A Dissertation Presented to the Faculty I of the University of Trier
In Partial Fulfillment of the Requirements for the Degree
Doktor der Naturwissenschaften

Evaluation of Adaptive Systems

by

Stephan Weibelzahl

Graduate Programme Human and Machine Intelligence

Kunzenweg 21

University of Education Freiburg

79117 Freiburg

October 2002

Stephan Weibelzahl
Evaluation of Adaptive Systems
Trier, 2003

Gutachter: Prof. Dr. Gerhard Weber, Prof. Dr. Karl F. Wender, Prof. Dr. Anthony Jameson
Dissertationsort: Trier

Contents

List of Figures

List of Tables

Preface

This dissertation project was started in October 1999 when I moved to Freiburg on a scholarship. Since then several people have gotten involved in my project and it is my pleasure to thank them for their support.

First of all, very special thanks go to Gerhard Weber, who gave me invaluable feedback whenever I needed it, and who supported me in every stage of the project. This book would probably not exist without his outstanding scientific experience and wisdom. His award-winning authoring system NetCoach was the basis of all the evaluation studies that are presented in this work.

The scholarship of the German Research Foundation (DFG) provided me with a wide scope of possibilities, thus, I was allowed to explore what I was interested in. What could be better? The graduate school on Human and Artificial Intelligence let me get a glimpse of very different research areas in computer science, psychology, cognitive science, and linguistics. I would also like to thank Gerhard Strube that I was allowed to attend the center of cognitive science at the University of Freiburg for about five months. His doctoral seminar that I visited for the last three years has been an interesting discussion forum for various topics of cognitive science and theory of science.

Many people gave me thoughtful advice and feedback on early dissertation proposals and drafts of this thesis: Daniel Friedmann, Eelco Herder, Anthony Jameson, Hans-Christian Kuhl, Stefan Lippitsch, Pia Schnorpfeil, Dr. Pia Weibelzahl, and many others. Many anonymous reviewers had useful remarks of great worth on the papers that I submitted to several conferences and workshops, which certainly improved the quality of this work.

Finally, I would like to thank Pia and Luisa for all the things they have done to make it possible for me to complete this thesis.

Stephan Weibelzahl
October 2002

Introduction

> "Adaptive systems are systems which can alter aspects of their structure, functionality or interface in order to accommodate the differing needs of individuals or groups of users and the changing needs of users over time" (Benyon and Murray, 1993, p. 199).

In the early beginnings of the adaptive systems development this new approach of software individualization promised to improve human-computer interaction considerably. Many frequently occurring problems of usability and learnability seemed to be easily solved. However, even today, after elaborating significantly the modeling techniques, it is not obvious whether this promise has been kept, because only few empirical evidence exists that supports this claim.

In fact, empirical evaluations of adaptive systems are hard to find—e.g., only a quarter of the articles published in *User Modeling and User Adapted Interaction* (UMUAI) report significant empirical evaluations (Chin, 2001). Many of them include a simple evaluation study with small sample sizes and often without any statistical methods. Several reasons have been identified as responsible for this absence (e.g., Eklund, 1999; Höök, 2000). Besides some structural problems (e.g., short development cycle) one of the major issues is methodological: What has to be done to guarantee the success of adaptation? Straightforward approaches (e.g., asking the users whether they enjoyed the interaction) frequently failed to proof an advantage of adaptive systems or suffer from low test quality.

Aim

The aim of this PhD thesis is to explore a methodology for the empirical evaluation of adaptive systems. Such a methodology consists of at least two components: First, a group of criteria that are proofed to be reliable and valid to measure adaptivity success. Probably, only a combination of several criteria will be adequately meaningful. Secondly, a specification of experimental designs and procedures is needed to apply those criteria.

The proposed approach is designed to be independent of the domain. We certainly do not ignore the fact, that there are domain specific differences between systems (i.e., there are criteria that evaluate system specific goals). However, we claim that such a general approach yields a methodology that is transferable to many systems and would enable researchers to

- find system deficits and failures, e.g., to uncover wrong assumptions about the user in the user model
- show that adaptivity in their system is useful and successful
- justify the efforts spent on making systems adaptive, because the development of an adaptive system still requires more exertion than building a non-adaptive system, though software tools simplified the implementation considerably
- point out deficits of non-adaptive systems, because the comparison of adaptive and non-adaptive versions of a system could also identify problems of the standard interface.

Overview

After introducing and defining adaptivity (Chapter 1) and software evaluation (Chapter 2), we offer a synopsis of current evaluations (Chapter 3) to outline the state of the art. We argue that few studies comply with methodological standards. Several reasons for this deficit are identified (Section 3.3).

Based on this overview of current evaluations, we introduce a framework for the evaluation of adaptive systems (Chapter 4). This framework defines four layers that have to be evaluated separately to guarantee the success of adaptivity.

The framework is then applied to the evaluation of an adaptive learning system—the *HTML-Tutor*—to demonstrate the framework's usefulness (Chapter 5). Several studies have been conducted to evaluate each layer.

Finally, the framework is discussed in terms of its applicability, advantages and limitations (Chapter 6).

Note that several chapters are based on workshops, conferences, and journal papers that have been published by the author previously. These articles include (in chronological order)

- Weibelzahl, S. and Weber, G. (2000). Evaluation adaptiver Systeme und Verhaltenskomplexität. In Müller, M. E. (Ed.), *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, ABIS-2000*, Osnabrück.

- Weibelzahl, S. and Lauer, C. U. (2001). Framework for the evaluation of adaptive CBR-systems. In Vollrath, I., Schmitt, S., and Reimer, U. (Eds.), *Experience Management as Reuse of Knowledge. Proceedings of the Ninth German Workshop on Case Based Reasoning, GWCBR2001*, pages 254–263. Baden-Baden: Shaker.

- Weibelzahl, S. (2001). Evaluation of adaptive systems. In Bauer, M., Gmytrasiewicz, P. J., and Vassileva, J. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 292–294. Berlin: Springer.

- Weber, G., Kuhl, H.-C., and Weibelzahl, S. (2001). Developing adaptive internet based courses with the authoring system NetCoach. In Reich, S., Tzagarakis, M. M., and de Bra, P. (Eds.), *Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 226–238. Berlin: Springer.

- Weibelzahl, S. and Weber, G. (2001). A database of empirical evaluations of adaptive systems. In Klinkenberg, R., Rüping, S., Fick, A., Henze, N., Herzog, C., Molitor, R., and Schröder, O. (Eds.), *Proceedings of Workshop Lernen — Lehren — Wissen — Adaptivität (LLWA 01); research report in computer science nr. 763*, pages 302–306. University of Dortmund.

- Weibelzahl, S. and Weber, G. (2001). Mental models for navigation in adaptive web-sites and behavioral complexity. In Arnold, T. and Herrmann, C. S. (Eds.), *Proceedings of the Fifth Annual Meeting of the German Cognitive Science Society, KogWis 2001*, pages 74–75. Leipzig: Leipziger Universitätsverlag.

- Weibelzahl, S. and Weber, G. (2002). Adapting to prior knowledge of learners. In de Bra, P., Brusilovsky, P., and Conejo, R. (Eds.), *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 448–451. Berlin: Springer.

- Weibelzahl, S., Lippitsch, S., and Weber, G. (2002). Advantages, opportunities, and limits of empirical evaluations: Evaluating adaptive systems. *Künstliche Intelligenz*, 3/02, 17–20.

- Weibelzahl, S., Lippitsch, S., and Weber, G. (2002). Supporting the authoring of Adaptive Hypermedia with structural information? In Henze, N., Kókai, G., Schröder, O., and Zeidler, J. (Eds.), *Personalization for the Mobile World. Proceedings of the German Workshop on Adaptivity and User Modeling in Interactive Systems, ABIS02*, pages 99–105, Hannover.

These peer reviewed papers were a good starting point for integrating these various aspects in a complete dissertation on the evaluation of adaptive systems.

1. Adaptive Systems

1.1. Functions and Definition of Adaptivity

The idea of individualizing software with user models can be traced back to the early 80's. While systems traditionally considered only one typical user or a few user stereotypes, it became more and more popular to see users as individuals with idiosyncratic preferences, needs, and tasks (Rich, 1983). Nielsen (1989) even found that the top three effects of the largest effects found in hypertext systems were due to individual differences between users. Thus, an individualized system promised to solve the most urging problems. Since these first ideas, many systems have been developed in different domains, solving different problems with varying degrees of adaptation.

Before we give an overview of these different functions of existing systems, it is imperative to define more precisely what adaptive systems are and delimit them from similar concepts.

According to Oppermann (1994) a system is called adaptive "if it is able to change its own characteristics automatically according to the user's needs" (p. 456). Adaptive systems consider the way the user interacts with the system and modify the interface presentation or the system behavior accordingly. Jameson (2001) adds an important characteristic:

> A user-adaptive system is an interactive system which adapts its behavior to each individual user on the basis of nontrivial inferences from information about that user (Jameson, 2001, p. 4).

In this definition adaptivity is limited to nontrivial inferences to exclude straightforward adaptations (e.g., a user might set the font color of the interface to blue and thus the system might display the font in blue). However, it is obvious that this kind of adaptations is trivial and is used regularly in all kinds of systems. We would thus refrain from calling this behavior adaptive. In the remainder of this book a system

adaptivity

is called adaptive only if it is an interactive system that changes its behavior depending on the individual user's behavior on the basis of nontrivial inferences from information about the user.

Comparing this definition to those first two definitions it is important to note that we included an additional requirement: adaptive systems receive the information about the user from observations of the user. This is in accordance with Jameson's model of adaptation (Jameson, 2001) and we think that it is important to mention this fact in the definition.

adaptability

Adaptivity is often confused with adaptability. A system is called adaptable, "if it provides the user with tools that make it possible to change the system characteristics" (Oppermann, 1994, p. 455). For example, adaptable systems are not based on intelligent algorithms that infer how to adapt on their own, rather they offer the flexibility to change the interface or the behavior manually according to one's needs or preferences. The adaptation decision is left to the user.

personalization

Both adaptivity and adaptability are often summarized by the term personalization (Jameson, 2001). Especially in e-commerce the demand for individually tailored products and services is growing constantly and adaptability as well as adaptivity are used increasingly.

functions

Adaptive Systems are used in many domains to solve different tasks. The following list of functions, adopted from Jameson (2001), is neither meant to be complete, nor does it describe distinct categories. Rather it should outline how adaptivity is applied in different domains and why it might be feasible to use adaptive systems for a specific task.

Help the user to find information: When searching large information spaces such as the web or literature databases, users are frequently either overwhelmed by the amount of retrieved documents or do not get any results because the query was too narrow. For example, by taking into consideration the user's relevance feedback on previous retrievals it is possible to improve both precision and recall of retrieved documents (Vogt, Cottrell, Belew and Bartell, 1999).

Tailor information to user: Billsus and Pazzani (1999) introduced an adaptive system that compiles a daily news program that is tailored to the individual preferences. These preferences are automatically learned from feedback in previous interactions. Electronic shops might tailor the product description and the way of presentation to the customer's needs and expertise (Jörding, 1999).

Recommend products: Adaptive e-commerce systems are an important field of application. Building a user model of the customer's needs and preferences enables the system to customize the sales interaction and to suggest suitable products (Ardissono and Goy, 1999). An electronic shopping guide might even be aware of the user's current location in the shop (Bohnenberger, Jameson, Krüger and Butz, 2002).

Help with routine tasks: Frequently occurring interactive tasks such as sorting incoming e-mail or formatting the layout of paragraphs can be supported by adaptive systems (e.g., Cohen, 1996; Segal and Kephart, 1999).

Adapt an interface: Usually the visual interface, i.e., the screen or display, is adapted; however, for users with motor disabilities it can be useful to adapt the input interface. A system that learns and models the user's keyboard skills may minimize or eliminate keystroke errors (Trewin and Pain, 1997).

Give help: Depending on the user's expertise or knowledge a system can provide help on commands (Chin, 1989). A well known example of help provision based on the users background, actions, and queries is the LUMIÈRE project that developed lifelike characters who assist in the interaction with word processing software (Horvitz, Breese, Heckerman, Hovel and Rommelse, 1998).

Support learning: There are many systems that support the learning process both with standalone applications (Weber and Möllenberg, 1995) as well as with Internet based courses and trainings (Brusilovsky, Eklund and Schwarz, 1998; de Bra and Calvi, 1998; Henze, Nejdl and Wolpers, 1999). Frequently applied methods for adaptation to the learner include adaptive annotation of links, adaptive hiding of links, and adaptive curriculum sequencing in dependence of the learners current knowledge. An overview of different adaptation methods and systems can be found in Brusilovsky (1996) and Brusilovsky (2001).

Conduct a dialog: The robustness of automatic dialogs via telephone (e.g., a ticket service) will be enhanced if the user's intentions are modeled (Horvitz and Paek, 2001).

Support collaboration: By modeling the user's goals, interests, and availability it may become easier to find collaborators in a distributed workspace environment (Bull, Greer, McCalla, Kettel and Bowes, 2001; Greer, McCalla, Collins, Kumar, Meagher and Vassileva, 1998).

Though adaptive systems perform very different tasks and adapt in very different ways it is possible to subsume them in a single model that describes the architecture of these systems abstractly.

1.2. Models of Adaptivity

In addition to the definition of adaptive systems above, this chapter will derive a model of adaptivity. The evaluation framework that is proposed in this thesis is supposed to hold for all adaptive systems. Thus, a clear model is required that enables researches to categorize their systems and which is a prerequisite for the comparison of different systems.

existing architectures

We will introduce three models (or architectures) of adaptive systems that have been proposed in the literature: a very early architecture by Benyon and Murray (1993), a proposal by Oppermann (1994), and a very similar model proposed by Jameson (2001). These architectures represent different points of views and focus on different aspects. For adaptive hypermedia, several reference models have been developed (e.g., de Bra, Houben and Wu, 1999; Koch and Wirsing, 2002; Ohene-Djan, 2002), but currently these are not generally applicable to adaptive systems and are rather intended to support software engineers in developing systems. Thus, these reference models are not taken into consideration here.

Benyon and Murray's model

Benyon and Murray (1993) introduced an architecture of adaptive systems that focuses on the components of adaptivity. It is designed to support developers in selecting appropriate representation techniques. The architecture basically consists of three main components: the user model, the domain model, and the interaction model (see Figure 1.1).

- The user model represents the system's beliefs about the user. It consists of three interlinking components: First, the student model, which contains the system's assumptions about the user's beliefs about the domain. For example, the system might assume that the user knows how to open a text file. Thus, this information is dependent on the application and the domain. The model's second component is the profile model, which holds information about the user's background, interests and general knowledge. And the third component, the psychological model, holds domain independent cognitive and affective traits of the user.

- The domain model defines the aspects of the system and the world that are important for inferences, e.g., functions that might be altered. These aspects might be described at different levels, such as the task level, the logical level,

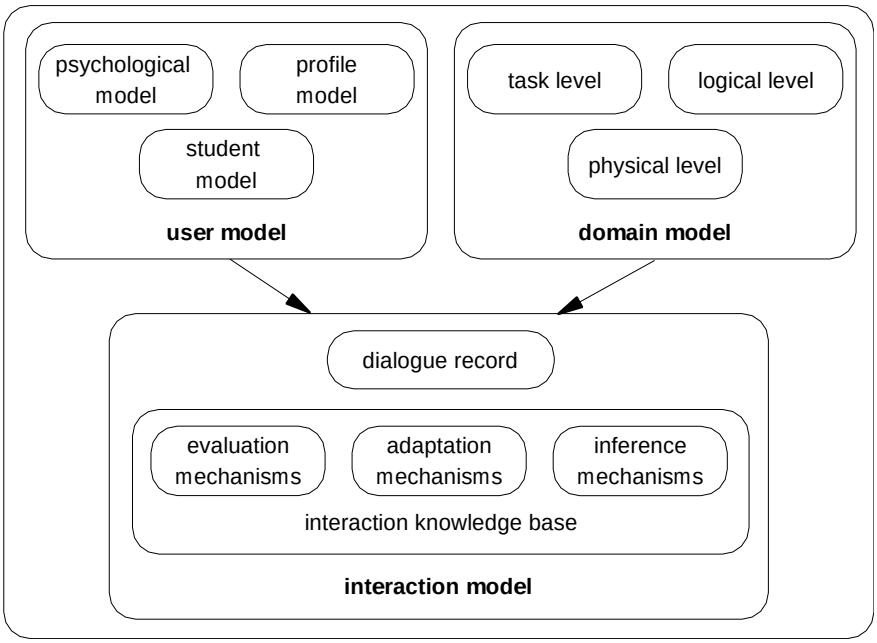


Figure 1.1.: Benyon and Murray's proposal for an architecture of adaptive systems; adopted from Benyon and Murray (1993)

or the physical level. Thus, the domain model is the basis for all inferences and adaptations.

- The interaction model handles the dialog between the user and the application. It might record the previous interaction and contains mechanisms for the inference of user properties, mechanisms for adaptation of the system to these user properties, and mechanisms for an evaluation of the adaptation.

In summary, this model describes the different kinds of information that are required for an adaptation. When proposing this architecture, Benyon and Murray might have had in mind intelligent tutoring systems, with explicit modeling of knowledge or misconceptions. Nevertheless, the model is applicable to other application domains, too. However, in some systems the main components might be not as distinguishable as the authors claim, e.g., in adaptive machine learning, inference mechanisms and the domain model might be mixed up. Moreover, the processes and the interaction between the component are not described. For evaluation purposes it is important to look not only at the system's status, but also at the processes that lead to the current status.

Oppermann's model

A much more process oriented architecture was introduced by Oppermann (1994). Its main idea is to distinguish between an afferential, an inferential, and an efferential component.

- The afferential component gathers the observation data. The system observes the users behavior, e.g., key strokes, commands, errors, movements, or navigation, which is the basis for further adaptation.
- The inferential component is the core of the system. An intelligent mechanism infers user characteristics from the raw data.
- Finally, the efferential component decides how the system should be adapted, i.e., how the system behavior should be changed. The adaptation might concern the presentation of objects, functions or tools, default values for parameters, sequences of dialogue, or system messages.

Jameson's model

A very similar model was proposed by Jameson (2001). The author also distinguishes an afference, called upward inference, on the one hand, and an efference, called downward inference, on the other hand (see Figure 1.2). Thus, the emphasis of this model is on the inference mechanisms, while Oppermann focuses on the components. An important fact that is considered in both models is the distinction between the inference of user characteristics and the concrete adaptation decision. Opposed to Benyon and Murray's model, they do not describe the structure or content of the components in more detail, e.g., different kinds of aspects of the users.

evaluation model

For the evaluation of adaptive systems, a process-oriented model is feasible, because it provides better insights into the actual data processing and thus offers obvious starting points for evaluations. Based on the above models, we propose a new model that is especially designed for evaluation purposes. We focus on the inferences that are involved in the adaptation, but add another information processing step explicitly, namely the user observation. The following sections describe this model, which is shown in Figure 1.3, in more detail.

1.2.1. Acquisition of Input Data

In classical (non-adaptive) systems the user interacts with the machine via the interface by entering input data, which are strictly task related. One of the main characteristics of adaptive systems is, as seen in the definitions above, that they acquire additional input data. The system might observe the user in different ways, e.g., a system might monitor the interaction by recording key strokes or error types, by registering navigation behavior and so forth. Some systems also monitor a kind of

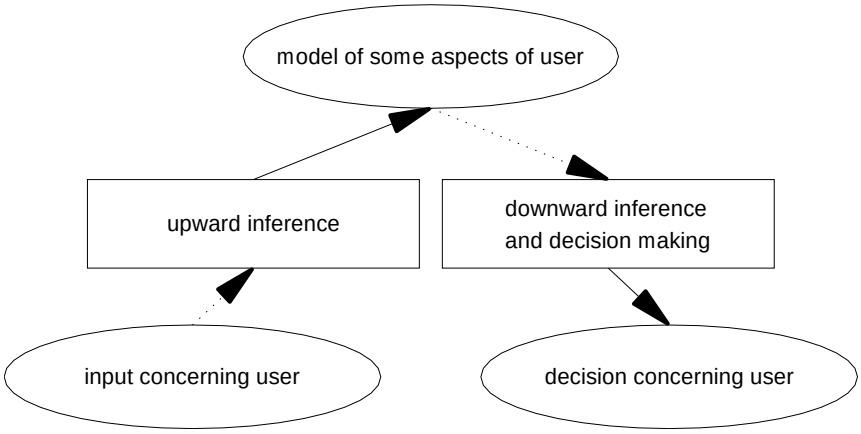


Figure 1.2.: Jameson's proposal for an architecture of adaptive systems; adopted from Jameson (2001)

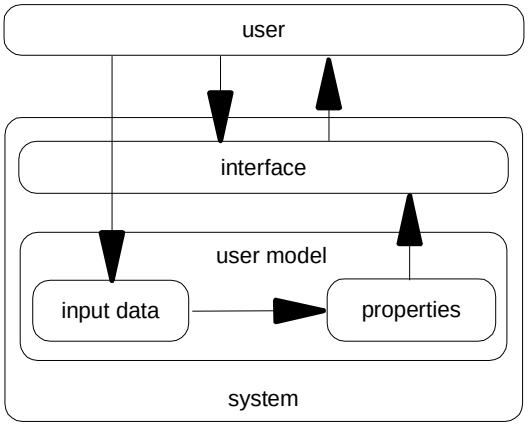


Figure 1.3.: Architecture of adaptive systems and flow of information

metadata, e.g., the frequency of commands, while others ask the user directly. In fact, all these systems gather information that is not strictly required for the task, and might be completely unrelated to the task.

1.2.2. Inference of User Properties

Based on these input data, the system now infers abstract user characteristics. For example, an adaptive learning environment might infer the user's current knowledge (Brusilovsky et al., 1998), an adaptive help system might infer the user's goals or tasks (Horvitz et al., 1998), and an adaptive recommendation system might infer the user's needs or preferences (Linden, Hanks and Lesh, 1997). The key point is that these properties are inferred from the input data and not accessed directly.

The inference is based on Artificial Intelligence (AI) techniques. Standard approaches include Bayesian Networks, machine learning, Case-Based Reasoning, rule based inferences, and combinations of these. In different domains and for different tasks all techniques have their strengths and weaknesses, but currently the most common approach uses inferences that are based on conditional probabilities.

1.2.3. Adaptation Decision

Finally, in the last information processing step, the user properties are used to adapt the interface. Note that the used definition of "interface" is not limited to design issues, but also includes presentation strategies, contents, commands, functions, annotations, etc. Changing the interface also includes altering the behavior of input devices, such as keyboards.

The distinction between user properties and adaptation decision becomes obvious when considering a product recommendation system: If a system infers a preference for a specific product it will probably recommend this object. But the adaptation decision does not stop at this point. There is a variety of ways to recommend a product. The system might either just offer a hint or might limit the assortment. Even no adaptation at all might have to be considered, because some properties start gaining relevance after time. The adaptation decision might consider several different dimensions in the user model, e.g., the product preference might be modulated by the user's knowledge, i.e., while domain experts receive a direct recommendation, novices with the same preference might only get a link.

In summary, in the process of adaptation decision, the system decides about concrete adaptation steps based on the inferred abstract user properties.

This model should be applicable to all adaptive systems. Chapter 4 describes how this model can be used as a heuristic for evaluation studies.

2. Empirical Evaluation

Empirical software evaluations can provide important hints to failures in interactive systems that can not be uncovered otherwise. What are the goals of evaluation? How can software be improved by the results of empirical evaluations? What kind of failures are uncovered and which remain undiscovered? The advantages and limits of empirical evaluations are outlined and usability is introduced as the most important criterion for this kind of evaluations.

2.1. Software Evaluation

definition

The term evaluation is most frequently used for determining the worth or merit of an 'evaluand'. Worthen, Sanders and Fitzpatrick (1997) define evaluation as the "identification, clarification, and application of defensible criteria to determine an evaluation object's value, quality, utility, effectiveness, or significance in relation to those criteria" (p. 5). Sometimes its meaning is limited to the assessment of social intervention programs, such as curricula, only (Rossi and Freeman, 1993). However, for the evaluation of adaptive systems we certainly prefer a broader sense. An evaluation of an interactive system ensures that it behaves as expected by the designer and that it meets the requirements of the user (Dix, Finlay, Abowd and Beale, 1998).

goals

Accordingly, three main goals of software evaluation are distinguished (Dix et al., 1998):

- *To assess the extent of system's functionality.* Does the system comply with the user's requirements? Which functional capabilities are offered to the user? How effectively does the system support the user's task?

- *To assess the effect of the interface on the user.* This goal covers many of the usability aspects introduced in the previous chapter. How easy is the system to learn and is the user satisfied with the system?

- *To identify specific problems with the system.* The last goal can be seen as a kind of feedback loop. Evaluations can give valuable hints for improvements by uncovering unexpected behavior of the system and by identifying incongruencies between user expectations and system design.

formative and summative evaluation

Both, assessing the value of and improving the object are important goals, which is often referred to as formative and summative evaluation (Scriven, 1967). Both aspects will be considered in this work, i.e., software evaluation is not just the last phase in the software development process, but should be seen as an important source of information throughout the complete software life cycle (Nielsen, 1993).

evaluation steps

Performing an evaluation requires an extended procedure. Several evaluation steps have been proposed. Exemplarily, we list the most essential steps for the evaluation of software according to Totterdell and Boyle (1990):

- *Identifying the purposes or objectives of the evaluation.* A well planned study includes a clear specification of the commissioner, the audience who is supposed to receive the results, and most importantly the criteria. In other words, there should be a clear objective before the data are collected.

- *Specifying experimental design.* The criteria need to be translated in suitable methods, subjects, tasks, measurements, experimental settings, and resources. Many different evaluation designs and according techniques have been proposed for different criteria and settings. An overview of evaluation techniques for the evaluation of software and interfaces is given by Howard and Murray (1987) and Dix et al. (1998) among other authors.

- *Collecting results.* Depending on the method, the results are collected by using log-files, behavior observation, and questionnaires.

- *Analyzing data.* Both quantitative and qualitative analysis may be applied. Note that the analysis highly depends on a proper specification of the criteria and methods used. Thus, both methods and criteria will be emphasized throughout this work. It will be shown that the specification of methods for the evaluation of adaptive systems raises specific problems.

