 2000, Athens, Greece (22 pages).
- Stephanidis, C., Paramythis, A., & Sfyrakis, M. (1999). Evaluating adaptable and adaptive user interfaces: Lessons learnt from the development of the AVANTI Web browser. In A. Kobsa & C. Stephanidis (Eds), Proceedings of the 5th ERCIM Workshop "User Interfaces for All", 28 November – 1 December 1999, Dagstuhl, Germany (pp. 193-199). Sankt Augustin: GMD.
- Akoumianakis, D., Sfyrakis, M., Paramythis, A., & Stephanidis, C. (1999). Universal accessibility: Process-oriented design guidelines. In H.-J. Bullinger & J. Ziegler (Eds.), Human-Computer Interaction: Communication, Cooperation, and Application Design [Proceedings of the 8th International Conference on Human-Computer Interaction (HCI International '99)] , 22-26 August 1999, Munich, Germany (vol. 2, pp. 807-811). London: Lawrence Erlbaum Associates.
- Stephanidis, C., Paramythis, A., Akoumianakis, D., & Sfyrakis, M. (1998). Self-Adapting Web-based Systems: Towards Universal Accessibility. In C. Stephanidis & A. Waern (Eds.), Proceedings of the 4th ERCIM Workshop "User Interfaces for All", Stockholm, Sweden, 19-21 October 1998 (17 pages).
- Stephanidis, C., Akoumianakis, D., & Paramythis, A. (1998). User Interaction in Digital Libraries: Coping with Diversity through Adaptation. In panel session "Interaction Design in Digital Libraries", 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL '98) , 21-23 September 1998, Heraklion, Crete, Greece (pp. 717-735). Berlin: Springer, Lecture Notes on Computer Science, 1513.
- Stephanidis, C., Paramythis, A., Sfyrakis, M., Stergiou, A., Maou, N., Leventis, A., Paparoulis, G., & Karagiannidis, C. (1998). Adaptable and Adaptive User Interfaces for Disabled Users in AVANTI Project. In S. Trigila, A. Mullery, M. Campolargo, H. Vanderstraeten & M. Mampaey (Eds.), Intelligence in Services and Networks: Technology for Ubiquitous Telecommunications Services - Proceedings of the 5th International Conference on Intelligence in Services and Networks (IS&N '98), Antwerp, Belgium, 25-28 May 1998 (pp. 153-166). Berlin: Springer, Lecture Notes in Computer Science, 1430.
- Stephanidis, C., Paramythis, A., Karagiannidis, C., & Savidis, A. (1997). Supporting Interface Adaptation: The AVANTI Web-Browser. In C. Stephanidis & N. Carbonell (Eds.), Proceedings of the 3rd ERCIM Workshop "User Interfaces for All", 3-4 November 1997, Obernai, France (14 pages).
- Stephanidis, C., Paramythis, A., Savidis, A., Sfyrakis, M., Stergiou, A., Leventis, A., Maou, N., Paparoulis, G., Karagiannidis, C. (1997). Developing Web Browsers Accessible to All: Supporting User-Adapted Interaction. In G. Anogiannakis, C. Bühler, & M. Soede (Eds.), Proceedings of the 4th European Conference for the Advancement of Assistive Technology (AAATE '97) , 29 September - 2 October 1997, Porto Carras, Greece (pp. 233-237). Amsterdam: IOS Press.
- Savidis, A., Paramythis, A., Akoumianakis, D., & Stephanidis, C. (1997). Designing user-adapted interfaces: the unified design method for transformable interactions. In Proceedings of the ACM Conference on Designing Interactive Systems: Processes, Methods and Techniques (DIS '97), 18-20 August 1997, Amsterdam, The Netherlands (pp. 323-334). New York: ACM Press.

- Stephanidis, C., Paramythis, A., & Grammenos, D. (1995). Communication Aid Interface for Speech-Motor Impaired users. In Proceedings of the 3rd European Conference on the Advancement of Rehabilitation Technology (ECART-3) , 10-13 October 1995, Lisbon, Portugal (pp. 100-102).

Conference and Workshop papers (abstract reviewed)