- *Drawing conclusions.* Finally, the interpretation of the results may be used to recommend the system or to recommend modifications.

This general procedure structure was used implicitly and explicitly for all of our studies that are described in this work. However, we do not give details for all steps in all studies, because most aspects are covered by common scientific behavior anyway

(e.g., specifying criteria and setting hypotheses), and because many specifications occur again and again.

2.2. Advantages: Why Empirical Evaluations are needed

the need for evaluations

Some areas of Artificial Intelligence apply empirical methods regularly. For example, planning and search algorithms are benchmarked in standard domains, and machine learning algorithms are usually tested with real data sets. However, looking at an applied area such as user modeling, empirical studies are rare. For instance, only a quarter of the articles published in *User Modeling and User Adapted Interaction* (UMUAI) report empirical evaluations of significant scientific value (Chin, 2001). Many of them include a simple evaluation study with small sample sizes and often without any statistical methods.

On the other hand, empirical research is absolutely necessary for an estimation of the effectiveness, efficiency, and usability of a system that applies AI techniques in real world scenarios. Especially user modeling techniques which are based on human-computer interaction require empirical evaluations. Otherwise certain types of errors will remain undiscovered. Undoubtedly, verification, formal correctness, and tests are important methods for software engineering, however, we argue that empirical evaluation—seen as an important complement—can improve AI techniques considerably. Moreover, the empirical approach is an important way to both, legitimize the efforts spent, and to give evidence to the usefulness of an approach.

adaptivity vs. consistency

Of especial interest is the evaluation of adaptive systems, because the potential lack of consistency has been criticized (Benyon, 1993). The flexibility of adaptive systems that is usually praised as their enormous advantage could also be a major threat to usability issues (Edmonds, 1987; Thimbleby, 1990; Woods, 1993). As seen above, the definition of usability includes dimensions such as learnability and memorability. If a system changes its behavior over time it might happen that remembering the functions and command usage become even more difficult, which is the "price of flexibility" (Woods, 1993) that the user has to pay. Obviously formal techniques such as verification cannot solve such subjective psychological issues.

2.3. Limits: Where Empirical Evaluations fail

The hypothesis testing procedure is responsible for an important limitation of empirical research. Empirical studies are very good at identifying design errors and

false assumptions but they do not suggest new theories or approaches directly. Even an explorative study requires some hypotheses about possible impact factors. Thus, empirical evaluations have to be combined with theoretical grounds to yield useful results.

When evaluating adaptive systems—as opposed to AI systems in general—at least two additional problems emerge:

adequate control groups

First, defining adequate control groups is difficult for those systems that either cannot switch off the adaptivity, or where a non-adaptive version appears to be absurd because adaptivity is an inherent feature of these systems (Höök, 2000). Comparing alternative adaptation decisions might relieve this situation in many cases, as this allows for estimates on the effect size that can be traced back to the adaptivity itself. But the underlying problem remains: What is a fair comparison condition for adaptive systems?

adequate criteria

Second, adequate criteria for adaptivity success are not well defined or commonly accepted: On the one hand, objective standard criteria (e.g., duration, number of interaction steps, knowledge gain) regularly failed to find a difference between adaptive and non-adaptive versions of a system. Usually, these criteria have not been proven to be valid indicators of interaction quality or adaptivity success. On the other hand, subjective criteria that are standard in human-computer interaction research (e.g., usability questionnaires, eye tracking) have been very rarely applied to user modeling. Probably, the effects of adaptivity in most systems are rather subtle and require precise measurement.

In summary, empirical research offers a lot of opportunities that could inspire current research in AI in general and in user modeling in particular. Empirical studies are able to identify errors in AI systems that would otherwise remain undiscovered. However, it has been largely neglected so far.

2.4. Usability as Evaluation Criterion

What is bad with the system and why? How good is the system? According to Oppermann (1994) these are the two types of questions that can be answered by empirical evaluations. While the first question assesses the system's absolute quality, the second question implies comparing different alternatives under certain aspects or testing a given system against fixed criteria.

The popular construct usability provides approaches to answer these questions and is thus the most important criterion for software evaluations. The International

definition

Standards Organisation (1998) defines usability in the following way: "A system is usable when it allows the user to achieve his task with effectiveness, efficiency and

satisfaction in a given context of use." A software is usable when the user can achieve his task with a minimum of resources required and when the system is pleasant to use. Thus, an evaluation has to check a least three dimensions of usability and to define criteria for each of them:

dimensions

- Effectiveness: A system is effective if the objectives of the users are achieved and if they can fulfill their individual goals.
- Efficiency: A system is more efficient than another if the resources required to achieve these goals, for example the time needed to achieve the task are limited. Users should be enabled to complete tasks with high productivity.
- Satisfaction: Users are satisfied if the system is pleasant to use, for example the criterion of satisfaction can be the inverse ratio to the number of negative remarks said by the user during the test.

In addition to this standard, several other dimensions of usability have been proposed (e.g., Nielsen, 1993), including:

- Learnability: To smooth the first contact with the system it should be easy to learn, i.e., the usage of commands and functions should be easy to understand.
- Memorability: In addition, the system's functions and commands should also be easy to remember so that the user doesn't have to learn it again when returning after an interval.
- Few and non-catastrophic errors: Users should make only a few errors, when working with the system, and should be able to recover from errors easily.

To measure the usability of an adaptive system we have to define criteria for each dimension. Several criteria have been proposed. The next section lists the criteria and methods that are used in current evaluations.

3. Current Evaluations of Adaptive Systems

In order to explore the state of the art of adaptive systems evaluation, we compiled a synopsis of current evaluation studies. The methods and criteria that have been found in these studies are categorized and problems of current evaluations are identified.

3.1. Systematic Synopsis

The overview of current publications affirmed our claim that only few studies are based on proper experimental designs and statistical methods.

The synopsis is separated into two kinds of entries: experimental studies on the one hand and adaptive systems on the other hand. While each study evaluates one or more systems, a system which is categorized by a specific function and an adaptation mechanism might be evaluated in several studies. A simple illustration of this n:m relation is shown in Figure 3.1. Due to this relation of entities it was impossible to use a standard literature database with some additional meta-tags for information specific for adaptivity. The distinction between evaluation studies and evaluated systems required a new approach.

Each evaluated system is described in terms of its *name*, the *function* it fulfills, the *task* that it performs, and a brief *description of the adaptation mechanism*. This way of characterizing an adaptive system and most of the categories are adopted from Jameson (1999). See Figure 3.2 for a detailed description of the categories.

systems

The *purpose* and the *method of adaptation* are important to help the reader understand what the system does. To find related systems the functioning and task are probably more important.

For each study the synopsis provides a *citation*, a *reference* to the evaluated system, and a detailed description of the *evaluation design* (see Figure 3.3). The criteria have been categorized in efficiency, effectiveness, and usability (Draper, Brown, Henderson and McAteer, 1996; Mark and Greer, 1993). Although the definition

studies

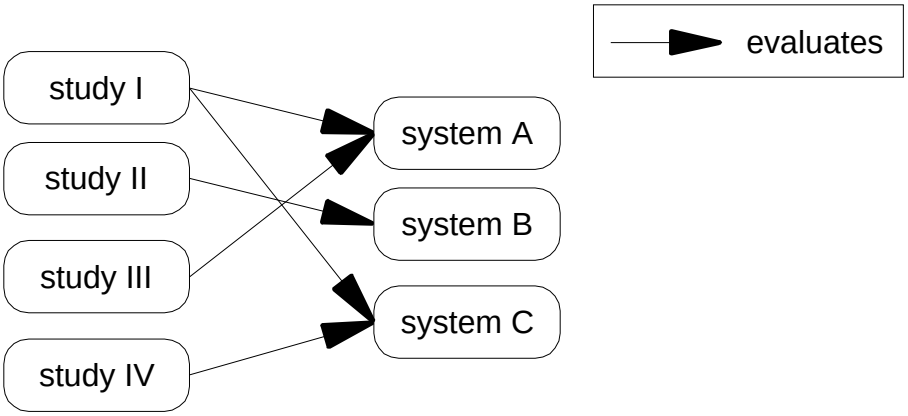


Figure 3.1.: Relation of evaluation studies and evaluated systems illustrated by an example. While one study might evaluate one or more systems (e.g., study I), a system might be evaluated by one or more studies (e.g., system A)

of usability above includes effectiveness and efficiency, in this context we used this categorization to distinguish objective criteria (e.g., duration) from subjective criteria (e.g., rating of satisfaction). The data type entry gives hints about the kind of measures that were used and the way the data were collected (McGrath, 1995).

For the experimental studies, *statistical data* and *methods of analysis* are reported as far as available (see Figure 3.4). The categories of *evaluation method* were also adopted from Jameson (1999). Similar classifications were proposed before (e.g., Runkel and McGrath, 1972; Whitefield, Wilson and Dowell, 1991).

The studies are also categorized in reference to an *evaluation framework* introduced by Weibelzahl (2001). The development and validation of this framework is the main focus of this thesis. In fact, we propose to evaluate four different information processing steps in a so-called layered evaluation (Karagiannidis and Sampson, 2000). Recently, two alternative frameworks have been proposed (Brusilovsky, Karagiannidis and Sampson, 2001; Paramythis, Totter and Stephanidis, 2001) which could serve as additional categorization. At this point it is only important to note that studies may be categorized according to the information processing step they refer to. More information about the framework itself can be found in Chapter 4.

Currently the synopsis contains 43 studies. Most of them are published in the *UMUAI journal* and in proceedings of *User Modeling Conferences* (Brusilovsky and de Bra, 1999; Jameson et al., 1997; Kay, 1999). We claim, that these publi-

Description of System

Name: the system's full name

Function of system: rough category to describe the system. These categories include

- tailor information presentation
- recommend products or other objects
- help user to find information
- support learning
- give help
- adapt interface
- take over routine tasks
- support collaboration
- other function

Task: description of the task user and system perform together. If possible the domain is reported, too.

Adaptation rate: micro or macro adaptation (Cronbach, 1967); basically a question of whether a long-term or a short-term user model is used

Purpose of adaptation: why is adaptation used; which goals or advantages have been the reason for using adaptation

Method of adaptation: in which way does adaptation take place (e.g., selection of appropriate level of difficulty of tasks, change layout of display)

Figure 3.2.: Description of the system categorization

Description of Study

Evaluated system: name of the evaluated system(s)

References: references where the information about the system was drawn from which might differ from the citation of the evaluation study itself

Evaluation layer: according to the framework proposed by Weibelzahl (2001) a study can be assigned to one or more of the following evaluation layers

- evaluation of input data
- evaluation of inference
- evaluation of adaptation decision
- evaluation of interaction

Method of evaluation: a short description of the evaluation, using one of the following categories

- without running system
  - results of previous research
  - early exploratory studies
  - knowledge acquisition from experts
- studies with a system
  - controlled evaluations with users
  - controlled evaluations with hypothetical (i.e., simulated) users
  - experience with real world use

Data type: brief description of the kind of analyzed data (e.g., observed behavior, questionnaire, interview, log-files, etc.)

Criteria: which were the main criteria, and which measures where used, if possible measures (e.g., elapsed time) are grouped in reference to the abstract criterion (e.g., user satisfaction)

Criteria categories: one or more of the following categories apply if at least one of the criteria belong to it

- efficiency
- effectiveness
- usability

Figure 3.3.: Description of the study categorization

Description of Experimental Study

N: number of subjects, sample size
k: number of groups or conditions
randomization: is the assignment of subjects to groups randomized or quasi-experimental
statistical analysis: which statistical methods are used, e.g., analysis of variance (ANOVA), multivariate analysis of variance (MANOVA), correlation

Figure 3.4.: Additional information for experimental studies

cations contain the most important work in the area of adaptive systems. Probably there are evaluation studies that are published in other journals or proceedings, but nevertheless, the coverage of this synopsis should be considerably high.

3.2. Current Methods and Criteria

To obtain an overview of the state of the art, the systematic overview has been analyzed in terms of the criteria that were used in the studies. The Tables 3.1 and 3.2 show which criterion was found in which study. In addition the studies are categorized in accordance with the type of evaluation (see Section 3.1).

evaluation methods

Most of the studies evaluate a *running system*, while only few *early exploratory studies* were identified. The categories *results of previous research* and *knowledge acquisition from experts* remained empty. This disproportionalness might result from the fact that the latter kinds of studies are more difficult to detect than experimental evaluations, i.e., some studies might have been dropped during literature search. Moreover, domain experts are probably frequently consulted in an informal way, but the results are not reported. Thus *knowledge acquisition from experts* might be more common than suggested by this overview. And of course, the synopsis is neither exhaustive nor representative, although it probably covers most of the current research.

sample sizes

Table 3.3 categorizes all studies according to their sample size and to the statistical methods that have been used. *Accuracy measures*, i.e., precision and recall, are listed as separate category, because these criteria are clearly not an inference statistical method, but they are adequate for evaluating document retrieval systems—opposed to reporting means without any further analysis. In information retrieval studies that

examined several users (e.g., [31]) it would be preferable to have additional measures of the results' certainty.

From a methodological point of view we have to accept that about a quarter (11/43) of the studies examines either only a single user, hypothetical users, or the sample size is not reported (Table 3.3). Thus, these studies have very limited value for evaluating adaptivity success. Representative samples are not required, but generalized predictions are impossible with minimal sample sizes. Hypothetical users help to verify that the system behaves in the expected way, but general statements are difficult here, too. Not to report the sample size at all, as found in four studies, should be considered a malpractice, because interpreting the results becomes somewhat arbitrary.

In fact, only 14 out of 43 studies are of high quality in terms of sample size and statistical analysis. Most of them have been carried out for adaptive learning systems, probably because learning gain cannot be evaluated other than empirically, and education has a tradition of empirical approaches.

criteria

A wide range of criteria has been found. The most frequent measures include *accuracy*, *domain knowledge*, and *duration of interaction*. As stated above, evaluation of learning systems are more frequent, and usually these evaluations assess both the learners' knowledge and the duration that was required to acquire this knowledge. Precision and recall of information retrieval are probably very common, because the quality of such a system obviously depends on these measures. Other criteria have been used in one to three studies each. This is an insufficient base for comparing approaches or inference mechanisms across studies. At best, it would be possible to compare information retrieval systems and learning systems respectively. In fact, some recent discussions[1] have aimed at establishing a competition of implementing the same learning content with different approaches.

The criteria can also be categorized in terms of the class of measure (McGrath, 1995). Table 3.4 shows which measures are self-reports, observations, or archival records. While not surprisingly no archival records were used at all, only few of the criteria assess the user's subjective experience with self-reports. These include *difficulty of learning*, *rating of solution quality*, *subjective rating of effect*, *usability questionnaire*, and *user satisfaction*. Only the usability questionnaires have been standardized and externally validated. All other self-report measures are more or less ad hoc questions or rating scales. Effect sizes of different treatments or reliability measures are not known.

self-reports

[1] see e.g., discussion on possible competition areas at the Workshop of Adaptive Systems for Web-Based Education (http://sirius.lcc.uma.es/WASWE2002/) at the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems (AH2002) or the Learning Open (http://www.LearningOpen.com) project of Neil Heffernan.

Table 3.1.: Part I of criteria and designs of studies that have been included in the synopsis. The numbers in brackets (e.g., [1]) refer to the alphabetical index in Figure 3.5 on page 40. A study might of course use more than one criterion. See also 3.2 for studies with running system

	studies without running system		
	results of previous research	early exploratory studies	knowledge acquisition from experts
accuracy, precision, and recall	–	–	–
accuracy of system prediction	–	[9]	–
amount of required help	–	–	–
amount of requested material	–	–	–
budget spent	–	–	–
correct categorization of users	–	–	–
computation time	–	–	–
difficulty of learning	–	–	–
domain knowledge; learning gain	–	–	–
duration of interaction	–	–	–
fixation times	–	–	–
number of communications	–	–	–
number of errors	–	–	–
number of navigation steps	–	–	–
overall impression	–	–	–
rating of solution quality	–	–	–
similarity of expert rating – system decision	–	[19], [41]	–
subjective rating of effect	–	–	–
stability of user model	–	–	–
system preference	–	–	–
task success	–	–	–
usability questionnaire	–	–	–
user satisfaction	–	–	–

continued in Table 3.2 on page 38

Table 3.2.: Part II of criteria and designs of studies that have been included in the synopsis. The numbers in brackets (e.g., [1]) refer to the alphabetical index in Figure 3.5 on page 40. A study might of course use more than one criterion. See also 3.1 for studies without running system

	studies with running system		
	controlled evaluations with users	controlled eval. with hypothetical users	experience with real world use
accuracy, precision, and recall	[5], [8], [20], [21], [25], [31], [38]	[42]	[6]
accuracy of system prediction	[10]	–	–
amount of required help	–	–	[3], [24]
amount of requested material	[17], [32], [33], [34], [43]	–	–
budget spent	[7]	–	–
correct categorization of users	[15], [28]	[4]	–
computation time	–	[22]	–
difficulty of learning	–	–	[3]
domain knowledge; learning gain	[11], [12], [13], [27], [32], [33], [34]	–	[24], [40]
duration of interaction	[7], [11], [12], [13], [32], [33], [34]	–	–
fixation times	[14]	–	–
number of communications	[23]	[22]	–
number of errors	[23]	[39]	–
number of navigation steps	[43]	–	–
overall impression	–	–	[6]
rating of solution quality	[1], [2], [26]	–	–
similarity of expert rating – system decision	[16], [18], [35]	–	–
subjective rating of effect	–	–	[3]
stability of user model	–	–	[37]
system preference	[2], [30]	–	–
task success	[23], [29]	–	–
usability questionnaire	[23], [36]	–	–
user satisfaction	[16], [17]	–	[3], [6], [40]

Table 3.3.: Sample sizes of evaluation studies. All studies in the synopsis were categorized according to their sample size. The numbers in brackets (e.g., [1]) refer to the alphabetical index in Figure 3.5 on page 3.5

sample size	no statistical analysis	accuracy measures	inference statistics	Σ
unavailable	[1], [9], [40], [41]	–	–	4
hypothetical	[4], [22]	[42]	–	3
1	[7], [18]	[20]	[26]	4
2 – 20	[2], [3], [16], [17], [19]	[5], [6], [21], [25]	[10], [11], [23], [30]	13
16 – 50	[39]	[31]	[12], [13], [14], [24], [28], [33], [36], [37], [43]	11
$>$ 50	[35]	[38]	[8], [15], [27], [29], [32], [34]	8
Σ	15	8	20	**43**

1. Ambrosini, Cirillo and Micarelli (1997), HUMOS and WIFS
2. Bares and Lester (1997), UCam
3. Beck, Stern and Woolf (1997), MFD
4. Berthold and Jameson (1999), READY
5. Billsus and Pazzani (1999), News Dude
6. Bueno and David (2001), METIORE
7. Chin and Porage (2001), Iona
8. Chiu, Webb and Kuzmycz (1997), FOIL-IOAM
9. Chu-Carroll and Brown (1998), Initiative Prediction
10. Corbett and Bhatnagar (1997), APT
11. study I in Corbett (2001), APT
12. study II in Corbett (2001), APT
13. study III in Corbett (2001), APT
14. Crosby, Iding and Chin (2001)
15. Draier and Gallinari (2001)
16. Encarnação and Stoev (1999), ORIMUHS
17. Fischer and Ye (2001), Code-Broker
18. Goren-Bar, Kuflik, Lev and Shova (2001), SOM
19. Green and Carberry (1999), Initiative in Answer Generation
20. experiment 1 in Kim, Hall and Keane (2001), RLRSD
21. experiment 2 in Kim, Hall and Keane (2001), RLRSD
22. Lesh, Rich and Sidner (1999), Collagen
23. Litman and Pan (1999), TOOT
24. Luckin and du Boulay (1999), VIS
25. Magnini and Strapparava (2001), SiteIF
26. Marinilli, Micarelli and Sciarrone (1999), Information Filtering System
27. Mitrovic (2001), SQL-Tutor

continued on page 41

Figure 3.5.: Alphabetical index of studies that have been included in the synopsis and name of evaluated system (part I). The index is continued in Figure 3.6 on page 41

continued from page 40

28. Müller, Großmann-Hutter, Jameson, Rummer and Wittig (2001), READY
29. Noh and Gmytrasiewicz (1997), RMM
30. Paris, Wan, Wilkinson and Wu (2001), Tiddler
31. Semerano, Ferilli, Fanizzi and Abbattista (2001), CDL Learning Server
32. experiment 1 in Specht and Kobsa (1999), AST
33. experiment 2 in Specht and Kobsa (1999), AST
34. field study in Specht and Kobsa (1999), AST
35. Sison, Numao and Shimura (1998), MMD
36. Strachan, Anderson, Sneesby and Evans (1997), P-TIMS
37. Spooner and Edwards (1997), Babel
38. Theo (2001), ELFI
39. Trewin and Pain (1997), Key Board Skills
40. Villamañe, Gutiérrez, Arruabarrena, Pérez, López-Cudrado, Sanz, Sanz and Vadillo (2001), HEZINET
41. Virvou and du Boulay (1999), RESCUER
42. Vogt, Cottrell, Belew and Bartell (1999), User lenses
43. Weber and Specht (1997a), ELM-ART II

Figure 3.6.: Part II of the alphabetical index of studies that have been included in the synopsis (continued from page 40)

Table 3.4.: Classes of measures. The criteria found in the evaluation studies can be categorized in terms of the class of measure

	measure class		
	self-report	observation	archival records
accuracy, precision, and recall		x	
accuracy of system prediction		x	
amount of required help		x	
amount of requested material		x	
budget spent		x	
correct categorization of users		x	
computation time		x	
difficulty of learning	x		
domain knowledge; learning gain		x	
duration of interaction		x	
fixation times		x	
number of communications		x	
number of errors		x	
number of navigation steps		x	
overall impression	x		
rating of solution quality	x		
similarity of expert rating – system decision		x	
subjective rating of effect	x		
stability of user model		x	
system preference	x		
task success		x	
usability questionnaire	x		
user satisfaction	x		

observations

All other measures are directly observable , e.g., duration of interaction, or number of navigation or dialog steps. These measures are easy to assess and do not distort the results. However, some of them are difficult to interpret. For example, a reduction of interaction duration might be caused by either an interface that is more easy to handle or by annoyed users who tried to minimize the interaction as much as possible.

3.3. Problems in Evaluating Adaptive Systems

Based on the synopsis of evaluation studies, we tried to identify problems and difficulties in evaluating adaptive systems that might be responsible for the lack of significant studies. Several reasons have been proposed responsible for this shortcoming (e.g., Eklund, 1999).

One structural reason is that computer science has little tradition in empirical research and, thus, evaluations of adaptive systems are usually not required for publication. Empirical methods are not even part of most curricula. New publication requirements of journals and reviewers could raise the amount and quality of evaluation studies considerably.

Second, the development cycle of software products is short. Evaluations might become obsolete as soon as a new version has been developed. The resources consumed by the evaluation cannot be put to use for further development. However, evaluations should be seen as important feedback for the design process, that is applied throughout the whole life cycle. Proper chosen methods and criteria might assure that the results are meaningful for a longer period of time.

Third, adaptive systems have an inherent property which makes system comparisons difficult. In some cases, we cannot simply switch off the adaptivity and make a non-adaptive system of it, because adaptivity is an essential part of that system (Höök, 2000). We run into trouble if the adaptive system is not an extended version of a preexisting non-adaptive system (as in the following example), but designed from scratch. Defining an adequate control group might be difficult here, because switching off the adaptivity in these systems might result in a rather useless product. We will introduce an evaluation method that might relieve this problem by comparing different possible adaptation decisions.

Fourth, evaluation in this area only considered the system's precision without taking the behavior and cognitions of users into account. Only recently there have been some proposals on evaluation of adaptivity in general (Karagiannidis and Sampson, 2000).

Fifth, the expected effect sizes are pretty low for some systems compared to the huge that stems from the individual difference. For example, an adaptive learning

system that annotates links according to the user's current knowledge might improve the learning gain or the navigation, but we would certainly expect that the improvement of such a relatively simple mechanism is small in comparison to the obvious differences of learners in general. Adaptivity will usually smooth and improve the interaction, but will not lift it to a new level. From an empirical point of view, these effects are difficult to detect, because they have to be separated from the background noise.

Finally, what is successful adaptation at all? Is it possible to define adaptivity success without referring to a specific domain and system? We will propose a new criterion, called behavioral complexity, that might hold for many different domains, but in general, a comparison of different approaches will usually be limited to certain user properties in certain domains. For example, a possible evaluation domain might be: Using a stereotype approach is useful for the adaptation to customer preferences in the traveling domain, but adapting to prior knowledge in learning environments is more successful with Bayesian Networks.

3.4. Developing a Database of Empirical Evaluations

The synopsis of studies can be used as a fundament of a searchable online database, that provides an overview of the state of the art to the scientific community and encourages other scientists to evaluate their own system.

3.4.1. Aims

exploring the state of the art

First of all, the systematic synopsis was established to get an overview of the state of the art. We wanted to identify currently used methods and criteria, as well as omissions and problems in current evaluations. The results are reported in this thesis. However, the main reason, why we decided to put the synopsis online with an interactive database, was to encourage empirical evaluations of adaptive systems. By providing a searchable categorized set of studies, interested people get suggestions of experimental designs, criteria, and other experimental issues.

encouraging evaluations

Such a database will help to identify pitfalls in the planning process as well as in the analysis of collected data. Moreover, it will help to identify omissions in the state of the art in the future, e.g., a certain category of systems might appear to be not evaluated at all.

For people outside the community the database will serve as reference for the usefulness (or insufficiency) of adaptive systems in general, of certain types of systems, or of a specific system as it describes the current state of the art.

3.4.2. Online Interface

The database is called EASy–D (evaluation of adaptive systems database) and is available online at http://www.softwareevaluation.de. Users may search for either a system or a study. For finding a related system it is most important to search for a specific function. The user might just check the required function categories and retrieve the relevant systems. Searching for a specific name, and full text search are supported, too. The presentation of results includes the complete information that is available about each reported system, as well as a link to all studies that evaluated this system.

searching for systems

When searching for a related study the user might either fill in a method of evaluation, specify the evaluation layer, or limit the search to a certain data type or criterion (see Figure 3.7). In principle, other search criteria (e.g., sample size) would be easy to implement, but appear to be not very useful or even confusing.

searching for studies

In addition, there is a glossary that explains the categories and entries, and a form for authors to submit a new study. The submission procedure is explained in the next section.

3.4.3. Implementation and Maintenance

EASy–D is based on MySQL[2] and PHP[3]. Currently the database contains 43 studies most of them from the *UMUAI journal* and from *User Modeling Conferences* (Brusilovsky and de Bra, 1999; Jameson et al., 1997; Kay, 1999). Of course this small number of records would not require a complete database and should be seen as a starting point only. However, we hope that other authors are interested in making studies (either their own studies or papers that are of importance) available in EASy-D.

New records are submitted with an online form by categorizing and describing a study and—as long as it is not available in the database—the evaluated system. However, submissions are reviewed before being published to avoid abuse and to keep entries consistent in terms of language and format.

submission of studies

[2] open source database; see http://www.mysql.com

[3] scripting language for web development; see http://www.php.net

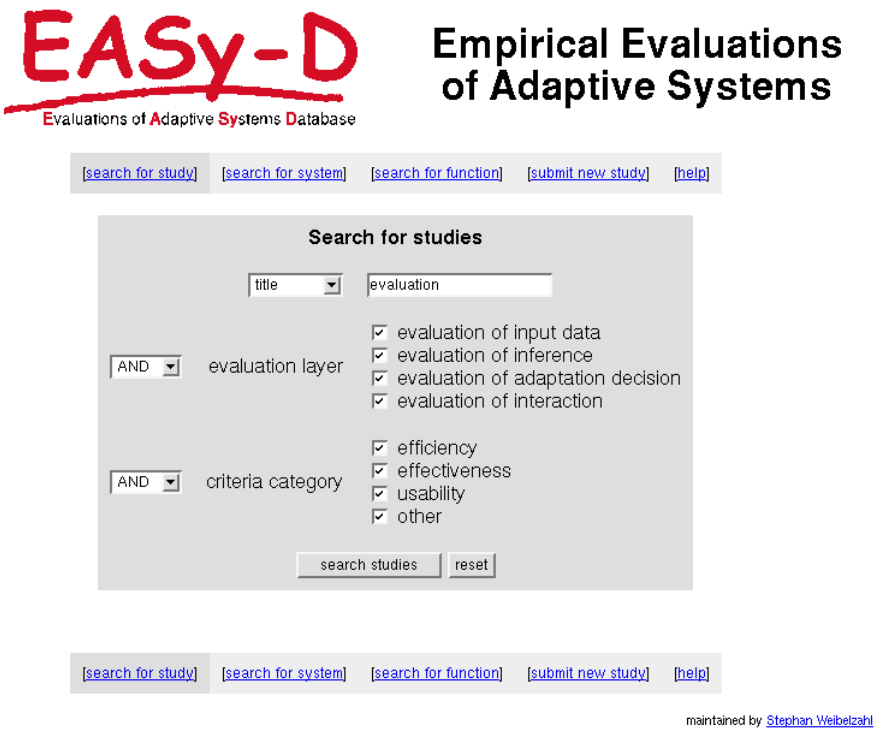


Figure 3.7.: One part of the online interface of EASy–D to search for studies that evaluate an adaptive system

Compared to a usual literature search users of EASy–D may search for system functions and evaluation methods very easily. Moreover, the studies are presented in a standardized way which gives a quick overview of the study. Another advantage is that related studies that evaluate the same system or a system with the same function are identified quickly.

We hope that this database will become the central contact point for researchers who are planning empirical evaluations of their adaptive systems and invite everybody to enhance EASy–D by submitting studies or giving feedback.

4. A Framework for the Evaluation of Adaptive Systems

Based on the systematic overview of current evaluations, we developed a framework that both categorizes existing studies and offers a systematic approach for evaluations.

4.1. Objectives and Scope of the Framework

In developing a framework for the evaluation of adaptive systems, we pursued two objectives: First, to specify what has to be evaluated to guarantee the success of adaptive systems, and second, to have a grid that facilitates the specification of criteria and methods that are useful for the evaluation.

evaluation targets

The first goal is very obvious. Currently, there is no guideline or comprehensive overview of what and how to evaluate an adaptive system. A framework like the one proposed here could help to systemize current approaches and offer hints how to identify failures and misconceptions in systems.

encouraging evaluations

The second goal is important to encourage further evaluations. Once suitable criteria and methods have been specified and collected it becomes much easier to establish an evaluation of a new system. Researchers can choose the required study design. This, in turn, could make evaluations more comparable which would even allow comparisons of different systems.

scope

As far as we can see, the framework is applicable to all adaptive systems with no limitation of the domain or inference mechanism. Probably, systems with explicit user models and simple inferences can be evaluated more easily, because failures can be backtracked in a straightforward way.

formative and summative evaluations

Both, formative and summative evaluations are supported (Scriven, 1967). While parts of the framework can be used even in very early stages of the software engi-

neering process even before anything else has been implemented, other parts look at the performance of the complete system. Thus, the framework can be integrated with existing software engineering models to identify very early failures and aberrations in the architecture and design.

Certainly, this framework focuses on empirical evaluations. Other software evaluation issues such as verification of algorithms are excluded. We do not doubt the advantages of formal methods for software testing. However, when evaluating the real world value of an adaptive system an empirical approach is inevitable.

In summary, we propose a systematic approach for the evaluation of adaptive systems that will encourage and categorize future evaluations.

4.2. Framework-Structure

rationale

The basic idea of the framework is to use the model of adaptive systems, introduced in chapter 1.2, to evaluate each information processing step on its own (Figure 4.1). In fact, we distinguish four evaluation steps (Weibelzahl, 2001; Weibelzahl and Lauer, 2001):

1. Evaluation of reliability and external validity of input data acquisition
2. Evaluation of the inference mechanism and accuracy of user properties
3. Evaluation of adaptation decisions
4. Evaluation of total interaction
   - 4.1 System behavior
   - 4.2 User behavior and usability

layered evaluation

The next sections describe these evaluation steps in more detail. All steps together can be seen as a so called "layered evaluation" (Karagiannidis and Sampson, 2000), i.e., a previous step is a prerequisite to the following steps. This procedure is outlined in chapter 4.3.

4.2.1. Evaluation of Input Data

To build a user model the system acquires direct or indirect input from the user (e.g., appearance of specific behavior, utterances, answers, etc.). These data are the basis of all further inferences. Thus, its reliability and validity are of high importance.

reliability of input data

While in some cases the reliability of the input data is unquestionable, there may arise serious problems in other systems. This is illustrated with two examples: An

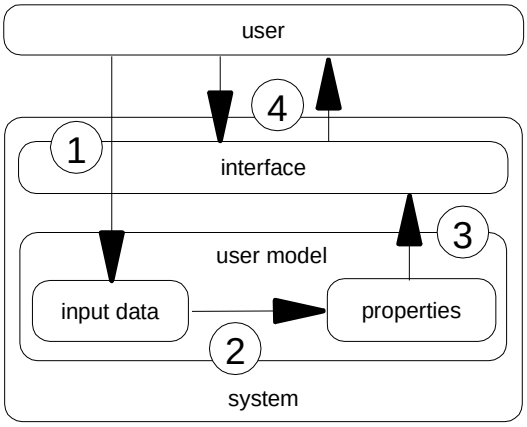


Figure 4.1.: Framework for the evaluation of adaptive systems. Each information processing step has to be evaluated on its own in a so called layered evaluation

adaptive user support system (Encarnação and Stoev, 1999) might exploit action sequences. Registering the number of sessions that the user completed or the number of *undo* actions is probably highly reliable. There is no subjective judgment or other noise involved in this observation. As opposed to that, an adaptive news broker (Billsus and Pazzani, 1999) has to care about the reliability much more. For example, users might provide feedback about a specific news story by selecting one of four categories: *interesting*, *not interesting*, *I already know this*, and *tell me more about this*. But the answer depends on many uncontrollable factors. The user might have read the story only roughly and might have overlooked some new facts. Or she might read the same story somewhere else afterwards. Or, just for the moment, she might not be interested in this kind of stories. Several other threats to reliability do arise here, and further inferences might be highly biased if the data quality is neither assured nor considered in the inference process.

validity of input data

Similar problems occur in terms of external validity. For instance, in adaptive learning systems, visited pages are usually treated as read and sometimes even as known (Brusilovsky et al., 1998; de Bra and Calvi, 1998; Weber, Kuhl and Weibelzahl, 2001). However, users might have scanned the page only shortly for a specific information, without paying attention to other parts of the page. Relying on such input data might also cause maladaptations. Again, it is important to check the validity in order to know which inferences are allowed or possible.

Anyway, in some systems the external validity is not of importance. For example, the number of mouse clicks is assessed objectively and no further interpretation is required to use this data as input for the inference of user properties.

For testing the data reliability the empirical test theory offers sufficient criteria and methods. The reliability of questionnaires, test items, and user feedback is assessed with retest- or split-half reliability measures. For instance, users of the news story broker could be asked about the same story after a while again. If the users' interest remains stable the data are assumed to be reliable. By observing the users' interaction with a learning system or by asking questions about the content of a read page it is possible to assure that read pages are actually known.

4.2.2. Evaluation of Inference

Based on the input, properties of the user are inferred. The inference itself is derived in many different ways ranging from simple rule based algorithms to Bayesian Networks or Cased-Based Reasoning systems. Similar to the first step we can evaluate the validity of the inference, too. In fact this means to check whether the inferred user properties do really exist.

validity of inference

An interesting method to do this, has been used in some of the evaluations listed in chapter 3.2 (e.g., Encarnação and Stoev, 1999; Green and Carberry, 1999; Sison et al., 1998; Virvou and du Boulay, 1999). Comparing the system's assumptions about the user with expert ratings or an external test may uncover false assumptions. For example, a system that adapts to key press errors (Trewin and Pain, 1997) can be evaluated in at least two ways. The system should either be able to detect different groups of users (e.g., subjects with disabilities vs. control group) or its assumptions about the users must be congruent with an expert's assessment. The assumptions of an adaptive learning system about the user's knowledge may be assessed in an external knowledge test.

The decision whether an a-priori or post-hoc approach is chosen depends on the availability of measurements and user groups; e.g., a knowledge test must be applied after the treatment, because the interaction influences the user's learning gain.

The quality of this comparison strategy certainly depends on the validity of the external test. There might be user properties that cannot be assessed directly or that cannot be assessed in another way than the one that is used by the system.

Note that this evaluation step checks the 'correctness' of the system's assumptions only. The usefulness of the properties is completely ignored here.

4.2.3. Evaluation of Adaptation Decision

During the so called downward inference, the system decides how to adapt the interface, e.g., how to change the layout, what additional information should be provided, which commands to offer, or how to tailor the presentation.

comparing adaptation decisions

Usually there are several possibilities of adaptation given the same user properties. Besides the way the system adapts usually it is often possible to ignore the user model completely or to use a single stereotype for all users. In addition, for most systems there are even more adaptation decisions. For instance, a product recommendation system might have inferred a strong preference for a specific product. It might now either recommend this product to the customer, only limit the possible selection to this product, indicate that there is a suggestion without naming it, or even recommend another product randomly. Comparing these alternative decisions might help to explore a kind of baseline that indicates what usual (non-intelligent) behavior could achieve and whether adaptation really has advantages. Most current evaluations compare the adaptive system with a non-adaptive version. However, this comparison might not be fair, because the non-adaptive version might be worse than what can be achieved with standard methods.

The aim of this evaluation step is to figure out whether the chosen adaptation decision is the optimal one, given that the user properties have been inferred correctly.

4.2.4. Evaluation of Total Interaction

The last step evaluates the total interaction and thus, it assesses the whole system in a summative evaluation. We can observe the system behavior and the user behavior, i.e., the usability and the performance.

System Behavior

Several dimensions of the system behavior may be evaluated. The most important is probably the frequency of adaptation. How often does adaptation occur during the interaction? For instance, an adaptive help system for UNIX (Virvou and du Boulay, 1999) might be perfect in identifying certain typing errors, but if these kinds of errors never occur in usual interactions the system cannot adapt.

frequency of adaptation

Moreover, the frequency of certain adaptation types is important, too. For example, a system might always chose the same adaptation decision. A product recommendation system might always come up with the same product, even if users differ, and different user properties are inferred, because not all combinations of user prop-

frequency of adaptation types

Table 4.1.: Hypothetical example of adaptation frequencies. Even if possible customer types would receive different product recommendations, empirical results might show that the same adaptation is chosen all the time

customer type	price	quality	recommended product	frequency
type 1	high	high	A	30%
type 2	high	low	B	0%
type 3	low	high	A	70%
type 4	low	low	C	0%
				100%

erties actually occur in real interactions. Table 4.1 illustrates this issue with a simple example. While theoretically three different products are recommended, empirical results might show that in fact only product *A* is chosen, because customers of type 2 and 4 never occur in real settings. Thus, the inference of user properties might be perfect, but because the adaptation is in fact static the user model is not required.

Other dimensions of system behavior include technical parameters such as required computation time, reaction time or delay of reaction and stability.

User Behavior and Usability

The user behavior can be evaluated separately, and is in fact the most important part. The adaptation is successful only if the users have reached their goal and if they are satisfied with the interaction. Thus, this final step has to assess both task success (respectively performance) and usability.

task success

Task success is domain specific and there are systems where the goal is not clearly identifiable. While for example learning systems obviously aim at improving the learning gain, an adaptive camera control system will always come up with a solution. Thus, in this case, task success is not a useful criterion but the subjective quality gains relevance.

performance

For some systems the performance of the users in terms of efficiency and effectiveness is crucial. For example, the success of an adaptive learning system depends on the users' learning gain (besides other criteria).

usability

Usability can be measured in different ways and may be broken down to very different domain specific measures (see Chapter 2.4). Most of the current evaluations

use duration of interaction and non-standardized questionnaires for the assessment of user satisfaction and subjective system preference.

As additional criterion we will introduce *behavioral complexity* which is a new measure that is especially tailored for adaptivity success (see Chapter 4.4.3).

behavioral complexity

4.3. Evaluation Procedure

The evaluation steps of the framework should be seen as a so called layered evaluation (Karagiannidis and Sampson, 2000). The evaluation of all previous steps is prerequisite to the current step. For example, we might find that a system infers incorrect user properties. This might have two reasons: either the inference itself has a failure, or the input data are unreliable. Thus, we first have to evaluate the input data, and will then be able to trace back the incorrect user properties to one of these reasons. The same holds for the evaluation steps 3 and 4. If there is no difference between an adaptive and a non-adaptive version of a system, this might be caused by wrong adaptation decisions, incorrect user properties, or unreliable input data.

layered evaluation

In some systems one of the steps might be redundant. For example, a system might acquire the number of mouse-clicks as input. These data are probably perfectly reliable and of high validity. In this case an empirical evaluation of this step is not required.

In summary, while step 1 through 3 are part of a formative evaluation that provides hints for shortcomings, step 4 represents a global summative evaluation of the complete system. All four steps are interdependent and are required for a full evaluation.

4.4. Methods and Criteria for the Evaluation Framework

The Chapter *Current Methods and Criteria* (3.2) has already listed many of the possibilities that may be applied within this framework. The following chapter categorizes these criteria according to the framework and introduces two new criteria, *structural characteristics* of the domain model of adaptive hypermedia, and so called *behavioral complexity*. Recently, the computation of several structural characteristics of the domain model in adaptive hypermedia has been proposed to evaluate the adaptivity degree of these systems. Thus, these measures could be useful for the evaluation of interaction (step 4 in the framework) in adaptive hypertext systems.

Behavioral complexity may be used both for the evaluation of the adaptation decision, when comparing different adaptations, and for the evaluation of the interaction. Both criteria are validated empirically and discussed in terms of their usefulness for the evaluation of adaptive systems.

4.4.1. Categorization of Current Methods and Criteria

The criteria, found in current evaluations (Chapter 3.2) may be assigned to the steps of the evaluation framework. An overview is given in Figure 4.2 and 4.3. Moreover, we added some criteria and methods that have been mentioned in the description of the evaluation framework. Certainly, some criteria apply for more than one evaluation step. In fact, the criteria for the evaluation of adaptation decision and for the evaluation of interaction are equal. These steps differ mainly in the methods and objectives. For example, while decision evaluation might compare the usability of four different adaptive versions of a systems, the interaction evaluation might estimate the system's usability in real world use for different user groups.

Of course, both tables overlap to a certain degree. Methods and criteria are not orthogonal, and some methods are closely coupled with a criterion. For instance, the comparison of user properties with an external assessment are measured in terms of congruency of both values.

4.4.2. Structural Characteristics of the Domain Model

Metrics for characteristics of the domain structure might be interesting criteria for the evaluation of adaptive hypermedia.

One of the central components of adaptive hypermedia systems is the underlying domain model. Different relations between the knowledge concepts determine the adaptation mechanism. Predicting the adaptivity degree of a system based on structural characteristics of the domain model, without empirical tests, would be very comfortable. At least two purposes could be accomplished in an easy way:

structural information for the evaluation of interaction

First, structural characteristics could serve as easy criterion for the evaluation of interaction (step 4 of the proposed framework). Both, the system behavior and usability could be estimated before any learner interacted with the system. The higher the adaptivity degree, the better the adaptation to different types of users, and "the larger will be the amount of users that can use the presentation in a personalized way" (Cini and de Lima, 2002, p. 498).

structural information for authoring support

Second, such a criterion could be interesting for authoring support. Existing adaptive learning environments (Brusilovsky et al., 1998; Carro, Pulido and Rodríguez, 2001; de Bra and Calvi, 1998; Murray, Shen, Piemonte, Condit and Thibedeau, 2000;

Criteria

Evaluation of input data: objectivity of data assessment; retest-reliability; split-half reliability

Evaluation of inference: accuracy of system predictions; congruency of assumed user properties and external test (e.g., χ^2-test); correct categorization of users; stability of user model

Evaluation of adaptation decision: accuracy, precision and recall; amount of required help; amount of requested material; behavioral complexity; budget spent; computation time; difficulty of learning; duration of interaction; fixation times; frequency of adaptation; number of communications; number of errors; number of navigation steps; overall impression; rating of solution quality; similarity of expert rating and system decision; subjective rating of effect; system preference; task success; usability questionnaire; user satisfaction

Evaluation of total interaction: accuracy, precision and recall; amount of required help; amount of requested material; behavioral complexity; budget spent; computation time; difficulty of learning; duration of interaction; fixation times; number of communications; frequency of adaptation; number of errors; number of navigation steps; overall impression; rating of solution quality; similarity of expert rating and system decision; subjective rating of effect; system preference; task success; usability questionnaire; user satisfaction

Figure 4.2.: Categorization of current criteria according to the evaluation steps in the framework

Methods

Evaluation of input data: *early exploratory studies*: objectivity, reliability and stability user assessment method;
controlled evaluation with users: occurrence of adaptation trigger under experimental conditions (e.g., high workload)

Evaluation of inference: *controlled evaluation with users*: comparison of assumed user properties with external test, expert rating, or self assessment; comparison of system predictions with actual user behavior;
controlled evaluation with hypothetical users: observation of user model in dependence of different hypothetical behaviors of users;
experience with real world use: observation of change in user model; comparison of assumed user properties with external test, expert rating, or self assessment; comparison of system predictions with actual user behavior; registration of frequency of adaptation

Evaluation of adaptation decision: *controlled evaluation with users* and *experience with real world use*: comparison of different adaptation decisions in terms of efficiency, effectiveness, and usability

Evaluation of total interaction: *controlled evaluation with users*: observation of system and user behavior under different conditions;
experience with real world use: observation of system and user behavior for different user groups in real world settings in terms of absolute efficiency, effectiveness, and usability.

Figure 4.3.: Categorization of current methods according to the evaluation steps in the framework

Sanrach and Grandbastien, 2000; Weber et al., 2001) offer different degrees of authoring support. However, the more widely these systems are used, the more obvious is the need for a good authoring tool, because adaptive hypermedia require activities that are beyond text editing, including several knowledge engineering activities.

relations between concepts

Most of the current adaptive hypermedia systems require the specification of at least two kinds of relations between concepts or pages: *is prerequisite of* and *infers*. Prerequisite concepts usually have to be learned before the related concept, i.e., understanding a concept requires to know a prerequisite concept beforehand. Concepts are inferred by other concepts, if knowing the second concept implies knowing the first. Note that this terminology is adopted from NetCoach courses. NetCoach is an authoring system that supports the generation of adaptive online courses (see Chapter 5.1). We prefer this terminology because most of the evaluation data presented below are based on courses that were created with NetCoach. Other authoring systems might use different terms for the same type of relations.

Based on this structural information, it is possible to provide adaptive features such as adaptive curriculum sequencing, adaptive annotation, and adaptive link hiding.

An authoring tool that is based on structural information of the domain model could indicate to the author whether and how to increase the adaptivity degree of their course or material. For example, a low adaptivity degree in terms of few concepts that have prerequisites might indicate that the author should add more prerequisites to improve the adaptivity.

Structural Information Measures

computation of adaptivity degree

Recently, Cini and de Lima (2002) proposed six measures that are concerned with the structure of the domain model. The exact definitions are listed below. In order to provide a better overview, we cite the definitions, name each of them and add textual formulas accordingly:

- The adaptivity degree of the user model in the generation of updates: percentile of pages that update other concepts in relation to the total pages of the presentation

$$A_{have\ inferences} = \frac{pages\ with\ inferences}{total\ pages} \tag{4.1}$$

- The adaptivity degree in the restrictions of the adaptation model: percentile of pages that have restrictions for their presentation in relation to the total pages of the presentation

$$A_{have\ prerequisites} = \frac{pages\ with\ prerequisites}{total\ pages} \tag{4.2}$$

- The user adaptable behavior degree in the presentation: percentile of concepts which can be altered directly by the user and are used as requirements to restrict other concepts in relation to the total number of concepts which can be altered directly by the user

$$A_{adaptable} = \frac{adaptable\ concepts\ that\ are\ prerequisites}{adaptable\ concepts} \tag{4.3}$$

- The content adaptation degree in the pages: percentile of pages that have conditional fragments in relation to the total number of pages

$$A_{have\ conditional\ fragments} = \frac{pages\ with\ conditional\ fragments}{total\ pages} \tag{4.4}$$

- The adaptive navigation degree in the pages: percentile of pages that have conditional links in relation to the total pages of the presentation

$$A_{have\ conditional\ links} = \frac{pages\ with\ conditional\ links}{total\ pages} \tag{4.5}$$

- The existence of an adaptive navigational map

The rational of the $A_{\text{have ...}}$ measures is, that only pages (or concepts) that *have* any relation to other concepts increase the adaptivity. All other pages are static. However, we argue that adaptivity degree could be interpreted the other way round as well: the more concepts that *are* prerequisite of another concept the more different adaptive suggestions may occur during interaction. Accordingly, the more concepts that are inferred by other concepts, the more pages might be skipped to reach a learning objective. Thus, we propose to consider the following $A_{\text{are ...}}$ measures as well:

- Percentile of concepts that are prerequisites of pages in relation to the total concepts. The more concepts, prerequisite of at least one concept, exist the more different guiding suggestions may occur.

$$A_{are\ prerequisites} = \frac{concepts\ that\ are\ prerequisite}{total\ concepts} \tag{4.6}$$

- Percentile of concepts that are inferred by pages in relation to the total concepts. The more concepts are inferred by a page the more changes may occur in the user model.

$$A_{are\ inferred} = \frac{concepts\ that\ are\ inferred\ by\ other\ concepts}{total\ concepts} \tag{4.7}$$

Both, $A_{\text{are prerequisites}}$ and $A_{\text{are inferred}}$ can be influenced by making implicit relations explicit without changing the structure. For instance, if A is prerequisite of B and B is prerequisite of C, then A is also prerequisite of C. Adding this last relation would increase $A_{\text{are prerequisites}}$, but the domain model would remain the same. Thus, for the computation of the above measures we also considered these indirect relations, because NetCoach uses them for the adaptation mechanism as well.

Moreover, we could also compare the number of relations that have been specified by the author. I.e., instead of counting the concepts that are or have prerequisites we could register how many prerequisites there are. The absolute number of relations should be standardized by the number of possible relations.

- Relative amount of prerequisites in relation to the maximum number of possible prerequisites

$$A_{prerequisite\ rate} = \frac{total\ prerequisites}{P_{max}} \tag{4.8}$$

- Relative amount of inferences in relation to the maximum number of possible inferences

$$A_{inference\ rate} = \frac{total\ inferences}{I_{max}} \tag{4.9}$$

  In NetCoach courses, for which we present some empirical data below, the maximum of prerequisites P_{max} and the maximum of inferences I_{max} that can be specified depends on the number of concepts n only, while cyclic prerequisites are disallowed.

$$P_{max} = I_{max} = \frac{n \times (n-1)}{2} \tag{4.10}$$

The prerequisite measures obviously require that the user is free to navigate trough the course. Otherwise a course with the maximal amount of prerequisites would be completely rigid, and not adaptive at all.

Empirical Validation of Structural Information Measures

The previous section lists many different measures, but which of them are useful? Should we urge authors of courses with low adaptivity degree to specify more concept relations to get better adaptivity?

empirical setting

We collected some empirical data from eight different NetCoach courses in different domains to answer this question. Most of these courses are part of the PSI project (Lippitsch, Weibelzahl and Weber, 2003) which develops adaptive online courses based on the authoring system NetCoach (Weber et al., 2001) to introduce students to pedagogical psychology. The course subjects include interpersonal communication (*Kommunikation*), student assessment (*Leistungsbeurteilung*), empirical methods (*Methoden*), social perception (*Personenwahrnehmung*), cognitive developmental psychology (*Piaget*), problem solving (*Problemloesen*), and psycholgical fields (*Psychologie*).