- Emiliani, P.L., Paramythis, A., Burzagli, L., & Stephanidis, C. (2003). PALIO as an enabling platform for disabled and elderly people. In C. Stephanidis, (Ed.), Universal Access in HCI: Inclusive Design in the Information Society – Volume 4 of the Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003), , 22-27 June 2003, Crete, Greece (pp. 547 - 551). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Paramythis, A., Alexandraki, C., Segkos, I., Maou, N., & Stephanidis, C. (2003). Personalisable, context-aware services: the PALIO approach. In C. Stephanidis, (Ed.), Universal Access in HCI: Inclusive Design in the Information Society – Volume 4 of the Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003) , 22-27 June 2003, Crete, Greece (pp. 582 - 586). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Paramythis, A., Alexandraki, C., Segkos, I., Maou, N., & Stephanidis, C. (2003). The PALIO framework for hypermedia adaptations. In C. Stephanidis, (Ed.), Universal Access in HCI: Inclusive Design in the Information Society – Volume 4 of the Proceedings of the 10th International Conference on Human-Computer Interaction (HCI International 2003) , 22-27 June 2003, Crete, Greece (pp. 587 - 591). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Paramythis, A., Savidis, A., & Stephanidis, C. (2001). AVANTI: A universally accessible web browser. In C. Stephanidis (Ed.), Universal Access in HCI: Towards an Information Society for All – Volume 3 of the Proceedings of the 9th International Conference on Human-Computer Interaction (HCI International 2001), 5-10 August 2001, New Orleans, Louisiana, USA (pp. 91-95). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Zarikas, V., Paramythis, A., & Stephanidis, C. (2001). Decision-theoretic approaches to user interface adaptation: Implications on Universal Access. In C. Stephanidis (Ed.), Universal Access in HCI: Towards an Information Society for All – Volume 3 of the Proceedings of the 9th International Conference on Human-Computer Interaction (HCI International 2001), 5-10 August 2001, New Orleans, Louisiana, USA (pp. 127-131). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Paramythis, A., Leidermann, F., Weber, H., & Stephanidis, C. (2001). Continuity through User Interface Adaptation: A perspective on Universal Access. In C. Stephanidis (Ed.), Universal Access in HCI: Towards an Information Society for All – Volume 3 of the Proceedings of the 9th International Conference on Human-Computer Interaction (HCI International 2001), 5-10 August 2001, New Orleans, Louisiana, USA (pp. 411-415). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Paramythis, A., Weber, H., Leidermann, F., Totter, A., & Stephanidis, C. (2001). Continuous interaction through user interface adaptation: The AVANTI scenario. In G.J. Doherty, M. Massink & M.D. Wilson (Eds), Continuity in Future Computing Systems, Proceedings of the i3 Spring Days Workshop, 23-24 April 2001, Porto, Portugal (pp. 113-115). Didcot, UK: Council for the Central Laboratory of the Research Councils.
- Paramythis, A., Sfyrakis, M., Savidis, A., & Stephanidis, C. (1999). Non-visual Web browsing: Lessons learned from the AVANTI case study. In H.-J. Bullinger & J. Ziegler (Eds.), Human-Computer Interaction: Communication, Cooperation, and Application Design [Proceedings of the 8th International Conference on Human-Computer

Interaction (HCI International '99)], 22-26 August 1999, Munich, Germany (vol. 2, pp. 812-817). London: Lawrence Erlbaum Associates.

- Stephanidis, C., Akoumianakis, D., Sfyrakis, M., & Paramythis, A. (1998). Universal accessibility in HCI: Process-oriented design guidelines and tool requirements. In C. Stephanidis & A. Waern (Eds.), Proceedings of the 4th ERCIM Workshop on "User Interfaces for All", 19-21 October 1998, Stockholm, Sweden (15 pages).
- Stephanidis, C., Savidis, A., & Paramythis, A. (1997). Addressing Cultural Diversity through Unified Interface Development. In G. Salvendy, M.J. Smith & R.J. Koubek (Eds.), Design of Computing Systems: Cognitive Considerations [Proceedings of the 7th International Conference on Human-Computer Interaction (HCI International '97)], 24-29 August 1997, San Francisco, USA (vol. 1, pp. 165-168). Amsterdam: Elsevier, Elsevier Science.
- Kouroupetroglou, G., Paramythis, A., Koumpis, A., Viglas, C., Anagnostopoulos, A., & Frangouli, H. (1995). Design of Interpersonal Communication Systems Based on a Unified User Interface Platform and a Modular Architecture. Proceedings of the TIDE Workshop on User Interface Design for Communication Systems, July 7 1995, Brussels, Belgium (pp. 8-17).

Invited Talks

- Paramythis, A. (2008). Adaptivity "2.0": Challenges on the Road ahead to Next-Generation Adaptive Systems. Keynote address at the 16th Workshop on Adaptivity and User Modeling in Interactive Systems (LWA ABIS 2008), 6.-8. October 2008.
- Stephanidis, C., & Paramythis, A. (1998). Facilitating Access of People with Disabilities to Computer-based Applications and Telematic Services. Workshop on "Information Technologies in the Vocational Training of Disabled People", organized on the occasion of the completion of the HESTIA Project, Horizon Programme, June 30, National and Kapodistrian University of Athens, Athens, Greece.
- Stephanidis, C., & Paramythis, A. (1998). Adaptable and Adaptive User Interfaces for Web-based Interaction. 10th ACTS Concertation Meeting, Applications and Security Cluster, User Interfaces Workshop, May 5 1998, Brussels, Belgium.
- Stephanidis, C., & Paramythis, A. (1997). Adaptation of Services for Different Users: The User Interface. 8th ACTS Concertation Meeting, Domain 5 Meeting, Workshop on Services, December 9 1997, Brussels, Belgium.

APPENDIX B: EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, im Oktober 2009

Alexandros Paramythis