Students had to complete these courses as part of their curriculum. In addition, all courses, including the *HTML-Tutor*, which introduces publishing on the web, are available online for everybody. At the end of each course a questionnaire was presented and the learners had to rate the course in terms of several dimensions, including navigation, orientation, adaptation in general, annotation, and page suggestions on a 10-point scale:

subjective rating

- Navigation: *navigating through the course is ...* (difficult ... easy)
- Orientation: *during interaction I knew my current location (chapter, page) in the course.* (never ... always)
- Adaptation in general: *the course adapted to your learning progress. Do you think this was successful?* (not successful at all ... very successful)
- Annotation: *in the table of contents on the left hand side, chapters were annotated with different colors in accordance with your current knowledge level. The system intended to improve your orientation throughout the course by this.* (not successful at all ... very successful)
- Page suggestions: *the system tried to suggest pages to you that are adequate for your knowledge level. Has this been successful?* (not successful at all ... very successful)

In addition, the learners had to rate their impression of the interaction with the system in respect to four dimensions on a 10-point scale: terrible ... wonderful; difficult ... easy; monotonous ... stimulating; rigid ... flexible. We will call the mean value of these four scales *overall impression* of a course.

empirical results

The upper part of Table 4.2 shows the mean values of these ratings for all courses.

In addition, we computed six of the structural measures for each course separately. The other measures cannot be applied to NetCoach courses, because there are neither conditional fragments, nor conditional links. Moreover, all pages are adaptable in terms of their knowledge status directly by the user, and thus $A_{\text{adaptable}}$ is equal to $A_{\text{are inferred}}$. Both, conditional fragments and conditional links are specific for *AHA!* courses, which have been the main targets of Cini and de Lima (2002). *AHA!* (Adaptive Hypermedia Architecture) supports authors in implementing adaptive hypertexts, similar to NetCoach. Note that many of the structural measures that are concerned with inferences are 0. In six of the eight courses it was impossible to specify any inference. Thus, the following results for the inference measures are limited. However, other courses in different domains will have few inferences as well, because the condition of implying a complete concept is hard to fulfill.

Given these data, it is possible to correlate the structural course measures with the subjective ratings, in order to estimate the relation between these variables. The results are shown in Table 4.3.

Despite the very big sample size, all bivariate correlations are quite low ($|r| \leq .1$), i.e., only $r^2 < 1\%$ or less of the variance in one variable can be explained by the other. Taking an effect size of $r = .1$ for granted (which is very low) the correlations have a test power of $1 - \beta > .95$, i.e., even very small effects would probably have been detected. Nevertheless, four correlation are significant, all of which are very low. In summary, we found some statistically significant correlations, but the empirical effect size is probably not of importance for educational purposes.

Discussion of Structural Information Measures

There are at least three possible interpretations of these results. First, the fact that we failed to find considerable relations between the learners' subjective ratings and the structure of the courses might indicate, that all of the proposed measures are useless for authors. The specified content structure does not provide hints for further improvement of course adaptivity. At least it seems not to be related to the subjective impression of the users. Nevertheless, adaptivity degree might be useful for authors to get a kind of summary of their presentation.

However, and this is the second interpretation, the subjective ratings might have been useless to indicate what the structural measures should detect. The learners' answers in the questionnaire might have been influenced by the overall impression of the system regardless of the factual adaptivity success. As shown in Table 4.3, the overall impression correlates highly with the subjective adaptivity success measures, but not with the structural measures. However, a partial correlation with control for overall impression improves only slightly the relation between subjective ratings and

Table 4.2.: Mean values of subjective ratings and structural information of eight NetCoach courses on a 10-point scale (0-9). The sample sizes for the subjective ratings are shown in table 4.3

		Kommunikation	Leistungsbeurteilung	Methoden	Personenwahrnehmung	Piaget	Problemloesen	Psychologie	HTML-Tutor
subjective ratings	**navigation**	7.35	7.24	7.09	7.53	7.06	7.22	6.77	6.86
	orientation	7.1	7.06	7.37	7.78	7.37	7.82	6.98	7.12
	adaptation	6.26	5.64	6.14	6.3	5.72	6.3	6.02	5.95
	suggestions	6.51	5.94	6.32	6.53	5.94	6.55	6.39	5.91
	annotation	6.5	6.07	5.99	6.64	6.2	6.3	5.52	6.09
	overall impression	5.54	4.59	5.30	5.53	4.68	5.78	5.42	6.07
structure	$\mathbf{A}_{\text{are prerequisites}}$	0.63	0.96	0.97	0.92	0.72	0.76	0.96	0.95
	$\mathbf{A}_{\text{have prerequisites}}$	0.97	0.98	0.97	0.95	0.95	0.96	0.96	0.92
	$\mathbf{A}_{\text{prerequisite rate}}$	0.79	1	1	0.95	0.79	0.83	1	0.88
	$\mathbf{A}_{\text{are inferred}}$	0	0	0.15	0	0	0	0	0.37
	$\mathbf{A}_{\text{have inferences}}$	0	0	0.13	0	0	0	0	0.1
	$\mathbf{A}_{\text{inference rate}}$	0	0	0.02	0	0	0	0	0.04

Table 4.3.: Correlations of structural measures with subjective ratings of course users. The bivariate correlation and the sample size are reported. Statistically significant results are indicated with * ($p < .05$) and ** ($p < .01$). For all correlations the power is $1 - \beta > .95$, given $\alpha = .05$ and an effect size $r = .1$. In addition we report the correlation of overall impression with subjective ratings (last column) and with the strutural measures (last row)

	$\mathbf{A}_{\text{have prerequisites}}$	$\mathbf{A}_{\text{are prerequisites}}$	$\mathbf{A}_{\text{prerequisite rate}}$	$\mathbf{A}_{\text{have inferences}}$	$\mathbf{A}_{\text{are inferred}}$	$\mathbf{A}_{\text{inference rate}}$	**impression**
navigation	.032	-.056*	-.048	-.023	-.035	-.034	.386**
	1379	1379	1379	1379	1379	1379	1240
orientation	-.006	-.023	-.039	-.005	-.018	-.016	.333**
	1412	1412	1412	1412	1412	1412	1269
adaptation	.010	-.010	.002	.003	-.008	-.006	.471**
	1377	1377	1377	1377	1377	1377	1237
suggestions	.035	-.006	.019	-.027	-.047	-.045	.455**
	1345	1345	1345	1345	1345	1345	1205
annotations	.049	-.046	-.031	-.079**	-.100**	-.099**	.394**
	1261	1261	1261	1261	1261	1261	1127
impression	-.097**	.036	.024	.099**	.052	.094**	—
	1384	1384	1384	1384	1384	1384	

course structure. The highest bivariate correlation ($A_{\text{have prerequisite}}$ correlated with annotation) is raised to .095. All partial correlations with other structural measures are negative. This is, in fact, an implicit problem of the evaluation of adaptivity. The perfect adaptation is not even noticed by the user and can thus not be reported.

Third, we have to consider the fact, that adaptation is never independent of the content. As opposed to the idea, that more concept relations and a higher adaptivity degree result in a better course, each content might have its own ideal structure. Adaptivity degree might be an inherent property of a content that cannot be influenced. While some contents have many internal dependencies, others might have only very few. Increasing the adaptivity degree by specifying additional relations will not improve the adaptivity any more, or might even yield mal-adaptations.

Thus, the proposed structural measures might be interesting to compare the degree of possible adaptivity across contents, but our data do not support the claim, that they are useful for authoring support and evaluation.

A better way of supporting authors in specifying relations might be to visualize the domain (e.g., as a network or a matrix) or to check the relations for consistency automatically (Wu, Houben and de Bra, 1999), in order to avoid loops and other failures that would disturb the adaptation process.

4.4.3. Behavioral Complexity

Several of the evaluations report only weak differences between adaptive and non-adaptive versions of systems in terms of measures like duration or learning gain (Specht, 1998; Weber and Specht, 1997a). Many criteria suffer from low reliability or at least have not shown to be reliable. Thus, we claim that only those criteria that are specific for adaptivity and that are validated are acceptable.

reduction of interaction complexity

The basic idea of our approach is that adaptivity aims at reducing the complexity of interaction. The users should achieve their goals more easily. Adaptivity shifts the division of labor between user and system (Jameson, 1999). The system may take over routine tasks, such as planning-, sorting-, or selecting tasks or it may reduce the complexity of the task itself. Here are some examples that illustrate what is meant by complexity reduction. An adaptive help system (Encarnação and Stoev, 1999) may infer the user's current goals and select help texts accordingly. Thus, an overview of all the topics is not forced upon the user. An adaptive product presentation system (Jörding, 1999) simplifies the search for information by offering the products in the most suitable way. An adaptive learning system (Specht, 1998) supports the user's navigation by link annotation and curriculum sequencing. Thus, learners do not have to pay attention to the navigation itself and may focus on the learning task.

Device Representation

To control a device the user needs a cognitive representation of the device. This is sometimes called a mental model (Norman, 1983). According to Schoppek (2002) two main types of knowledge are discussed in the literature for controlling a dynamic system: first, input-output knowledge (I-O-knowledge) represents specific input values together with the corresponding output values. Second, structural knowledge is defined as general knowledge about the variables of a system and their causal relations. The author proposes that both knowledge types are required, but the impact of each type depends on the system size, and in turn its complexity. While for very large systems with several thousands of states general knowledge about the structure can be helpful for controlling it, for rather small systems it appears to be sufficient to acquire I-O-knowledge. Experimental results show that uninstructed subjects did not even try to identify general rules, while I-O-knowledge is acquired spontaneously (Schoppek, 2002). In accordance with this approach, Broadbent, Fitzgerald and Broadbent (1986) describe the interaction with small systems as a kind of lookup table, where the users try to reach their goals by anticipating the device behavior based on the result of previous interaction steps.

I-O-knowledge

structural knowledge

Most of the adaptive systems described in this thesis probably belong to the small or mid-sized system, thus I-O-knowledge is probably very important here. Its role is even more emphasized if we consider, that many adaptive systems are single-use system, i.e., they are used only once to complete a certain task, e.g., learning a specific content or finding a product. Often the interaction is too short to extract explicit rules.

If we accept that I-O-knowledge is crucial (though of course not sufficient) for defining the user's device knowledge, a description of the interaction behavior becomes important. According to Kieras and Polson (1999) the concrete behavior of a user can be described as a state-transition-network. The system changes its current state when the user initiates an action. For example, mouse-clicks, commands, or the selection from a menu initiate such a transition and the system enters a new state or returns to a previous visited state. Text-editing skills can be modeled with such a network (Bovair, Kieras and Polson, 1990).

interaction as state-transition networks

The analysis of protocol data yields an individual transition network for every user. Users that are familiar with the system are able to find the shortest path through the network to reach the final state (Borgman, 1999). Other users that have incomplete or even incorrect knowledge have to enrich the entire concrete task solving process with a lot of heuristics or trial and error strategies (Rauterberg and Fjeld, 1998). They will return to a previous state if they realize that the chosen transition did not result in the effect they wanted.

protocol analysis

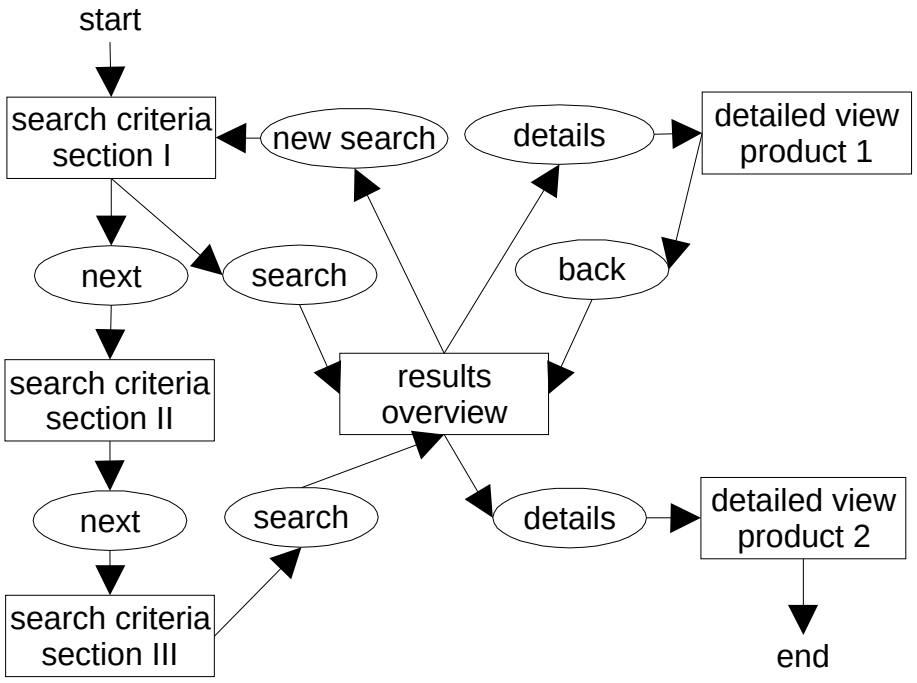


Figure 4.4.: Example of an state-transition-network for the interaction of a user with the product recommendation system CASTLE. States are represented by rectangles, transitions by ellipses

By exploring the system the users can increase their knowledge about the system's functions, but this exploration also results in a more complex network with an increased number of states and transitions, an increased number of cycles within the network, and a higher network density.

Figure 4.4 shows an example of a state-transition-network for the interaction of a user with an adaptive online product recommendation system called CASTLE (Weibelzahl, 1999). Basically, each state represents a page with different search forms or presentation modes. The user navigates between the states with different links, e.g., a search button. Note that this is not an one-to-one representation of the interaction. State and transitions that have been used more than once result in a single node only. Thus, the same network is derived regardless of whether the user used the same path once or several times.

Other methods have been proposed to model the knowledge that is required for controlling a device, and the user's goals and tasks, referred to as cognitive task

analysis (de Haan, van der Veer and van Vliet, 1991; Dix et al., 1998; Grant and Mayes, 1991). Some of these methods, such as *Goals, Operators, Methods and Selection* , *GOMS* (Arend, 1991) or the *task action grammar*, *TAG* (Reisner, 1984), might help to identify the relevant states and transitions. However, they are limited to error-free behavior and might thus fail to represent the complete interaction (Rasmussen, 1997). State-transition-networks allow for representation of more than *one best way* of task completion, which means they are able to distinguish between task and activity (Juvina, Trausan-Matu, Iosif, van der Veer, Marhan and Chisalita, 2002).

cognitive task analysis

In summary, we propose to model the interaction between user and system as a state-transition-network, which allows inferences about the user's knowledge. We claim, that adaptivity aims at reducing the interaction complexity which can be measured by means of these networks.

Measurement of Behavioral Complexity

If adaptivity reduces the interaction complexity, adaptivity effects can be measured by assessing the complexity of the state-transition-networks.

Rauterberg (1992) compares four different complexity measures, that are derived from graph theory (Curtis, Sheppard, Milliman, Borst and Love, 1979; McCabe, 1976). The most simple measure is C_{state}. Here, complexity equals the number of *s*tates found in a network.

complexity measures

$$C_{state} = S \tag{4.11}$$

But obviously, complexity of devices must consider the relations between the states, too. Otherwise, a system that bundles all functions on a single page would be categorized as less complex, while intuitively at least clusters of functions that belong together should be separated to improve the usability. Thus, C_{fan} computes the relation of states and *t*ransitions, which represents a kind of structural complexity.

$$C_{fan} = \frac{T}{S} \tag{4.12}$$

The third measure extracts the number of cycles in the network. Thus, it indicates how often a user returned to a previous state.

$$C_{cycle} = T - S + P \tag{4.13}$$

For the examples cited throughout this thesis the constant P always equals 1, and is used for correction purposes only.

The fourth complexity measure ($C_{density}$) shows the network's density in relation to the maximal possible density.

$$C_{density} = \frac{T}{S \times (S-1)} \tag{4.14}$$

The following equations compute these four complexity measures for the example network shown in Figure 4.4. The number of states equaling $S = 6$, and $T = 8$ transitions have been used.

$$C_{state} = 6 \tag{4.15}$$

$$C_{fan} = \frac{8}{6} = 1,33 \tag{4.16}$$

$$C_{cycle} = 8-6+1 = 3 \tag{4.17}$$

$$C_{density} = \frac{8}{6 \times (6-1)} = 0,27 \tag{4.18}$$

software-evaluation with complexity

All four measures have been applied to software-evaluation. According to Rauterberg (1992) all of them discern between novices and experts. The latter showed less complex behavior when completing a task with a database system. However, opposed to what was expected, C_{state} and $C_{density}$ varied with different tasks, i.e., they are only useful for experimental settings with constant tasks. Comparing different tasks with these measures is impossible. For the evaluation of adaptive systems this is not a serious limitation, because adaptivity aims at simplifying a constant task.

Empirical Validation of Behavioral Complexity

In a laboratory experiment Weibelzahl and Weber (2000) compared four measures of complexity for the interaction with an adaptive product recommendation system. CASTLE recommends vacation homes in France (Weibelzahl, 1999; Weibelzahl and Weber, 1999). It adapts to the user's needs and preferences by referring to the experience with similar customers. It will suggest vacation homes that are similar to those that have been booked by customers who have indicated similar preferences.

method

Participants were ask to find a suitable vacation home under one of two conditions. While *group 1* was supported by the user modeling component, *group 2* received no individual recommendations but was allowed to use the same functions as the first group when searching the electronic catalog. Afterwards, both groups had to fill in selected items of the *Questionnaire of User Interaction Satisfaction, QUIS* (Chin, Diehl and Norman, 1988).

Table 4.4.: Comparison of four complexity measures for two groups of users. Participants in *group 1* were supported by an adaptive version of CASTLE, while *group 2* received no individual recommendations. Sample size (N), mean ($\bar{x}$), standard deviation (σ), statistical significance of the mean difference (α), and effect size (d) are reported

	group	N	$\bar{x}$	σ	α	d
C_{state}	1	25	18.36	3.28	.029*	0.71
	2	17	20.70	3.31		
C_{fan}	1	25	1.47	0.17	.985	—
	2	17	1.47	0.21		
C_{cycle}	1	25	9.88	4.05	.003*	0.99
	2	17	13.88	4.04		
$C_{density}$	1	25	0.087	0.015	.029*	4.27
	2	17	0.100	0.023		

The experimental group required less time to find a suitable home and was more satisfied with the interaction. However, the results of this first evaluation study were not statistically significant.

results

The same data set was reanalyzed in terms of behavioral complexity. Table 4.4 reveals the differences between the groups for the four complexity measures.

treatment effects

Users who had been supported by the adaptive system produced behavior of reduced complexity compared to users that completed the same task ("Find a suitable vacation home in this electronic catalog") with a non-adaptive version. Certainly, the absolute value of the measures does not matter, only the differences between the groups is of importance here. In fact, C_{state}, C_{cycle}, and $C_{density}$ can discern the different treatments.

external validity

For further validation of the measures we explored the relation of behavioral complexity to the *total duration of interaction* and to the *subjective satisfaction with the interaction* based on the QUIS items. Higher values indicate higher interaction satisfaction. Furthermore, we expected that participants who are experienced in the use of computers and the internet will show less complex behavior. Subjects assessed their experience on a five grade scale.

Table 4.5.: Correlation of the four behavioral complexity measures with the interaction satisfaction (QUIS), duration of interaction, and experience with computers and internet ($N = 42$). Statistically significant results ($p < .05$) are marked with *

	satisfaction	duration	computer experience	internet experience
C_{state}	.15	$.50^*$	.17	.23
C_{fan}	$-.04$	$.53^*$	$-.25$	$-.22$
C_{cycle}	$-.04$	$.57^*$	$-.17$	$-.15$
$C_{density}$	$-.33^*$	$-.07$	$-.29$	$-.47^*$

As shown in Table 4.5, $C_{density}$ correlates highly with interaction satisfaction, while the other measures are related to duration only. All measures with the exception of C_{state} relate to computer and internet experience in the expected way.

Discussion of Behavioral Complexity

While traditional criteria, such as *duration of interaction* and *interaction satisfaction*, indicated only a vague difference between the adaptive and non-adaptive versions, three of the complexity measures were able to discern the groups. C_{fan} did not show the expected effect and is thus probably not very useful for the evaluation of adaptive systems. On the other hand, C_{cycle} and $C_{density}$ appear to be interesting measures, as they correlate with experience in the expected way. Especially $C_{density}$ is encouraging for evaluation purposes, because it is strongly related to subjective satisfaction but circumvents the problems of asking the user directly.

comparing systems

A major advantage of behavioral complexity is that very different systems can be compared. For the evaluation of adaptive systems, not only adaptive and non-adaptive versions might be used, but also different adaptation decisions (see section 4.2.3) might be compared. The problems of defining adequate control groups (Höök, 2000) are in this way alleviated.

preconditions

The correlation of $C_{density}$ with *interaction satisfaction* and *internet experience* suggests that behavioral complexity is based on differences in the users' device representations. However, at least three preconditions have to be fulfilled for such an interpretation. First, the states and transitions have to be represented on the same level of granularity. A system might have many states that can be discerned theoretically from a technical point of view, but behavioral complexity refers to the states

and transitions that are perceived by the user. For hypertext systems the definition of states and transitions is not too complicated, because web pages are usually static and clearly separated. In stand-alone applications, with a continuous interaction, states and transitions have to be defined carefully. Cognitive task analysis suggests adequate granularity.

Second, all users must solve the same task, i.e., aim for the same goal state. Thinking aloud or explicit instruction assures this fact. Otherwise the complexity measures become useless for comparisons.

Third, the complexity measures imply that all transitions have the same costs, i.e., all transitions can be learned equally. In CASTLE the transitions are very similar. But an adaptive system could in principle replace three easy to learn steps by one very difficult to learn command, e.g., offering a command-line interface instead of several selection forms. Obviously, the aggregated version would result in lower behavioral complexity, but from a cognitive point of view the selection task is preferable. Thus, we either have to make sure that the transitions have the same costs, or the complexity formulas have to be extended by weights for different transitions.

related work

Several other metrics have been proposed to assess the complexity or the structure of navigation behavior in hypertext (see e.g., Pitkow and Pirolli, 1999, and Herder, 2002, for an overview). Most of these measures are designed to assess the structure of a complete site. However, if we see the user behavior as an overlay of the site-structure, the same metrics are applicable for individual navigation as well.

Similar to the graph density ($C_{density}$), Botafogo, Rivlin and Shneiderman (1992) introduced a *compactness* measure which is based on the distance between all pages. The *stratum* metric indicates the linearity of the graph which is similar to the number of cycles (C_{cycle}). Moreover, the depth of a node (Botafogo et al., 1992), i.e., the distance from the root node, and the size of the cycles in the graph (Buckley and Harary, 1990) might be important characteristics of the behavior, especially because shorter navigation paths can improve the user satisfaction (Smyth and Cotter, 2002). The usefulness of these measures for the evaluation of adaptive systems needs to still be explored.

4.5. The Framework as Categorization Grid

The evaluation framework can be used as both, a structured approach for evaluation studies, and a categorization grid for existing studies. We categorized the current evaluation studies in accordance with this framework to see which evaluation steps have been applied regularly, and where we can identify omissions.

Table 4.6.: Categorization of evaluation studies according to the evaluation framework. The numbers in brackets (e.g., [1]) refer to the alphabetical index in Figure 3.5 on page 40

	evaluation of input data	evaluation of inference mechanism	evaluation of adaptation decision	evaluation of total interaction	Σ
tailor information presentation	–	[1], [4], [28]	–	[1], [2], [30], [42]	7
recommend prod. or other obj.	–	[6], [17], [25], [38]	–	[5], [7], [17]	7
help user to find information	–	[20], [21], [26]	–	[23], [26]	5
support learning	–	[8], [10]	[32], [33], [34], [43]	[3], [11], [12], [13], [24], [27], [40], [43]	14
give help	–	[16], [41]	–	[16], [36]	4
adapt interface	[37]	[31], [39]	–	–	3
take over routine tasks	–	[18], [35]	–	–	2
support collaboration	–	[19], [22]	–	–	2
other function	–	[9]	–	[29]	2
no concrete system	[14], [15]	[14], [15]	–	–	4
Σ	3	23	4	20	–

As shown in Table 4.6 almost all studies evaluate either the *inference mechanism* or the *total interaction*. Different *adaptation decisions* are examined only for adaptive learning systems. In other words, for all other systems it has not been considered that the same user properties could have been used in another way than the one already implemented.

Three studies are related to the evaluation of input data. These studies are concerned with visual search, access log analysis and keyboard skills. Only the last one is part of an existing system.

However, the synopsis did not contain studies about the evaluation of input data for any other task. For instance, for an adaptive product recommendation system utterances are not always that clear and customers might even try to cheat the system. Thus, some basic studies about the reliability of customer data is urgently required, but also for other user properties.

input data evaluations

This analysis suggests two important requirements for further evaluations. First, the input data of adaptive systems are evaluated in only a few studies. More investigations that demonstrate the quality of these data are necessary for all kinds of adaptive systems. The assessment of input data is the first information processing step and all further inferences rely on this data. This is actually a lack in current evaluations that should be considered in the future.

adaptation decision

Secondly, different adaptation decisions need to be evaluated on a larger scale, too. While there are some studies for adaptive learning systems, evaluations for other system functions are completely missing. In fact the chosen adaptation decision often seems obvious, but only by looking at alternatives can it be revealed whether this is the optimal decision. A recent study (Jameson and Schwarzkopf, 2002) demonstrates that users may differ markedly in their responses to different adaptation decisions.

4.6. Related Work

Recently, two other frameworks for the evaluation of adaptive systems have been proposed (Brusilovsky et al., 2001; Paramythis et al., 2001). The main idea behind all three frameworks is to break down the monolithic evaluation process into several components which can be evaluated separately. Early thoughts on how to define different measures for different components of the adaptation process have been formulated by Totterdell and Boyle (1990).

Brusilovsky, Karagiannidis and Sampson

The main difference between the three approaches is the number of components that are identified. Brusilovsky et al. (2001) demonstrated the benefits of the layered evaluation approach with two evaluation steps. The authors distinguish the *interact-*

ion assessment from the *adaptation decision*. The evaluation of the first layer covers both the low level monitoring (i.e., the observation of the user) and the high-level inferences. Thus, this approach is very similar to the one proposed in this work, though we claim that several distinctions that are found in our framework only (e.g., input data assessment vs. inference of user properties) are relevant and should not be skipped.

Paramythis, Totter and Stephanidis

Paramythis et al. (2001) proposed distinguishing five components that are very similar to the steps found in our framework: *interaction monitoring*, *interpretation / inferences*, *modeling*, *adaptation decision making*, *applying adaptation*. Thus, in comparison with our framework there are two differences. First, *inferences* and *modeling* are two distinct components here. However, the authors are not quite clear in respect to the fact that inference module represents a process while the modeling module is a component. Second, the last component (*applying adaptations*) has been subsumed under *adaptation decision* in our model as it is done usually in the literature as the authors admit. However, in some cases a distinction of these components might be feasible. In addition to these 'standard components' the authors specify three 'optional components'. Two of them can be seen as shortcuts in the standard cycle (*explicit provided knowledge* instead of *inferred knowledge* and *transparent models* instead of automatic *adaptation decision making*). The third additional component is used for systems with second level adaptation (Totterdell and Rautenbach, 1990). Thus, the underlying model of this approach is very similar to our framework though it provides some additional modules for special system architectures. However, the proposed evaluation procedure is a little bit different. Instead of the layered evaluation approach, the authors define eight 'modules' consisting of one or more components that could be evaluated in combination. Finally, even these modules may be evaluated in combination. In fact, the main difference to our framework is, that this approach is more explicit about involved components and procedures, but ends up with a confusing amount of different modules.

5. Empirical Evaluation of an Adaptive Web-Based Learning Course

In order to demonstrate the usefulness of the evaluation framework and to show which kind of failures are detected by which methods and criteria, the framework has been applied to an existing adaptive system. The *HTML-Tutor*, an adaptive learning course for HTML and publishing on the web, was object of several evaluation studies. First, we will introduce the underlying architecture of this adaptive system. Afterwards, we present the results of evaluation studies for each step of the framework.

5.1. NetCoach Courses

The *HTML-Tutor* is based on the authoring system NetCoach. A considerable number of courses has been developed by this tool. Though, the appearance of the NetCoach courses may look very different, the underlying structure and the inference mechanism is always the same.

NetCoach is an authoring system that supports the generation of adaptive online courses (Weber et al., 2001). It is derived from ELM-ART[1], one of the first and by now most comprehensive adaptive web-based educational systems (Weber and Specht, 1997a).

While authors generate the content by filling in templates and forms, the course functions including user management, adaptation, communication facilities, and tutoring is provided by NetCoach. All NetCoach courses share the same structure. Similar to chapters and subchapters in a book, the learning material (i.e., texts, images, animations) is stored in a hierarchical tree-structure of concepts. Learners may navigate through this structure freely.

[1] cogpsy.uni-trier.de/projects/ELM/elmart.html

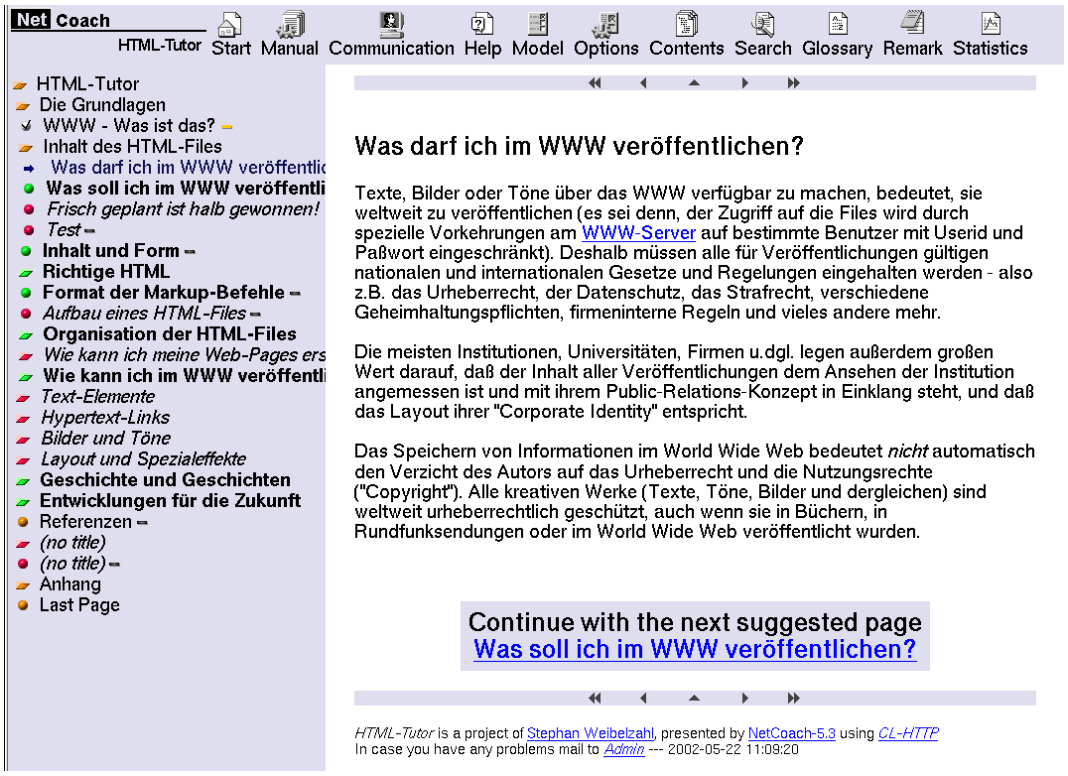


Figure 5.1.: Snapshot of the *HTML-Tutor*

learners' interface

Figure 5.1 shows a snapshot of the course interface for learners. The chapter overview on the left hand side unfolds and folds depending on the currently visited part. The links are annotated with colored bullets. The panel of buttons in the upper part of the window is visible at all times and provides functions such as searching, inspecting the glossary, and help texts. The main frame displays the learning material, tests, and suggestions on which page to go to next.

adaptive and adaptable features

In fact, the courses adapt to the learners' knowledge, to the learning objectives, and to the navigation behavior. This adaptation mechanism is detailed in the following sections. Moreover, the courses are adaptable to individual preferences such as colors, warnings, etc.. For example, a learner might adjust whether she gets a warning when she visits a page that requires additional prior knowledge. Table 5.1 gives an overview of the adaptive and adaptable features.

Table 5.1.: Adaptive and adaptable features of NetCoach

adapting to what?	how?
preferences	learner may change colors, the kind of warnings and annotations, etc.
learning objective	learner decides which predefined learning objective to fulfill (e.g., complete course, overview, specific contents only)
knowledge	learner has to answer test items (in pre-test or post-test) that check whether a concept is known or not
navigation	all visited pages are registered. As long as no other information is available (i.e., a test group), visited pages are assumed to be known by the learner.

5.1.1. Assessing the learner

The courses assess the learner's objective, they administer pre-tests and tests and register the navigation behavior (see Figure 5.2).

objectives

In the beginning of a course the learner is asked about her objective. A learning objective consists of a set of concepts that have to be successfully worked on by the learner. Thus, objectives are especially useful for learners who do not want to complete the whole course. For instance, the objective *I want to get an introduction on this topic* might include the introductory high level chapters only, while the second objective *I am familiar with ... but I want to know more about ...* would omit the first chapter and guide the learner to more advanced sections directly.

tests

The crucial parts of NetCoach to assess the learner's knowledge are the so called test groups. A test group consists of a set of weighted test items that are related to a concept. There are forced choice, multiple choice, and gap filling items. All of them are evaluated online automatically. The same items can be administered either as pre-test or as post-test. Pre-tests are presented optionally before the content of a chapter has been presented (*If you already have prior knowledge about this chapter you may want to complete the pre-test*). Post-tests are administered after the learner has read a chapter, usually in small groups of items (e.g., two in a row). These tests

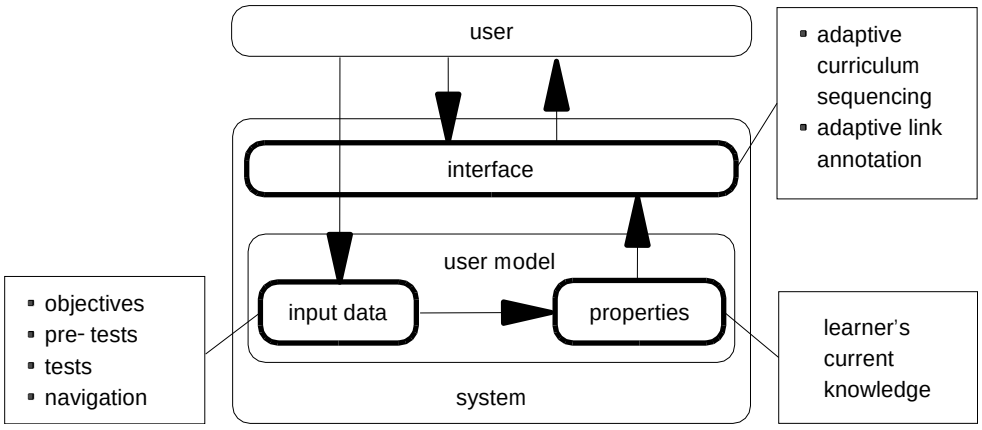


Figure 5.2.: Architecture of NetCoach courses illustrated with the evaluation framework. The courses acquire input data from specified learning objectives, pre-tests, and tests. Based on these data the learner's current knowledge is inferred. Finally chapter links are annotated and chapters are suggested in dependence of this knowledge

assure that the content of a chapter is actually known. For both, pre-tests and post-tests, the items are presented randomly (besides the fact that not yet presented items and incorrectly answered items are preferred).

5.1.2. Inference Mechanism

The inference of the learner's knowledge is based on the tests, described above, and on relations between concepts, that are stored in the knowledge base.

knowledge base

The knowledge base contains two kinds of relations between concepts. First, the author can decide which other concepts must be learned to understand the current concept. The system will guide learners to theses prerequisite pages before suggesting the current concept. Prerequisite concepts might themselves contain prerequisite concepts that are called indirect prerequisites.

prerequisite

inference

Second, a concept might infer another concept, i.e., the fact that the learner knows concept A implies that she also knows concept B. Note that prerequisites and inferences are related but are not equal. For example, knowing A might be required to understand B, but if one knows B this does not necessarily imply that A is known.

In addition to these relations between concepts the knowledge base contains relations between test items and concepts. Test items may not only test one concept but

also assess aspects of other concepts. Thus, it is possible to quantify the inference of test items to other concepts.

test groups

The results of the test groups are treated in the following way: users achieve points for correct answers and loose points for false answers until they reach a critical value. Hints for further reading and explanations support those users who have a lack of knowledge.

A concept is supposed to be learned if one has reached a critical value. If there are already some inferences from test items of other concepts, the learner is closer to this critical value and has to solve less test items in this test group.

user model

Based on the descriptions in the concepts, all pages are computed individually with respect to the learner's user model. The user model used in NetCoach is a multi-layered overlay model (Weber, 1999). Individual information about each learner is stored with respect to the concepts of the course's knowledge base (as described in the previous section). The first layer describes whether the user has already visited a page corresponding to a concept. The second layer contains information on which exercises or test items that are related to this particular concept the user has worked at and whether she has successfully worked on the test items up to a certain criterion. The third layer describes whether a concept could be inferred as known via inference links from more advanced concepts the user has already successfully worked on. Finally, the fourth layer describes whether a user has marked a concept as already known. That is, the user model can be inspected and edited (Bull and Pain, 1995). Sometimes, this is called a cooperative user model (Kay, 1995). Information in the different layers is updated independently. This leads to the fact that information from each different source does not overwrite others. For example, if a student unmarks a concept because she realized that she has not enough pre-knowledge about it, the information about tests on this concept is still available. See Figure 5.3 for an example of a student's overlay model. A concept is assumed to be learned if it is either tested to be known, inferred from other learned concepts, or marked by the user. In case no test group is available the concept is assumed to be learned if it has been visited. In other words, the visited layer and the test layer are applied alternatively. For instance, *Concept 6* of the students' model is assumed to be learned because the test group has been solved, though it also has been marked as *known*, i.e., the system trusts the first information more than the second one.

learning state

Finally, NetCoach summarizes the learner's current knowledge by assigning one of six states to each concept. Table 5.2 lists the states and describes the conditions of assignment. The current configuration of states is called a user's learning state. As it is computed on the fly for each user individually, the learning state models the idiosyncratic learning process during the interaction.

Table 5.2.: Possible states of a concept with a test group. The states are computed individually during interaction in dependence of the user's behavior

state	condition	annotation
not ready	there are prerequisites for a concept (e.g., concept A has to be learned before concept B) that are not fulfilled	red ball
suggested	all prerequisites are fulfilled	green ball
solved	the learner completed the test group of this concept successfully	grey ball with tick
inferred	the learner solved a more advanced concept first and thus the current concept is inferred to be already learned as well.	ball with tick
known	the learner marked the concept as known without solving the test group	crossed ball
other	the learner may access this concept, because all prerequisites are fulfilled, but it is not part of the current learning objective	orange ball

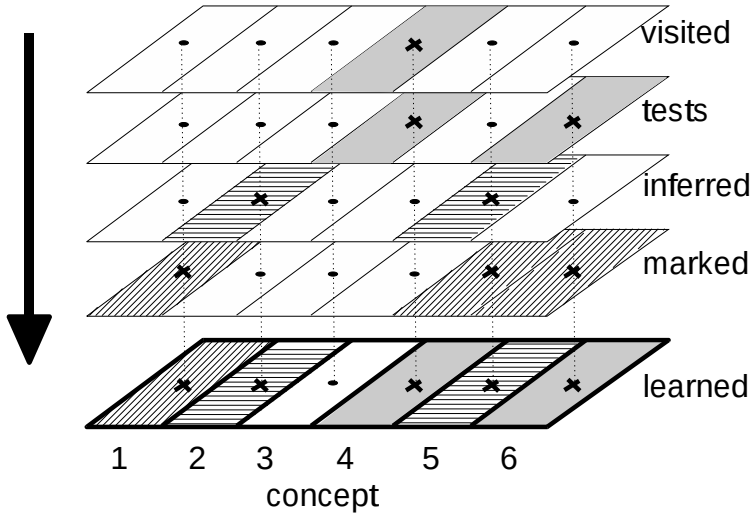


Figure 5.3.: Example of a student's overlay model. NetCoach infers the student's current learning state from four independently updated layers. Concepts without tests are treated as learned if they have been visited. Adopted from Weber et al. (2001)

In summary, the inference of a user's current learning state is done in three steps. First, the items of the pre-tests and tests are evaluated. Points are weighted by a factor and accordingly assigned to test groups . Second, if a test group has been solved because the critical value has been reached, that chapter is assumed to be learned and other chapters might be inferred to have been learned in accordance with the knowledge base. Third, each chapter is assigned to one of six learning states that describe the learner's knowledge of that concept.

5.1.3. Adaptation Decision

Finally, the learning state is used for adaptation. The two adaptation decisions used in NetCoach courses are *adaptive link annotation* and *adaptive curriculum sequencing*.

Links to other concepts in the overview (left hand side in Figure 5.1) are annotated with colored bullets that correspond to the learning state. Thus the learners can see, which concepts are assumed to be learned, which are recommended, and which are not yet ready. Table 5.2 gives an example of the default color configuration. However, NetCoach authors are free to predefine other colors.

link annotation

Table 5.3.: Adaptation decisions in adaptive learning systems. Comparison of authoring systems for adaptive hypermedia courses in reference to different adaptation decisions that are used for adaptation

	Net-Coach	AHA	ECSAI-Web	Inter-book	Meta-Links
adaptive guidance	yes	yes	yes	yes	yes
adaptive annotation	yes	yes	yes	yes	yes
adaptive hiding of links		yes			
adaptive navigation maps		yes			yes
adaptive text presentation		yes			yes

curriculum sequencing

In addition, NetCoach suggests the next concept to be learned. The system selects the next concept in the hierarchy with the state *suggested*, i.e, that has no unfulfilled prerequisites, and that is part of the learning objective. Concepts that are assumed to be learned are skipped. The learner can either choose a link to the recommended concept (*Continue with the next suggested page*) or follow the *suggested-arrow* in the navigation bar on each page. Learners who visit a page with the state *not ready* get a warning and are informed that the system assumes that they should study the prerequisites first.

warnings

Several systems have been proposed that apply intelligent inference mechanisms to adapt to learners, e.g., AHA! (de Bra and Calvi, 1998), Hyperbook (Henze and Nejdl, 1999; Henze et al., 1999), ECSAIWeb (Sanrach and Grandbastien, 2000), Interbook (Brusilovsky et al., 1998), and MetaLinks (Murray, 2000).

As shown in Table 5.3 NetCoach implements the most commonly used adaptive features, but does not adapt the text presentation (as e.g., AHA) and refrains from hiding links. We argue that the student should have full freedom of navigation and content access while the adaptive system should provide hints and suggestions only to scaffold the learner (e.g., Hübscher and Puntambekar, 2002).

5.1.4. Overview of existing courses

Several courses have been developed with NetCoach. They are used at different universities in Germany and in some companies. Up to now, most courses are written in German, though some are written in English, French, Spanish, and Italian. NetCoach does not require any programming knowledge, thus many different authors

from many disciplines developed courses in different domains including programming, spelling rules, cognitive and pedagogical psychology, medicine, and product presentation. At the University of Education in Freiburg, students develop simple courses on their own and test these courses with pupils in secondary schools. NetCoach has been used for 'learning on demand' settings, as well as for supplementing courses at universities and adult education. For instance, two courses on programming LISP and HTML are available world-wide for training purposes, while several courses on pedagogical psychology are used by students to prepare lessons and exams (Lippitsch et al., 2003). The students can thus work at their own pace until they acquired enough knowledge to attend the seminar session. The teachers save time because they do not have to present the complete content and can thus start at once to answer open questions and to discuss further issues. Nevertheless, they may control the performance of every student with the tutor interface. Other courses are used for further education in big companies, e.g., to inform the staff about the features of new products.

According to these different learning settings, the courses differ widely in terms of the offered functions. For instance, while document sharing is an interesting feature for closed groups, such as university seminars, this function is switched off for public courses like the *HTML-Tutor*. NetCoach is very flexible in terms of changing the interface in accordance with the requirements of learner groups and the courses may look quite different on the surface but in fact, the adaptation works in the same way for all of them.

The following empirical evaluations mostly focus on the *HTML-Tutor*, but many of the results might thus be generalized to the other NetCoach courses, too.

5.2. The HTML-Tutor

Before outlining the evaluation studies that we performed with the *HTML-Tutor*, the content and structure of the course is described in more detail.

5.2.1. Course Description

The *HTML-Tutor* teaches programming HTML and publishing in the Internet. It is based on a static online version written by Partl (1999). The 138 concepts (which equals about 120 printed pages) of the static version were extended by 125 test items. A table of contents is shown in Appendix A.1.

The topic has two properties that make it especially suitable for online courses. First, there are very clear relations between concepts. For example, to understand

concept relations

Table 5.4.: Comparison of the *HTML-Tutor* with other NetCoach courses. Note that for the computation of the number of prerequisites per concept pure test-concepts were excluded

	HTML-Tutor	**RR2000**	**Piaget**
concepts	138	141	42
learning objectives	12	11	1
test groups	49	110	13
test items	125	1399	63
prerequisites per concept	1.50	0.99	1.05

how to format a paragraph in HTML learners should know how HTML pages are structured in principle and what HTML tags look like. In turn, a learner who knows how to format a paragraph obviously has some knowledge about the syntax of tags. Accordingly the *HTML-Tutor* implements a prerequisite relation between the chapters 'structure of HTML files' and 'paragraphs and line breaks', as well as an inference relation between the latter chapter and 'format of markup tags'.

prerequisite structure

A total of 207 prerequisites and 52 inferences has been specified. The structure of prerequisites is outlined in Appendix A.2. For each chapter (shown in the table rows) a dot indicates for which other chapter (columns) this chapter is a prerequisite. Dots are spread over the upper right half of the table only, because NetCoach forbids to set a subsequent chapter as prerequisite. A diagonal means that the structure is very linear, because, in this case, the previous chapter is always prerequisite of the subsequent one. Structures that are very close to this diagonal can be found in courses like *Piaget*, an introduction to the developmental psychology of Jean Piaget, where the inherent dependencies are much weaker. The *HTML-Tutor* obviously deviates from this linear structure considerably. Accordingly, the number of prerequisites per concept is lower in other NetCoach courses. See Table 5.4 for a comparison of the *HTML-Tutor* to *Piaget* and *RR2000*, which is a course for learning the new German spelling rules.

inference structure

The structure of inferences is outlined in Appendix A.3. It looks quite different from the prerequisite structure because an inference relation is usually specified from summary chapters in the end of a larger content block which results in a kind of vertical lines. There are no inferences for the last chapters, because these sections provide a kind of supplemental information (history of HTML; future perspectives;

1. Which of the following statements about markup tags is correct?
   - ○ markup tags can be found both in HTML and SGML
   - ○ markup tags are always surrounded by '<' and '>'
   - ○ markup tags are always paired
   - ○ markup tags might have attributes
   - ○ markup tags have to be in lower case characters

2. Where in the HTML file is the definition of the so called <meta>-tags?
   - • at the beginning of the file before the <head>-tag
   - • between <head> and </head>
   - • between <body> and </body>
   - • anywhere

3. Small images that are used as preview of large images are called _____ .

Figure 5.4.: Example of three test item types: a multiple choice, a forced choice and a gap filling item from the *HTML-Tutor*. The correct solutions are: 1a, b, d; 2b; 'thumbnails'

references) that is not as strongly related to HTML programming as the previous content.

differences in prior knowledge

A second property makes HTML especially suitable for online courses: People who access such a course probably differ in prior knowledge. Opposed to very specialized topics (e.g., the developmental psychology of Piaget), HTML is often learned in packages where new demands of a task require the learner to acquire additional knowledge. Thus, the *HTML-Tutor* implements a total of 12 different learning objectives, including e.g., *I want to work on the complete course* and *I do know much about the WWW and want to learn more about the HTML tags now*.

functions

As supplemental functions the *HTML-Tutor* offers a chat and a discussion board.

Which of the following statements about markup tags is correct?

The answer was false.

A: your answer
C: correct solution

A	C	
	×	markup tags can be found both in HTML and SGML
×	×	markup tags are always surrounded by ‘<’ and ‘>’
		markup tags are always paired
	×	markup tags might have attributes
		markup tags have to be in lower case characters

Reason: Markup tags are part of the syntax of HTML as well as of other languages like SGML. All tags are surrounded by ‘<’ and ‘>’.
Usually the tags are paired, however there are several exceptions, e.g., <li> or
.
Some markup tags contain supplemental attributes. This is written as <tag attribute=value>.
Markup tags are usually written in lower case characters, but this is optional.

Figure 5.5.: Example of a feedback to a false answer with the correct solution and an extended explanation

However, both are used sparsely. Moreover, learners may search for a term in the material, browse the table of contents and inspect their user model (see Section 5.1.2). Finally there is a glossary that explains the most important terms.

The 125 test items are forced choice, multiple choice and gap filling. Figure 5.4 shows translations of examples for the three item types. In case of a false answer the answer is presented to the user again in conjunction with the correct solution and with an extended explanation (Figure 5.5). test items

The course has been online since September 2000 and has been accessed by several thousand users. For evaluation purposes we used both this online version with additional experimental conditions as well as an almost identical seminar version with different learning objectives.

If not specified otherwise, the following studies rely on the online users who have visited the course until the 23^{th} of November 2001. For some of the studies we excluded learners who had worked with the course for less than 15 minutes in order to filter persons who just wanted to have a look at the course, but did not really interact with it.

5.2.2. Overview of Evaluation Studies with the HTML-Tutor

The evaluation studies described in this work were designed to form a complete evaluation of the *HTML-Tutor* according to the proposed framework. Thus, each study is assigned to one of the four evaluation layers.

The *input data* were evaluated by computing the difficulty and retest-reliability of test items. We found that the item difficulties are equally spread across the full range. The retest-reliability of some items is low which might be caused by the measurement design.

The *inference* mechanism was evaluated in two studies. First, we compared the assumptions of the *HTML-Tutor* about the learners (i.e., the user model) with the results of an external assessment. The congruence of these measurements is considerably high, though a few incongruencies were uncovered. Second, we demonstrated that users who had worked on a page in one of five NetCoach courses, including the *HTML-Tutor*, made more mistakes when answering test items on this page if they were assumed to be *not prepared* for this page according to the user model.

Two aspects of the *adaptation decision* are evaluated. First, we were able to show that adapting to prior-knowledge by pre-tests saves time for the users, but keeps the learning gain stable. Second, we compared four different adaptation decisions that are all based on the same information in the user model.

Finally, the *interaction* in general is evaluated in terms of several dimensions of the system behavior, the user behavior, and the interaction quality.

Table 5.5.: Overview of evaluation studies with *HTML-Tutor* in respect to their design and criteria

evaluation layer	chapter	study design	criteria
evaluation of input data	5.3	observation of online-users of *HTML-Tutor*	difficulty and retest-reliability of test items
evaluation of inference	5.4.1	assessment of users in seminar	congruence of user model and external assessment
	5.4.2	observation of online-users of five NetCoach courses	comparison of test item answers and user model
evaluation of adaptation decision	5.5.1	observation of online-users of *HTML-Tutor* with and without pre-tests	duration of interaction and knowledge of users
	5.5.2	observation of online-users of *HTML-Tutor* and *RR2000* with different adaptation decisions	behavior, knowledge and feedback of users
evaluation of interaction	5.6	observation of online-users of *HTML-Tutor*	system behavior, user behavior, interaction quality

5.3. Evaluation of Input Data

The *HTML-Tutor* takes three types of input data into consideration: learning objectives, navigation behavior, and responses to test items (cf. Figure 5.2). Objectives and navigation can be regarded as reliable, at least there is no evidence that these two kinds of data are seriously threatened. Learning objectives are seldom changed. While most users selected only one objective (Table 5.7), 21% of the users changed the objective at least once. Most users decided to work on the complete course anyway (Table 5.6). On average each user selected 1.33 objectives. Looking at the interaction duration, this equals a changing about 0.63 times per hour of usage, i.e., the objectives of the users were very stable.

learning objectives

The observation of the navigation behavior is completely non-reactive, as the users' traces are recorded without any impact. Despite the fact that slips might occur due to the nature of the internet when learners use the back-button of their browser, the server has complete control over the material that is presented to the user. In other words, the *HTML-Tutor* can record exactly which material has been requested by which learner.

navigation behavior

However, the reliability of the third kind of input data, the responses to test items, requires a closer look, because the assessment of knowledge is the crucial part of the NetCoach course and the computation is not as straightforward as for the other two kinds of input data.

5.3.1. Method and Criteria

The most preferred way of testing the reliability of single test items is the retest reliability. However, a pilot study showed that participants of a seminar who were tested twice—once when registering for the seminar and the second time at the beginning of the seminar—did not differ in previous knowledge at all. None of them was able to answer any of the items, although simpler items had been selected. Thus, we decided to measure retest reliability by registering items that had been answered twice during interaction.

First of all, the difficulty of each test item was determined empirically by computing the relation of correct answers to the total number of answers. A total of 55189 responses ($\bar{x} = 445.07$ responses per item; $\sigma = 36.07$) were included. Figure 5.6 shows that the items are spread equally over the full range. These difficulty values were fed back to the course. All test groups, i.e., the collections of test items that are presented after a chapter (see Section 5.1.1), rely on these empirical difficulty values, as the achieved points per item (v_i) depend on the item's weight (w_i) and its difficulty (P_i).

item difficulty

Table 5.6.: Frequency of the 12 learning objectives in the *HTML-Tutor*. The table shows how many users selected the objective at least once. 1714 users selected 2197 objectives ($\bar{x} = 1.33$). Taking the duration of interaction into consideration, the users changed their objective only 0.63 times per hour on average

	objective	pages	frequency
#1	I want to work on the complete course	133	1168
#2	I want to print this course	1	316
#3	I want to look up a certain tag	1	87
#4	I want to know what WWW and HTML is	23	53
#5	I do know the WWW, but I want to learn everything about HTML	94	221
#6	I want to publish a piece of information quickly	29	27
#7	I want to publish voluminous information in a structured way	56	77
#8	I want to compile hot links to cool sites	19	7
#9	I want my pages to look colorful and modern	89	49
#10	I want to attract many readers	57	26
#11	I want to know more about the background of WWW and HTML	13	47
#12	I want to use this course for my teaching	1	119
			Σ 2197

Table 5.7.: Number of selected learning objectives. 1714 users selected 2197 objectives. The table shows how many users selected how many of the 12 learning objectives during the interaction with the course

objectives	frequency
1	1349
2	222
3	99
4	32
5	10
6	1
7	0
8	1
> 8	0
Σ	1714

NetCoach computes the user's score S in a test group t in the following way:

$$v_i = a_i \times w_i \times (P_i + 0.5) \quad \text{with} \quad \begin{array}{l} v_i \text{ score for test item i} \\ a_i \in \{0;1\}, \text{ 1 if answer was correct, else 0} \\ w_i \in [0.5;1.5] \text{ weight of test item i} \\ P_i \in [0;1] \text{ difficulty of test item i} \end{array}$$

$$S_t = \sum_{i=1}^{n_t} v_i \quad \text{with} \quad \begin{array}{l} S_t \text{ score for test group t} \\ n_t \text{ number of answered test items in test group t} \end{array}$$

The score in a test group equals the sum of scores that has been achieved by answering test items in this test group so far. Note that the adjustment of the difficulty values by 0.5 is used to simplify authoring, as the default values ($w_i = 1$ and $P_i = .5$) result in a score of $1 \times 1 \times 1 = 1$ per item, which is easier to handle. Computationally it has no impact. Negative scores for false test items can be achieved by setting a_i to negative values in the case that the answer is not correct, but the *HTML-Tutor* registers positive evidence only (i.e., $a_i = 0$ for false answers).

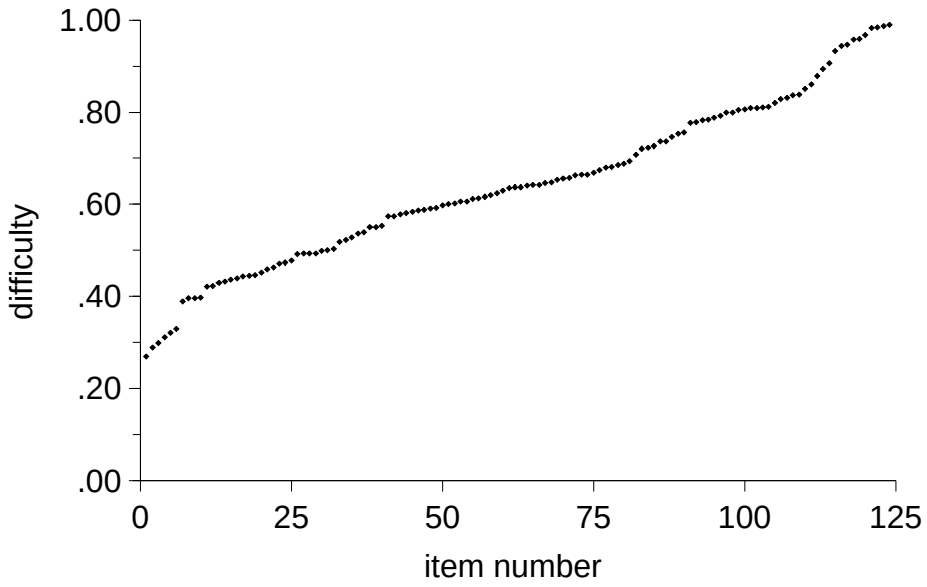


Figure 5.6.: Empirical difficulty of the 124 test items. For a better overview, the items have been sorted according to the difficulty value

5.3.2. Results

Retest reliability of test items was computed based on 7535×2 item responses of 1286 users who had answered at least one item more than once. The last two answers were correlated. Figure 5.7 shows the retest reliability of 98 out of 124 test items. The remaining 26 items had to be excluded due to too small sample sizes ($N < 10$) or due to a lack of variance ($\sigma_i^2 = 0$).

The reliability for most test items is low, but positive and ranges between 0 and .8. Several items have been identified that had a negative correlation, i.e., the first and the second answers to this item frequently differed. The overall values are too low in terms of a regular diagnostic instrument. However, we underestimated significantly the item reliability, because our experimental design does not control for learning effects. While knowledge might be stable between answering the item in a test group during the interaction and in a final test at the end of the course, there might be learning effects if the item was answered twice in the same test group. Then the answer pattern will usually equal 'false,correct', because the item was presented again after a false response.

The test items with the negative reliability belong to different test groups. Obviously, the users have more difficulties to solve these items in the second attempt.

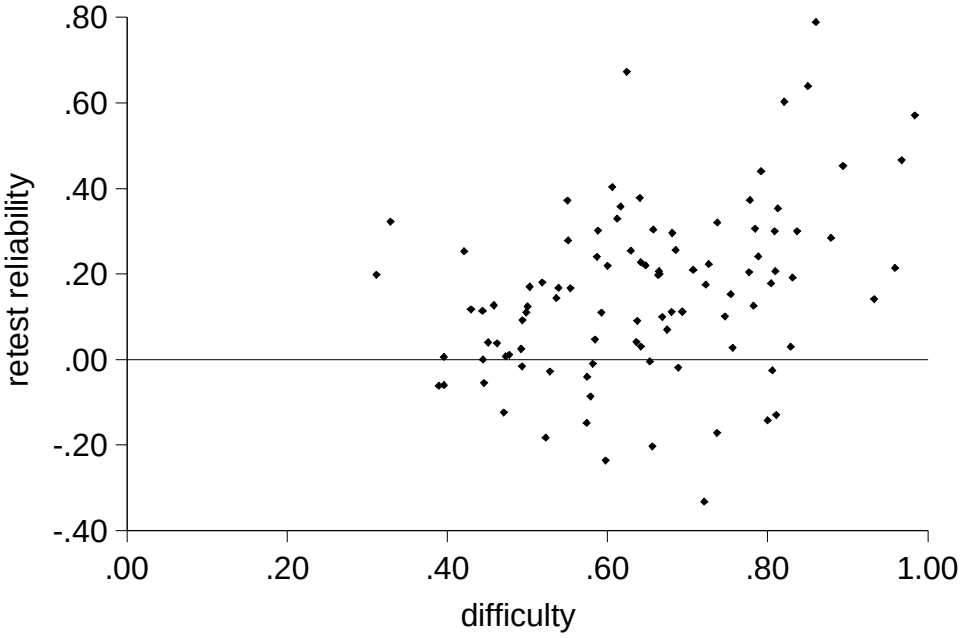


Figure 5.7.: Retest reliability of 98 test items. 1286 users answered 7535 $\times$ 2 items ($\bar{x} = 76.89$ users per item, $\sigma = 69.13$). 26 items were excluded due to too small sample sizes or lack of variance

Which of the following tags does not make the image visible on the current page, except when it is requested with a 'click'?

- <img src="bild.gif" alt="Bild">
- Bild
- <img href="bild.gif" alt="Bild">
- Bild

Figure 5.8.: Example of a multiple choice test item from the *HTML-Tutor*. In the test-retest design it correlated -.33 with itself. Correct solution: b

Figure 5.8 shows the item with the lowest correlation. It was impossible to identify any factors that were responsible for this fact. Neither item type (multiple choice, gap filling, forced choice), nor content seem to be related to the retest reliability.

5.3.3. Discussion of Input Data

In summary, the measured retest reliability of some test items is pretty low, however, we argue that this is an effect of the empirical design. The real reliability of the input data should be sufficiently high to allow the further inferences, especially if we consider the fact that usually at least three items are aggregated in a test group. The previously intended study design of measuring two separate points in time (e.g., registration and beginning of seminar) with stable knowledge of the users would have been better from a statistical point of view, but it is more difficult to find an appropriate sample for this design.

5.4. Evaluation of Inference

Are the assumptions about the learner, called the current learning state, inferred correctly from the input data? We conducted two studies. The first study investigates the learning state itself. We compared the assumptions of the *HTML-Tutor* about the learners' knowledge with an external assessment. The second study compares the assumption of the *HTML-Tutor* with the actual behavior of the learners.

5.4.1. Assessing the Learner's Current Knowledge

congruence of user model and external assessment

If the current learning state is a valid model of the learner's knowledge it should be in congruence with an external independent assessment. An external test can validate the user properties that are stored in the user model. A perfect user model is fully congruent with reality and thus, it should also agree with a valid test.

Method and Criteria

We assessed 32 students who took part in one of three compact-seminars between April 2001 and April 2002. The 10 male and 22 female students are studying at the University of Education in Freiburg for 0 to 9 semesters ($\bar{x} = 3.55$). Their studies included different academic areas, most frequently teaching. The announced requirement for taking part in this seminar was general familiarity with the usage of computers and to be ability to browse the Internet. The last requirement was checked by setting an obligatory online registration for the course. In fact, all participants had sufficient computer skills to follow the instructions during the sessions.

external knowledge assessment

The seminar consisted of 20 lessons on HTML and publishing in the Internet. During the seminar the students had to learn with the *HTML-Tutor* at least twice. After the seminar the students had to take part in a test. This test was designed to assess their performance as exactly as possible and consisted of three parts: first, the students had to generate an HTML page that fulfills certain conditions. Using their computer they had to produce source codes that yield a given layout and functions, e.g., clicking on the image should link to the homepage. Second, a paper and pencil test included three questions on more comprehensive knowledge, e.g., they had to explain why HTML is not very suitable to produce a specific layout. Third, they had to identify and correct errors in a given source code. A copy of the complete test is shown in Appendix B.

The test was evaluated individually in regards to the concepts of the *HTML-Tutor*. Given a learner's test performance we decided which concepts are already known or unknown. The test collects different data types (source code generation, open questions, source code correction) and it can thus be assumed to be a good estimator of the 'real' domain knowledge. However, it is obviously not a perfect test which might bias the results. In fact, the proposed congruency approach can be seen as a kind of parallel test reliability. If the external test was not reliable the expected congruency would be reduced. We tried to improve the test's external validity by including different task types and by considering as much information about the learner's performance as available. That is, in a qualitative analysis for each concept we decided whether the learner has complete knowledge about it.

Table 5.8.: Frequencies of congruence and incongruence of assumptions about the learner's knowledge in the *HTML-Tutor* and an external test. The *HTML-Tutor* assumes a concept either to be *solved* or *inferred*, otherwise there is *no information* whether the concept is known or not

		user model			
		solved	**inferred**	**no information**	Σ
external test	**known**	129	2	129	260
	unknown	9	0	23	32
	no information	601	261	382	1244
	Σ	739	263	534	1536

system's assumptions

The results of this analysis were contrasted with the system's assumptions about the learners in the user models. These assumptions relied on answers to test items during the seminar when the students were interacting with the *HTML-Tutor* and on 40 randomly selected test items that had to be completed after the external test. The *HTML-Tutor* overlay model is designed to consider positive evidence about the learners' knowledge only. In other words, the user model represents the learners' knowledge but not their misconceptions. Thus, there is no direct counterpart in the user model for the *unknown* category in the external assessment.

Results

congruence

We found that most assumptions were in congruence with test performance (see Table 5.8). 131 concepts were assumed to be either solved (i.e., the learner completed the test group successfully) or inferred (i.e., a higher concept had been solved before), while the external test also indicated that these concepts were known.

The high number of concepts that were not covered by the external assessment results from the fact that the time for such a test is limited. Compared to the 20 hours of teaching one hour for testing can obviously assess only a few selected aspects.

incongruence

However, we identified nine incongruencies, i.e., there were nine cases where the system's assumptions about the knowledge of a chapter differed from the result of the external knowledge assessment with the test. These incongruencies were caused by three concepts, namely the chapters 1.6, 2.3, and 2.5 (see Figure 5.9). For all three concepts we were able to show that the test group did not measure the same kind of knowledge as the external test did. For instance, in five cases the external test indicated that the learners do not encode German umlauts correctly. Nevertheless

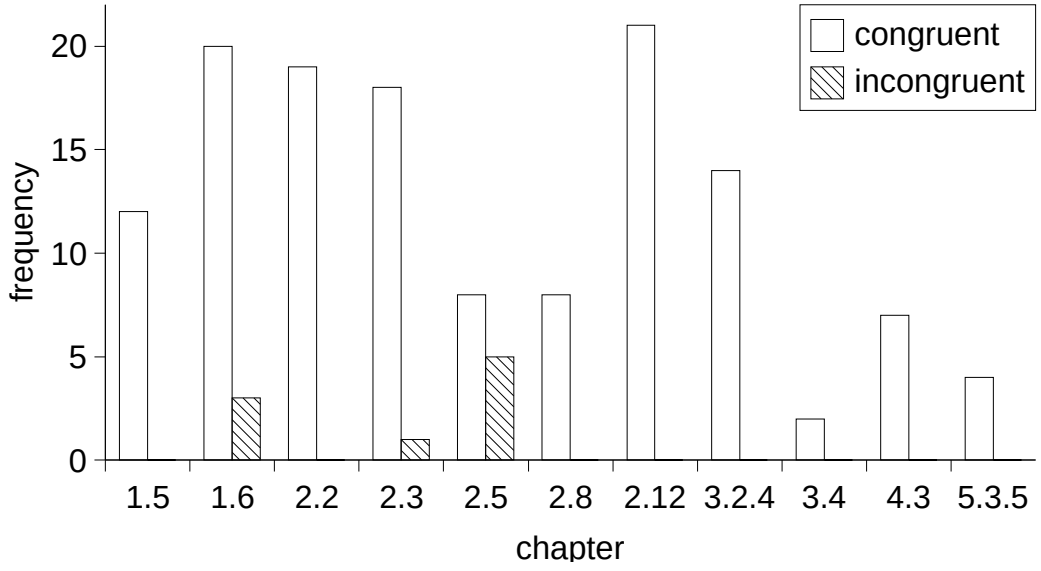


Figure 5.9.: Frequencies of congruence and incongruence of *HTML-Tutor* assumptions about the learner and an external test, grouped by concepts. The nine identified incongruencies (Table 5.8) were caused by three different concepts

they were able to respond to test items on this topic correctly. Obviously there is a mismatch between the declarative knowledge (as measured by the test items) and the displayed performance in real world settings. Similar results were found for the second concept (chapter 1.6): in test items the students were able to produce a correct structure of an HTML page, but when they had to work on their own, two students skipped the header of the page which resulted in an incorrect page. The third concept (chapter 2.3) introduces the line break tags (
). One student encoded them correctly when answering the test items, but sometimes forgot to do so when generating a page.

Evaluations of adaptive learning systems usually observed the impact of adaptivity on the user's behavior (e.g., Brusilovsky and Eklund, 1998, or Weber and Specht, 1997a) or the learning gain (Specht, 1998). Contrasting the user model and the external assessment provides important information about the system's quality. The study gave valuable hints for further improvements: it will be necessary to measure performance knowledge in test groups if we want to guarantee that the learners will apply their knowledge afterwards. Comparing assumed user properties with external assessment results provides important information about mal-adaptations.

contingency

To get a measure of how close the relation of user model and external assessment is, it is possible to compute the χ^2-contingency-coefficient based on the data given in table 5.8 (the exact formulas are listed in Appendix C). A contingency-value of $C_{corr} = .24$ suggests, that the two assessment methods are related, but do not measure the same. However, this coefficient must not be interpreted in the same way as usual correlations. We suggest to use C_{corr} for comparisons of different adaptive systems or of different versions of the same system, to estimate which system's user model is closer to the external assessment. However, from a formative evaluation perspective it is more important to know which concepts are incongruent rather than how close the assessments are related, because this provides hints for further improvements.

5.4.2. Assessing the Learner's Behavior

Adaptive link annotation and adaptive curriculum sequencing try to prevent learners from working at inadequate material. Users should perform worse on concepts that have the status *not ready* in the user model. In an online experiment we observed the behavior of subjects in reference to their current learning state.

Method and Criteria

We collected data from five online courses in different domains that had all been developed with the NetCoach authoring tool. These courses included the *HTML-Tutor*

as well as four introductory courses on psychology such as problem solving (*Problemloesen*), Piaget's developmental psychology (*Piaget*), communication (*Kommunikation*), and interpersonal perception (*Personenwahrnehmung*). 3501 users (both students and visitors from the internet) interacted at least 15 minutes with one of the courses. Everyone was free to choose which concepts to work at. For each concept, we observed both the learners' behavior and their learning state. For each test group we specified the minimum number of items that were required to solve the test group. The mean proportion of correct answers ($\bar{c}$) was computed for those who were assumed to be prepared for this concept (pre) (i.e, the current learning state of this concept was either *suggested* or *solved*) and for those who were assumed to have some missing prerequisites ($\neg$pre) (i.e., the current learning state of this concept was *not ready*).

comparison of test item answers and user model

Experimental Results and Discussion

Figure 5.10 shows that learners who are supposed to be prepared for a concept performed better on the test items than those who were not fully prepared. Note that all students had the same information about the concept, the only difference between the groups is that the latter did not fulfill all prerequisites for this concept, because the learners did not follow the suggested path. For two courses (*Piaget* and *Problemloesen*) we were not able to demonstrate a statistical difference between the groups (see Table 5.9). However, the statistical analysis makes obvious that the effect is not in the opposite direction.

From a statistical point of view, it would have been desirable to reduce the variance within the groups to get a clearer picture of the relevant effects. A considerable amount of variance is probably caused by varying difficulties of the test groups. While some test groups are easy to solve and therefore the mean proportion of correct answers is high, other test groups are more difficult. Thus, we computed separate t-tests for every concept in the *HTML-Tutor* as well. Naturally, not all users worked on every concept which decreases the sample size rapidly and statistical significance becomes difficult to reach. The 38 concepts can be categorized in reference to the direction of the mean difference and to the significance of the result. As shown in Table 5.10 most concepts conformed with our hypothesis. Note that a concept-wise analysis for the other courses was not possible because sample sizes were too small.

In summary, the study suggests that NetCoach courses model the learner's knowledge correctly. The assumed learning state predicts at least parts of the learner's performance. However, the effect sizes are rather small. But if it was possible to improve the learning process by adapting to the user's knowledge this approach should at least be considered when a new learning environment is designed.

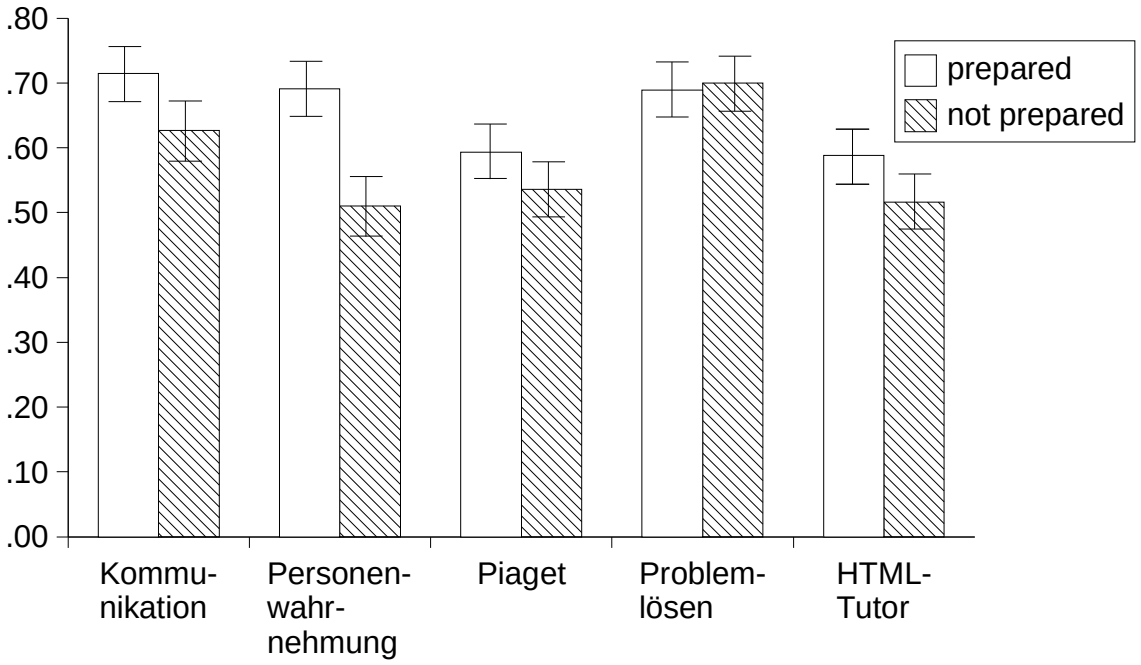


Figure 5.10.: Mean proportion of correct responses to test items for learners that were assumed to be prepared or not prepared for this concept. The error bars indicate $\pm$ one standard deviation. See Table 5.9 for detailed results

Table 5.9.: Comparison of user behavior in dependence of the assumed knowledge state. 3501 users (N_{users}) completed a total of $11770 + 1183 = 12953$ concepts in five different NetCoach courses. Learners who were prepared to work on a concept responded more often to test items correctly ($\bar{c}_{pre}$) than those who did not fulfill all prerequisites for this concept ($\bar{c}_{\neg pre}$). The number of included concepts (N_{pre} and $N_{\neg pre}$), the standard deviation (σ), and the significance (α) of the t-tests are reported. For non-significant results the test power ($1-\beta$) is shown

course	N_{users}	$\bar{c}_{pre}$	$\bar{c}_{\neg pre}$	N_{pre}	$N_{\neg pre}$	σ_{pre}	$\sigma_{\neg pre}$	α	$1-\beta$
Kommunikation	172	.71	.63	1665	125	.43	.47	.04	
Pers.-wahrnehmung	321	.69	.51	1629	132	.43	.46	.00	
Piaget	1004	.59	.54	4218	169	.39	.45	.10	1.0
Problemloesen	272	.69	.70	748	40	.43	.44	.87	.86
HTML-Tutor	1732	.59	.52	3510	717	.41	.42	.00	
Σ	3501			11770	1183				

Table 5.10.: Frequency of result types for 38 concepts in *HTML-Tutor*. We expected that the proportion of correct responses should be higher if the learner was prepared to work on this concept ($\bar{c}_{pre} > \bar{c}_{\neg pre}$). While most results were conform with this hypothesis only three of them were statistically significant

	$\bar{c}_{pre} > \bar{c}_{\neg pre}$	$\bar{c}_{pre} < \bar{c}_{\neg pre}$	Σ
significant	3	0	3
not significant	24	11	35
Σ	27	11	38

5.4.3. Discussion of Inference

accuracy of user model

These two studies (Section 5.4.1 on the congruence of system assumptions and external assessment and Section 5.4.2 on the displayed behavior of learners) outline a way for the evaluation of adaptive learning systems in general. The evaluated NetCoach courses seem to assess the learning state correctly. We were able to show that it is possible to evaluate the accuracy of assumptions about the user. Such evaluations might point to possible (and otherwise indiscoverable) improvements of adaptation.

The studies above evaluate the inference of user properties. Obviously, we do not generally measure adaptivity success. Assuring a correct user model is a prerequisite for the further adaptation process.

5.5. Evaluation of Adaptation Decision

Given that the user model is correct, different adaptation decisions are possible. In this section we explore two kinds of adaptations. First, we demonstrate that it is useful to skip chapters that are assumed to be learned based on a pre-test, because learners may save much time without deficits in the learning gain. Second, we compare the effects of different annotation and sequencing combinations.

5.5.1. Adapting to the Learners' Prior Knowledge

Online learning courses are used by people that differ widely in prior domain knowledge. Especially in further education and learning on demand settings, some learners will have a partial background of the course-topic while others are complete beginners.

opportunities

However, regardless of prior knowledge, everyone should have the same knowledge after course completion. On the one hand, users might get bored if they have to work on topics that they are already familiar with. On the other hand, they are probably not able to estimate whether they do really know everything on a topic of a course without having seen the chapters. Thus, letting users decide on their own whether they have enough knowledge or not might result in incomplete knowledge acquisition.

Moreover, prior knowledge has an impact on the learning gain. When constructing a hypertext, authors should consider the users' prior knowledge (Park and Hannafin, 1993). It might be useful to adapt the hypertext's structure (McDonald and Stevenson, 1998) or to provide different advisements (Shin, Schallert and Savenye, 1994).

In any case, such adaptations require the assessment of prior knowledge. NetCoach courses provide both a mechanism to assess the user's prior knowledge and to adapt the course accordingly. We evaluated the pre-test mechanism, i.e., the adaptation decision to skip chapters based on a pre-test, with the *HTML-Tutor*.

Method and Criteria

pre-test conditions

We observed a total of 140 users who accessed the public course from all over the world. Two groups of users were distinguished: the first group (no pre-test) ignored the pre-test and completed the chapters as usual, while the second group (pre-test presented) decided to answer the pre-test. A random selection of test items from subordinated chapters was presented to this group. Consequently most of them were advised to omit at least some subchapters.

knowledge test

At the end of the course, users completed a final test that included several test items on the pre-test chapters. If the pre-test assessment was successful the second group should know as much as the first group about the chapters, even though they did not read the contents. Moreover, we computed the time that each learner required to complete the chapters.

Obviously, 140 users are only a small subset of those observed in the previous studies. However, for this evaluation we had to select those who completed the final test, i.e., they completed the whole course, which takes a least 6–8 hours. Most users cancel before they reach this final objective. This is typical for online courses,

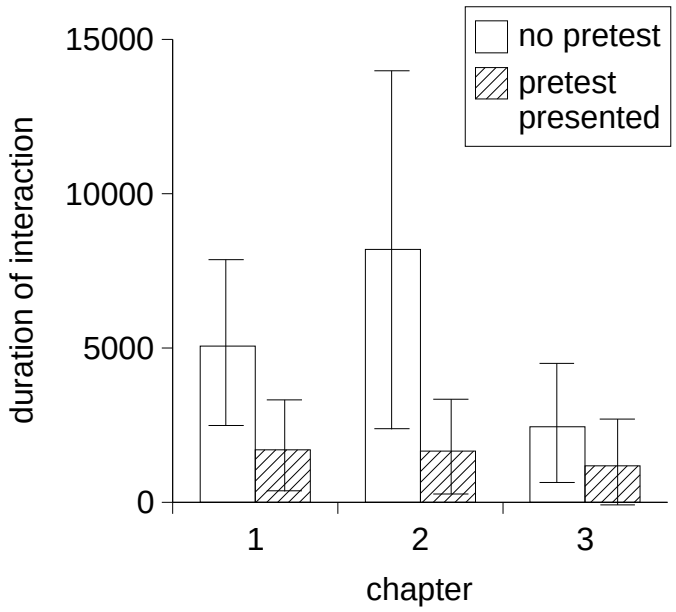


Figure 5.11.: Duration of interaction (seconds) in depending on the pre-tests. People who solved the pre-test on one of the three chapters required less time to complete these chapters. The error bars indicate one standard deviation ($\pm\sigma$)

because users visit the site voluntarily and are often interested in finding the answer to a specific question ('how can I include a picture in my HTML page') opposed to learning the complete offered content. As the course can be visited for free, many users just check out what the course is about and leave again.

Experimental Results and Discussion

We found that the pre-test group completed the chapters much faster than the standard group. For all three pre-tests that have been included in this analysis the mean duration of interaction was lower (see Figure 5.11). A 2-factor multivariate analysis of variance (MANOVA) yielded significant differences between these groups in terms of the required time for completing the chapters (see Table 5.11).

However, the analysis of the post-test shows that the pre-test users had at least as much knowledge on these chapters as the standard group. The MANOVA shows, that the pre-test group even scores slightly better. Their relative number of correct

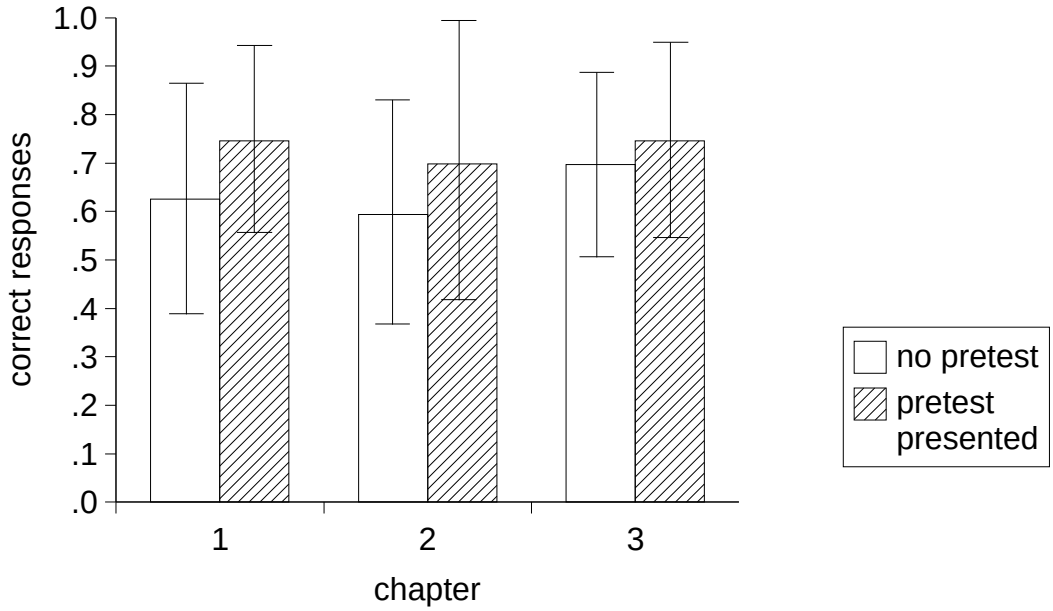


Figure 5.12.: Relative number of correct responses in the post-test in dependence of pre-tests. People who solved the pre-test on one of the three chapters gave equally or more correct responses to test items in the post-test. The error bars indicate one standard deviation ($\pm\sigma$)

Table 5.11.: Statistical results of 2-factor MANOVA for the effects of *pre-test presentation* and *chapter* on the *duration of interaction* and *knowledge*. A sample of 140 subjects was observed. For each factor the F-value (F), the degrees of freedom ($df_{effect,\ error}$), the statistical significance (α), and the effect size (η^2) are reported

	factor	F	df	α	η^2
duration	F1: pre-test presentation	31.07	1, 134	.000	.19
	F2: chapter	6.56	2, 134	.002	.09
	F1*F2	5.54	2, 134	.005	.08
knowledge	F1: pre-test presentation	3.58	1, 134	.061	.03
	F2: chapter	0.69	2, 134	.503	.01
	F1*F2	0.18	2, 134	.839	.00

responses was even higher for chapter 1 (subsequent t-test $t(71) = -2.05$, $\alpha = .044$). The remaining chapters did not differ significantly due to the small sample size (see Figure 5.12). Probably, the pre-test group was very familiar with the topic while the standard group forgot some aspects of the content while working at other chapters.

In summary, the pre-test group had at least as much knowledge about the chapters although they spent less time browsing these chapters. Note that the users saved up to 80% of the interaction duration, but performed about 10% to 20% better in the post-test. This could be an especially important benefit for learning on demand settings where people want to learn specific contents as efficiently as possible.

Our results show that the *HTML-Tutor* assesses the prior knowledge correctly, and that it is thus adequate to suggest to skip the already known chapters. Despite the fact, that people were adaptively guided to omit those chapters, they were able to answer test items on the chapter's contents even better than the standard group.

Thus, assessing knowledge with test items facilitates interesting adaptation opportunities. Adapting to prior knowledge is an important approach to increase the effectiveness and efficiency of learning courses and might even increase the users' satisfaction.

5.5.2. Comparison of Different Adaptation Decisions

To estimate the effect that stems from the adaptation, it is possible to compare different adaptation decisions. In principle a wide range of different decisions may be applied, however, in this study we only compared four combinations of annotation and sequencing.

Method and Criteria

In a 2×2-factor design we observed how users behave *with* and *without annotation*, and *with* and *without curriculum sequencing*. Between July and September 2002, all online users of the *HTML-Tutor* were assigned randomly to one of these four conditions. When annotation is switched off, all links to chapters are presented in the same color regardless of the users' current knowledge. In the *without sequencing* condition the system behaved in quite the same way as the standard *HTML-Tutor* version besides the fact that the 'next' button was not available, and the system did not suggest pages in any other way. As the fourth condition (with both annotation and sequencing) is identical to the standard version, we included the data of the previous sessions in the analysis, too, after checking that the mean values and variances of the two samples were equal.

dropouts

First, the number of dropouts was registered. We counted the number of users who completed the seven subsections of the first chapter. The completion rate was computed for each subsection separately.

number of concepts

Second, we observed the user behavior in terms of the number of concepts that have been visited during interaction. Moreover, we computed the number of concepts that have been visited in relation to the duration of interaction.

These measures were also assessed for *RR2000*, the course on German spelling rules. Between October 1999 and June 2000 a total of 5703 users visited the course. Analogous to the *HTML-Tutor* sample they were assigned randomly to one of the four conditions. We expected that the more linear structure of *RR2000* (as shown in Table 5.4) will result in smaller differences between the adaptation conditions.

overall impression

In addition, we evaluated the feedback questionnaire that was presented at the end of the HTML-Tutor. As already described in Chapter 4.4.2, we extracted the users' overall impression of the system by computing the mean value of four usability ratings on a 10-point scale. The satisfaction with the adaptation was assessed by the mean value of four questions that are concerned with the effect of different adaptation aspects (see Chapter 4.4.2 for the exact formulation of these rating scales). The users had to rate the success of the adaptation in terms of the page suggestions, the

subjective adaptivity success

annotation, and the number of items in a test group, as well as the general adaptivity success.

Experimental Results and Discussion

dropouts

We only found slight differences between the four adaptation conditions. The distribution of dropouts across the subsections with test groups in the first chapter is shown in Figure 5.13. Though the non-adaptive condition (*without annotation* and *without sequencing*) yields the most dropouts, there is no statistical difference between the four groups (repeated measurement ANOVA with $N = 13733$ for annotation and sequencing). Obviously, the adaptation did not stimulate the users very much to keep on working. In *RR2000*, the no-annotation-groups even completed slightly more chapters (repeated measurement ANOVA with $N = 5703$, $F_{1,\ 5699} = 6.57$, $\alpha = .010$, and $\eta^2 = 0.001$). The peaks in the dropout curve (Figure 5.14) result from the fact that some users skipped subsections and turned directly to subsequent sections.

For the analysis of the number of concepts and the subjective ratings users who interacted less than 15 minutes were excluded to get a more homogenous sample and to assure that the users really worked with the courses.

number of concepts

The number of concepts that have been visited was equal for all four adaptation conditions. While the mean values differ slightly (Table 5.12), the analysis of variance reveals that users in all conditions requested about the same amount of material, i.e., they visited a equivalent number of concepts (Table 5.13).

concepts per minute

Surprisingly, the number of concepts per minute is increased if people receive annotations but no sequencing. The statistical analysis manifests this difference, though the effect size is very small. In *RR2000*, the adaptation conditions had no impact at all. In other words, the users in this course requested the same amount of material and browsed the chapters at the same pace regardless of the adaptation (Table 5.14).

previous studies

These results do not agree with previous studies. While Klein (2000) found no difference for different types of sequencing and annotation conditions, Weber and Specht (1997b) report that especially unexperienced users benefit from annotation and sequencing. However, even a sample split according to the self reported experience with computers and the internet did not change our results. All three studies have been conducted with NetCoach courses in different domains. While Klein (2000) implemented a huge course on cognitive psychology (INCOPS), Weber and Specht (1997b) explored programming in LISP. The different contents, its structure, or characteristics of the samples might be responsible for these inconsistent findings. Nevertheless, the value of adaptive annotation in hypermedia has been reported several times (see Eklund and Brusilovsky, 1998, for an overview) with similar ex-

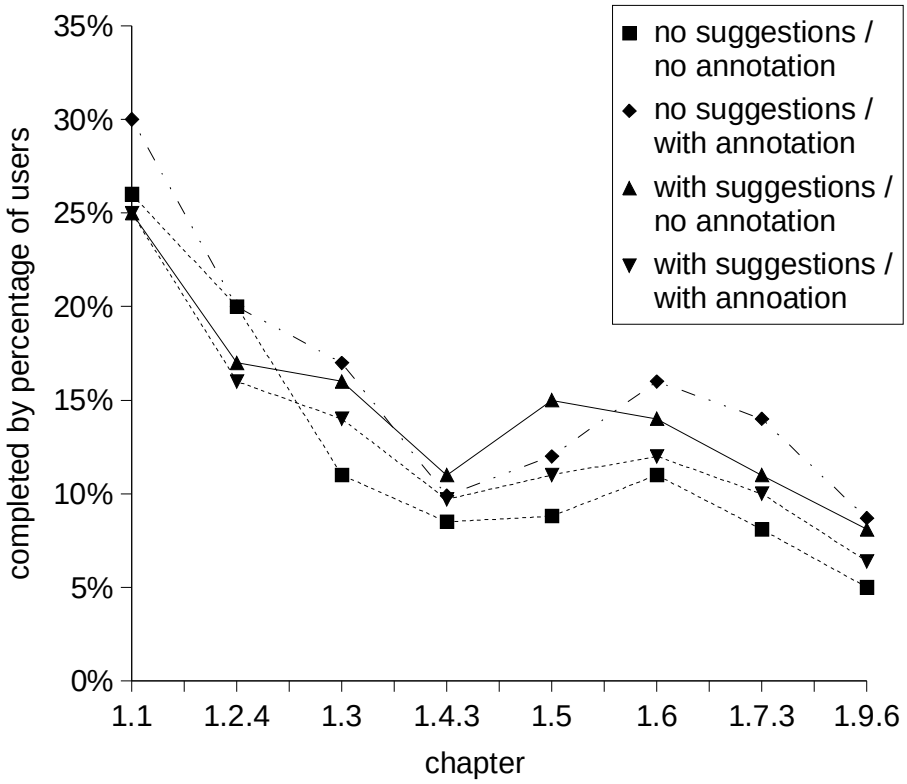


Figure 5.13.: Percentage of users who completed (i.e., solved) the eight subsections of chapter 1 in the *HTML-Tutor* with a test group, categorized by the adaptation condition ($N = 259 + 252 + 283 + 10218 = 11012$ in the same order as in the legend)

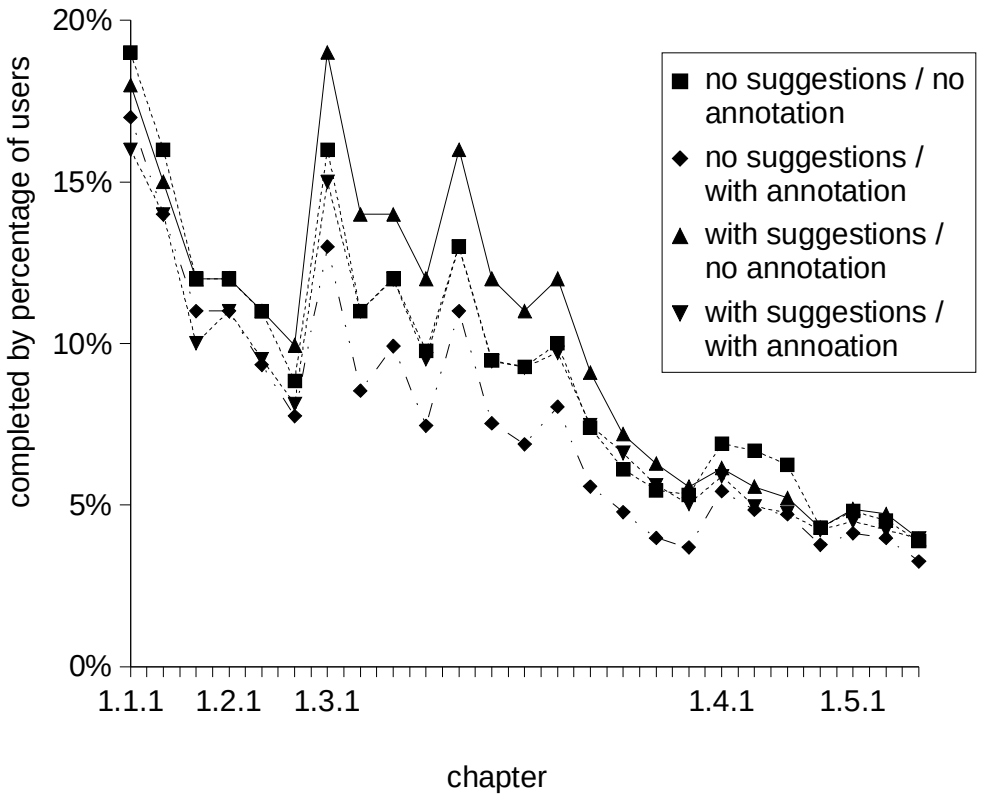


Figure 5.14.: Percentage of users who completed (i.e., solved) the 25 subsections of chapter 1 in *RR2000* with a test group, categorized by the adaptation condition ($N = 1392 + 1381 + 1418 + 1512 = 5703$ in the same order as in the legend)

Table 5.12.: Mean values of number of visited concepts, visited concepts per minute, and subjective ratings in reference to the four adaptation decision conditions of link annotation and curriculum sequencing in the courses *HTML-Tutor* and *RR2000*

		without annotation		**with annotation**	
		without sequencing	**with sequencing**	**without sequencing**	**with sequencing**
HTML-Tutor	**concepts**	34.83	39.66	42.55	35.70
	concepts per minute	0.82	0.74	0.97	0.72
	overall impression	5.25	5.83	7.00	6.17
	successful adaptation	8.75	6.72	5.33	6.04
RR2000	**concepts**	31.64	29.59	30.28	30.28
	concepts per minute	0.90	0.80	0.87	0.86

Table 5.13.: Statistical results of four 2-factor ANOVAs for the effect of adaptive annotation and curriculum sequencing in the *HTML-Tutor* on four variables: First, the number of concepts that have been visited by the users. Second, the number of concepts that have been visited by the users in relation to the duration of interaction. Third, the overall impression of the course computed by the mean of four usability ratings on a 10-point scale. And finally, the mean satisfaction with the adaptivity, measured with four questions on a 10-point rating scale. For each factor the F-value (F), the degrees of freedom ($df_{effect,\ error}$), the statistical significance (α), and either the effect size (η^2) or the test power ($1-\beta$) for an effect size $f = 0.25$ and $\alpha = .05$ are reported

	factor	F	df	α	η^2	$1-\beta$
concepts	F1: annotation	0.343	1, 2750	.558		1.000
	F2: sequencing	0.099	1, 2750	.753		1.000
	F1*F2	3.306	1, 2750	.069		1.000
concepts per minute	F1: annotation	1.798	1, 2750	.180		1.000
	F2: sequencing	2.503	1, 2750	.001	.004	
	F1*F2	2.937	1, 2750	.087		1.000
overall impression	F1: annotation	0.440	1, 112	.509		.760
	F2: sequencing	0.006	1, 112	.938		.760
	F1*F2	0.202	1, 112	.654		.760
successful adaptation	F1: annotation	1.366	1, 103	.245		.726
	F2: sequencing	.142	1, 103	.707		.726
	F1*F2	.605	1, 103	.439		.726

Table 5.14.: Statistical results of two 2-factor ANOVAs for the effect of adaptive annotation and curriculum sequencing in *RR2000* on two variables: First, the number of concepts that have been visited by the users. Second, the number of concepts that have been visited by the users in relation to the duration of interaction. For both factors the F-value (F), the degrees of freedom ($df_{effect,\ error}$), the statistical significance (α), and either the effect size (η^2) or the test power ($1-\beta$) for an effect size $f = 0.25$ and $\alpha = .05$ are reported

	factor	F	df	α	η^2	$1-\beta$
concepts	F1: annotation	0.049	1,1508	.825		1.000
	F2: sequencing	0.457	1,1508	.499		1.000
	F1*F2	0.449	1,1508	.503		1.000
concepts per minute	F1: annotation	0.206	1,1508	.650		1.000
	F2: sequencing	1.272	1,1508	.260		1.000
	F1*F2	0.669	1,1508	.414		1.000

perimental designs. We argue, that further explorations are required to identify the underlying mechanisms.

subjective ratings

Neither overall impression nor subjective adaptivity success differed across the four conditions (Table 5.13). However, the sample size for this analysis was quite small due to a data assessment failure and thus the test power remains too small for definite statements. Nevertheless, asking the users for adaptivity success seems to induce weird effects. The group who interacted with a completely non-adaptive system (without annotation and without sequencing) rated the success of the adaptation very high. In other words, though this group could not observe anything that actually adapted, they were quite satisfied with the result. We argue, that subjective ratings are obviously not useful for evaluation here, because there is no baseline of what is successful or unsuccessful as long as the users have not seen another version that behaves differently.

5.5.3. Discussion of Adaptation Decision

In summary, the differences between the various adaptation mechanisms are small in terms of the explored criteria. Even if we take into account that adaptation strategies like annotation and sequencing are quite simple, and that the used criteria might be

misleading, it is somewhat surprising that even the no-adaptation condition scored well. We argue that this emphasizes the necessity of the evaluation framework and of the selection of adequate criteria even more. As the previous evaluation step has shown, the inferred user properties are correct the results suggest to test other adaptation decisions that are based on the same user properties as well or to define more specific criteria that are able to extract the underlying effects better than the measures that have been used here.

However, the effects of the adaptation to prior knowledge are evident. This is a clear advantage of the adaptivity that is especially of importance in settings were learners differ widely in prior knowledge, e.g., in further education. This might become a crucial field of application for adaptive learning systems.

5.6. Evaluation of Total Interaction

As a final evaluation step, we present some summative data how the system and the users behave in real world settings and how this influences the usability.

5.6.1. System Behavior

First, we observed the system behavior in real interactions. Does the system really adapt to the user or is the behavior still quite static, because learners do not differ as much as expected?

We observed a total of 2438 users, who interacted with the *HTML-Tutor* for at least 900 seconds. This time limit was chosen to sort out visitors who were looking for different contents and to assure, that the users really learned with the system.

solved concepts

856 of these learners had at least one concept in their user model that was *solved but not read*, i.e., they never visited this page. This happens if users work on pre-tests. Thus 35% of the users had an advantage of the adaptation, because they were allowed to skip these chapters.

inferred concepts

379 of the learners in the sample had at least one concept in their user model that was *inferred* to be learned but that was neither *solved* in a test group nor visited. That is, 15% of the users profited directly from these inferences, because they had not to work on these chapters to reach the learning objective.

The 2438 users worked on 84702 chapters. 2144 (2.5%) of these chapters were *inferred* from other chapters and have never been visited. 4754 (5.6%) chapters were *solved* in pre-tests and have not been visited. Thus, the system made predictions for 8.1% of the chapters.

Table 5.15.: The number of pages (p) that have been requested by 1833 users is categorized according to the state of that page in the user model. The mean number of pages ($\bar{p}$) and the relative proportion of each category is reported as well. The exact meaning of the states is described in Table 5.2 on page 82

	$\bar{p}$	p	proportion
ready	4.95	9073	22.3%
visited	1.22	2234	5.5%
suggested	9.98	18294	45.1%
solved	.71	1294	3.2%
inferred	0.09	174	0.4%
not ready	4.78	8755	21.6%
other	.43	783	1.9%
Σ	22.15	40607	100.0%

requested pages

Moreover, it is possible to categorize the pages that have been requested by the users in terms of the state of the page in the user model. Due to technical reasons, only 1833 users are included in this evaluation. Table 5.15 shows that 21.6% of the requested pages were *not ready*. Thus, in one of five cases the system warned the user that there are prerequisites for this page which were not yet fulfilled. As seen in chapter 5.4.2, users who visited a *not ready* page score worse on the test items. Thus, these warnings were obviously justified.

These data are important, because the previous evaluation steps have shown that the *inferred* and *solved* chapters are actually known. The descriptive data reported here show that under real life conditions the system actually adapts to the learners, and that thus these users probably have or at least might have an advantage from this adaptation. The impact of the adaptation is considerable: one of five pages were supplemented dynamically with a warning that *the system assumes that there are missing prerequisites for this page* and another 8.1% of the pages was assumed to be solved before the learners visited these pages. Thus, adaptivity in the *HTML-Tutor* is not just a theoretical consideration of what should be adapted but is actually relevant in real world interactions, because the system obviously behaves quite different to a static system.

5.6.2. User Behavior and Usability

duration of interaction

On average the users interacted with the course for 60.5 minutes, while some learners worked for up to 14 hours. Usually, the users followed the system suggestions (45.1%), and only few returned to pages that they had seen before (see Table 5.15).

In Chapter 5.5.1 we already demonstrated that many learners benefited from the adaptation to prior knowledge, because they saved much time without skipping relevant content. The number of dropouts and the amount of requested material has been reported in Chapter 5.5.2: the users worked on 13.68 concepts on average and completed 1.30 concepts per minute.

overall impression

adaptivity success

By evaluating the subjective feedback we found that the learners rated the system quite good. The overall impression of the system on a 10-point scale (0–9) equals 6.17 (Table 5.12) which is better than the feedback of users to other NetCoach courses (Table 4.2). The rating of the adaptivity success is also quite good (6.04), though the variation of adaptation conditions showed no effect here (Chapter 5.5.2).

5.6.3. Discussion of Total Interaction

In summary, the subjective ratings are good, but it is not clear whether this is an effect of successful adaptivity or just an overall impression, because the different adaptation conditions did not differ in terms of these ratings. As already outlined above, subjective ratings are problematic for the evaluation of adaptivity success if the users do not have any baseline or scale of what successful or unsuccessful means.

However, it is evident that the adaptivity has an important impact on the system behavior, i.e., the interaction changes considerably due to the assumptions of the system.

6. Discussion

6.1. Generalization of experimental results

Finally, we discuss the implications of the results reported above for other NetCoach courses and other adaptive learning systems as well as for adaptive systems in general.

6.1.1. Implications for Adaptive Learning Systems

The results differ in terms of their impact. While the effects of adaptation to prior knowledge are rather high, adaptive annotation and guiding seem to influence the users only slightly. However, we did not expect that changing the color of buttons would have an extreme impact on the learning gain, though some findings on these adaptation techniques are rather encouraging (Eklund and Brusilovsky, 1998). Learning depends on many factors and supporting the navigation and sequencing of material is only one of many aspects that are important for a smooth interaction and the learning process.

The *HTML-Tutor* seems to assess the learner's knowledge correctly. We found, that the congruence of the user models with an external assessment is pretty high, and the adaptation to prior-knowledge would not be that successful if there were important misinterpretations. Thus, other adaptation decisions than the used ones might be considered. For example, instead of just pointing out that there are missing prerequisites the system could provide short summaries of the prerequisite pages (Lippitsch, Weibelzahl and Weber, 2002). This could help the learner to estimate whether she is already familiar with the topic or not. This can be seen as a general trend: current intelligent systems tend to support the user and ask for confirmation of assumptions instead of patronizing the learner as if the system was omniscient. Furthermore, adaptation decisions that are based on the learners' knowledge include different ways of guiding (e.g., forcing the user to go to the suggested page or explicitly explaining, why a certain page is suggested). These adaptation decisions should

other adaptation decisions

be tested not only with NetCoach courses but with other adaptive learning systems as well.

other assessment methods

Second, the fact that NetCoach assesses the knowledge properly imposes the question whether techniques that are used in other systems are equally successful. The studies reported above demonstrate that test items may assess the learners' knowledge correctly. Simple methods like "visited pages are known" might fail here, but empirical evidence is required for such a claim. On the other hand, more complex techniques, such as modeling the user's knowledge with an adaptive test that is based on the item response theory (Guzmán and Conejo, 2002) need to support their claim that these systems assess the knowledge more exactly.

generalization across domains

Third, some of the studies report results across domains. However, it is not obvious which domains the results are transferable to. HTML is a very structured content with clear knowledge dependencies. Other contents are much more linear, and guiding and annotation will probably be of less importance there. For instance, the *RR2000* course on German spelling rules is for the most part a collection of subjects that are quite independent. Only some subchapters are interdependent, e.g., several sections are prerequisite for a summary with a test group.

We claim that a formal method that could categorize a content in terms of its usefulness and appropriateness for adaptation would be highly appreciated. It would allow for estimations of whether a domain or subject should be presented in an adaptive way or not. However, as long as such a method is not available (and we doubt whether the development of such a method is possible at all) empirical evaluations in different domains are still required and absolutely necessary. Adaptive learning systems will become standard only if their usefulness and efficiency has been proved. We hope that the reported studies and especially the evaluation framework will encourage further investigations.

costs and benefits

Finally, the studies contribute to the discussion of costs and benefits of adaptive learning systems. Throughout this work we explored and discussed the accuracy and success of adaptivity. Apart from this scientific point of view, in economic settings the relation of efforts and achievements is of importance. Do the results that are reported here justify the efforts that are needed to implement the adaptivity? Annotation and curriculum sequencing as a supplemental feature seem to have only a small impact, which is not sufficient to be added to an existing system. However, there are two important exceptions.

First, the additional efforts that are needed to generate a NetCoach course are very little. As described above, the authoring system supports the complete course generation process. Authors prepare the material as usual. The only additional task is to write test items and to specify prerequisite and inference relations between chapters. Thus, the efforts for adaptivity in NetCoach courses (and probably other author-

ing systems) are pretty low, and a good course should be based on such relations implicitly, anyway. Even for big courses like the *HTML-Tutor* the specification of relations required low efforts in comparison to the authoring of the material. The possible benefit will justify these efforts and the individual feedback is definitely a comfortable feature for the learners.

Second, curriculum obviously has a high potential in terms of adaptation to prior knowledge. We demonstrated that learners might save a lot of time if they are advised to skip chapters that are already known. For on-the-job training and further education this is a factor of high relevance, because the efficiency of learning is considerably increased. We claim that the adaptation to prior knowledge will be an important field of application for adaptive learning systems.

6.1.2. Implications for Adaptive Systems in General

It might be surprising for some readers, that the straightforward adaptation techniques that are used in NetCoach yield quite good results. This challenges more complex techniques that require more knowledge engineering or more costly implementations. Are there user properties that actually require a more complex approach, because a simple rule based technique is unable to model it exactly? And is this exactness of importance for the interaction? For example, it is possible to adapt to the problem solving skills of learners by simulating the learning process with a cognitive model (Corbett and Anderson, 1992). For teaching programming this approach yielded impressive results (Corbett, 2001). However, a less elaborate approach could also support the learners to a certain degree. Is it really necessary to run a full blown cognitive model in the background to adapt to the learner? We argue, that the evaluation framework could be useful to answer these questions by comparing different modeling approaches (or inference mechanisms) in terms of their correctness (evaluation of inference) as well as in regard of the impact on the interaction (evaluation of interaction). This would allow to estimate the effects that are actually achieved by the adaptation in different ways.

6.2. Discussion of the Evaluation Framework

This work applied the complete evaluation framework to an adaptive learning system. In this chapter we summarize our experiences with the framework and consider the applicability of the framework to other adaptive systems.

6.2.1. Experiences with the Framework

efforts of evaluation studies

Undoubtedly, good evaluations require a certain amount of efforts, and an elaborated and detailed evaluation as it is presented in this work is certainly impossible for every adaptive system. However, we argue that the complete procedure is not required all the time. Once evaluations become more frequent, it would be possible to take previous evaluations of the same system or of other systems as a basis for further investigations. For example, if several studies proved that knowledge on HTML can be assessed very accurately with test items, subsequent evaluations of other systems with the same assessment method could focus on different variations of adaptation decisions.

overlapping steps

Furthermore, in our experience step 3, the evaluation of adaptation decision (Chapter 5.5), and step 4, the evaluation of interaction (Chapter 5.6), are very similar. The main difference is the perspective on the results, but the methods and criteria might overlap. While the first one compares different possibilities, the latter is a kind of summary that estimates the quality of the total interaction. Thus, it might be difficult to categorize some studies definitely. However, the framework aims at the opposite direction: What has to be done to guarantee successful adaptivity? The framework gives clear answers to this question.

layered evaluation approach

Finally, the layered evaluation has passed the test. Especially for the last two steps, the results were much more non-ambiguous to interpret because the previous studies demonstrated that the underlying data are accurate. For example, the description of the system behavior gets much more importance if the high frequency of adapted pages refers to adaptations that are proved to be valid and useful.

6.2.2. Applicability of the Framework to Other Systems

In general we claim that the evaluation framework should be applicable to all adaptive systems. We present some minor limitations that might be of interest for planning further evaluations.

evaluation of input data

As already stated above, in some cases the evaluation of input data is not required because the reliability is guaranteed anyway. For example, the number of mouse clicks is well defined and can be assessed without limitations objectively and reliably.

evaluation of inference

Second, there might be limitations in the evaluation of inference for some kinds of inference mechanisms. For example, adaptive systems that are based on machine learning algorithms do not represent the user properties explicitly, but connect input data and adaptation decision directly, separated for every adaptivity feature (e.g., Krogsæter, Oppermann and Thomas, 1994; Pohl, 1997, 1998). Thus, it is obviously

not possible to compare the user model with an external assessment and step 2 (evaluation of inference) and 3 (evaluation of adaptation decision) have to be integrated. However, in these systems the evaluation of input data is probably even more important, because the inference rules have also been learned on the basis of these data. As second example we introduce adaptation techniques that use Bayesian Networks. It might be difficult to test the complete assumptions of these networks, i.e., not only parameter values of user properties but probability distributions. For example, a system might infer a .95-probability of the fact that the user knows the command 'more' (Jameson, 1996). An external assessment would usually categorize this user property as either *known* or *unknown* without any probabilities. However, these systems usually use a cut-off value for the subsequent adaptation decision, i.e., only if the probability exceeds .90 it is assumed that the user actually knows the command and the adaptation takes place regardless of the exact probability value. Thus, a comparison of user model and external assessment could also be limited to these cut-off values.

evaluation of adaptation decision

Different adaptation decisions are probably applicable in all adaptive systems. In some cases the possible variation might be little, but it will still help to estimate the effect that stems from the adaptivity.

Finally, the implementation of EASy–D (Chapter 3.4) demonstrated that many different evaluations can be categorized according to the framework which supports our claim that the framework is independent of the domain and system type.

6.3. Future Perspectives

As future perspective on how the approaches that are described in this work could proceed we outline four aspects.

exploring adaptation decisions

First, as introduced before, the NetCoach courses as well as other adaptive learning systems might be improved if different adaptation decisions are compared in more detail. Adaptive annotation, adaptive curriculum sequencing, adaptive feedback, or adaptive testing might take place in different flavors and with many underlying rationale. The appropriateness of these adaptation decisions for different kind of learners, in different setting with different contents could improve adaptive systems considerably.

evaluating across different contents

Second, we claim that future evaluations should emphasize at the same time the consideration of different contents. For instance, evaluations of adaptive learning systems should include courses of different topics and evaluations of adaptive product recommendation systems should contain different product categories. The quality of the results and the possibility to generalize the effects is obvious. We

strongly encourage this kind of investigations, because this would not only demonstrate the adaptivity effects but also explore how much the adaptivity depends on the context. As adaptivity may depend on the domain or topic this kind of design would help to abstract from this factor or at least help to explore the limits of an approach. We guess that this dependency is much stronger than sometimes suggested, though the empirical basis for this claim is currently quite thin (Chapter 5.5.2).

adaptivity for everyone

Third, in addition to the adaptive support of the user it will be more and more important to support those people who implement and maintain these systems. This holds especially for adaptive learning systems, where authors as well as tutors need to understand and predict how the adaptivity mechanism will behave and how they can use these functions. For example, a tutor might observe that a learner skips several subsections after working on a pre-test. The tutor should be advised whether the system suggested to skip these sections or not. Authors must be able to predict how the system behavior will change if they add new concept relations. But also editors of an adaptive news system, authors of adaptive help texts etc. should be supported. In some cases this might be in an adaptive way again (e.g., by considering the authors experience), in other cases a more appropriate way might be to visualize or explain the adaptivity results. For instance, an adaptive learning system might provide the structure of concept relations (similar to the tables in Appendix A.2 and A.3), which might help to understand the system behavior.

adaptivity competitions

Finally, we propose to announce a kind of competition for the best adaptivity approach. Certainly, this is a major but also appealing task. A procedure that has been found to be successful in other communities, could become a regular approach for the user modeling and adaptive hypermedia scientists, too. Standard domains or tasks as well as suitable criteria are announced and different adaptation mechanisms and approaches can be compared directly. This would both allow to explore advantages and disadvantages of different approaches as well as making adaptivity more popular (or at least noticed by a broader audience) as the results of these competitions are much easier to explain to non-experts than a conglomerate of studies that are not directly related. We would highly appreciate it if such competitions could be established at future User Modeling Conferences.

Appendix

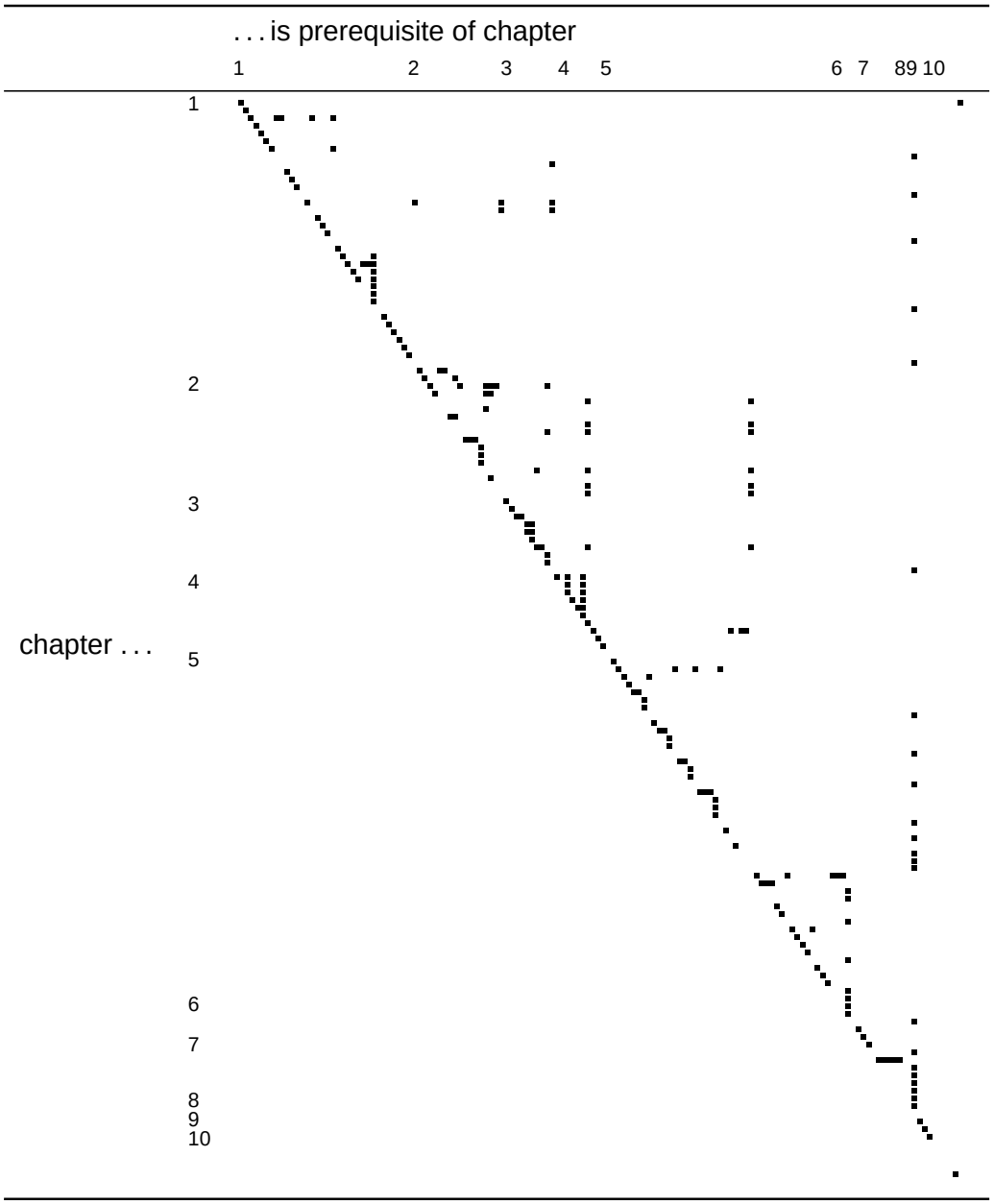
A. Description of the HTML Tutor

A.1. Table of Contents

We provide the complete table of contents of the *HTML-Tutor*. The numbers are identical to those used in the text when referring to a specific chapter. The label [T] indicates that this chapter has a test group and thus the learners' knowledge is assessed with test items.

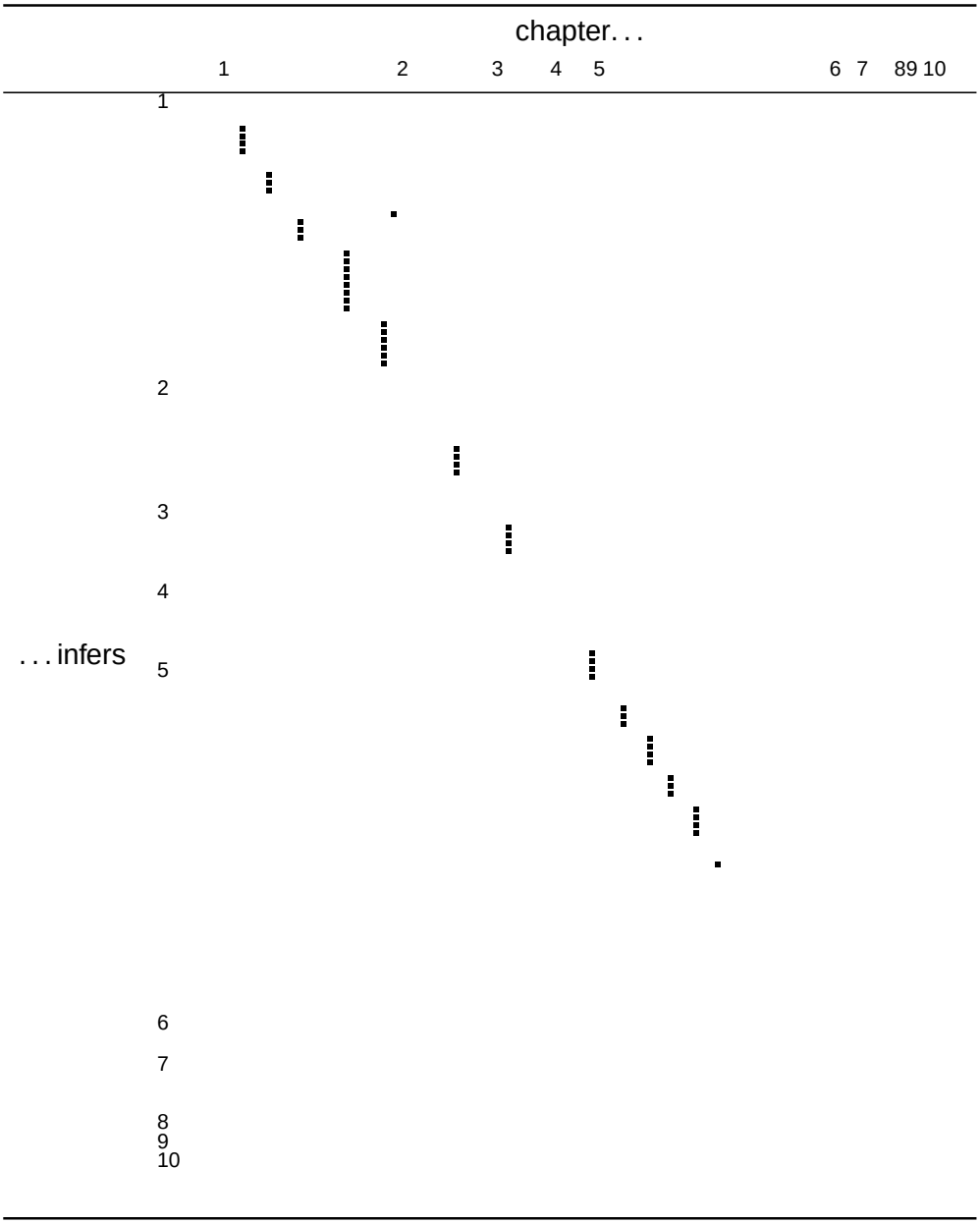
A.2. Structure of Prerequisite Relations

Structure of prerequisite relations in the *HTML-Tutor*. Each row and each column shows one of the 138 chapters (main chapters 1 to 10; subchapters are not labeled). A dot (.) indicates that this chapter (vertical) is prerequisite of another chapter (horizontal). Certainly, the table cannot list all chapter names, thus a list of the complete names is presented in Appendix A.1



A.3. Structure of Inference Relations

Structure of inference relations in the *HTML-Tutor*. Each row and each column shows one of the 138 chapters (main chapters 1 to 10; subchapters are not labeled). A dot (.) indicates that this chapter (vertical) is inferred by another chapter (horizontal). Certainly the table cannot list all chapter names, thus a list of the complete names is presented in Appendix A.1



B. Post-Test

Chapter 5.4.1 contains a study on the assessment of the learners' knowledge with an external test. Below we give a translation of this test. Afterwards, the original version is display on pages 136 to 138.

Final Test

1. Please, create an HTML file called *abschluss1.html*. When looking at this file with a WWW browser, this page should look similar to *Figure 1*.

   - The horizontal line takes one half of the page width.
   - Style-sheets are not required. It will be sufficient to use simple HTML tags.
   - Section 'Kapitel 3' is a little bit tricky. Only those who like riddles should try to solve it.

2. Create a second HTML file called *abschluss2.html*, please. When looking at this file with a WWW browser, this second page should look similar to *Figure 2*.

   - The link 'Tabellen' points to the table ('Kapitel 2') in *abschluss1.html*.
   - Include the picture *logo.gif* in your file. You will find this image in the directory *public_html/schluss/img* on your network drive.
   - Clicking on the image will direct the user to the Homepage of the University of Education Freiburg.

3. Transfer the files *abschluss1.html* and *abschluss2.html* to the directory *public_html/schluss* on your network drive.

4. Please explain, why HTML is not appropriate for enforcing a specific layout.

5. *Listing 1* shows the source code of a HTML file. Please, identify all mistakes. Indicate the line number and the correct syntax (or what was probably meant).

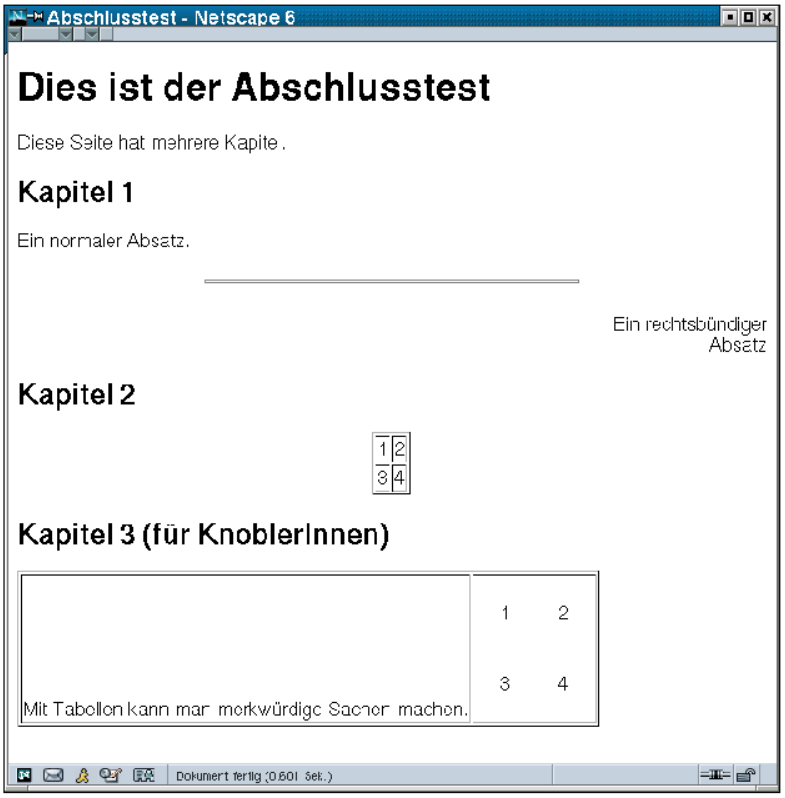
   Example: *line 27: this is correct *

Name:_____

Abschlusstest

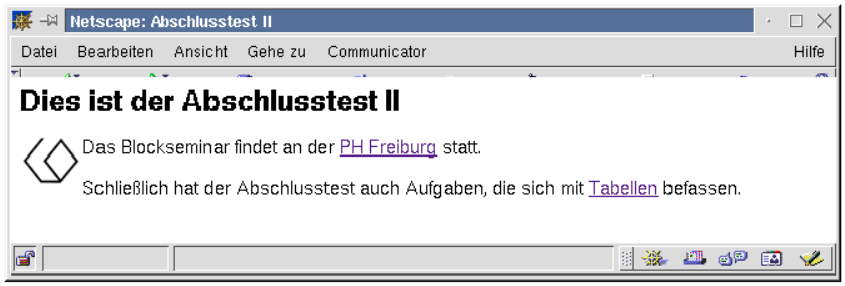
1. Erstellen Sie eine HTML-Datei `abschluss1.html`. Diese Seite sollte, wenn man sie mit einem WWW-Browser betrachtet, in etwa so aussehen wie *Abbildung 1*.
   - Die Trennlinie nimmt etwa die Hälfte der Breite ein.
   - Style-Sheets sind nicht erforderlich, es genügt eine einfache HTML-Formatierung.
   - Der Abschnitt "Kapitel 3" und die dazugehörige Tabelle ist als freiwillige (knifflige) Zusatzaufgabe gedacht und muss nicht bearbeitet werde.

Abbildung 1



2. Erstellen Sie eine zweite HTML-Datei `abschluss2.html`. Diese Seite sollte, wenn man sie mit einem WWW-Browser betrachtet, in etwa so aussehen wie *Abbildung 2*.
   - Der Link "Tabellen" verweist auf die Tabelle ("Kapitel 2") in der Datei `abschluss1.html`
   - Binden Sie das Bild als `logo.gif` ein. Sie finden diese Datei im Verzeichnis `public_html/schluss/img` auf Ihrem Netzlaufwerk `W923131B`.
   - Ein Klick auf das Logo soll zur Homepage der PH führen.

Abbildung 2



3. Legen sie die Dateien `abschluss1.html` und `abschluss2.html` in das Verzeichnis `public_html/schluss` in ihrem Netzlaufwerk-Verzeichnis auf `W923131B`.

4. Begründen Sie kurz, warum HTML nur bedingt geeignet ist, um ein bestimmtes Layout zu erzwingen.

5. *Listing 1* zeigt den Quelltext einer HTML-Datei. Finden Sie alle Stellen, die fehlerhaft sind. Geben Sie jeweils die Zeilennummer mit dem Fehler, und die korrekte (oder vermutlich gemeinte) Syntax an.

   Beispiel: *Zeile 27: so ist es richtig *

Listing 1

```
 1 <html>
 2       <head>
 3           Blockseminar
 4       </head>
 5       <body>
 6           <ül>Überschrift</ül>
 7           <p>
 8           Dies ist ein Verweise auf
 9           <a name="andereseite.html">
10           eine andere Seite</a>.
11           </p>
12           <p>Bilder lassen sich sehr
13           einfach einbinden:
14           <image src="logo.gif" align="top">
15           </p>.
16
17  </html>
```

C. The C-Contingency-Coefficient

The χ^2-contingency-coefficient C is designed analogous to the product-momentum-correlation but for two categorial variables. It is based on χ^2-contingency tables. For two categorial variables with G respectively H categories, given the expected (e) and empirical (n) frequencies in all cells, the coefficient is computed in the following way:

$$\begin{aligned} C &= \sqrt{\frac{\chi^2}{\chi^2 + N}} \\ \text{where } \chi^2 &= \sum_{g=1}^{G} \sum_{h=1}^{H} \frac{(n_{gh} - e_{gh})^2}{e_{gh}} \\ \text{and } e_{gh} &= \frac{n_g \cdot n_h}{N} \end{aligned}$$

C may reach values $C > 1$. It has thus to be normalized with the maximal value of C. Given $G > H$, the corrected coefficient C_{corr} is computed by

$$\begin{aligned} C_{max} &= \sqrt{\frac{G-1}{G}} \\ C_{corr} &= \frac{C}{C_{max}} \end{aligned}$$

Bibliography

Ambrosini, L., Cirillo, V., and Micarelli, A. (1997). A hybrid architecture for user-adapted information filtering on the World Wide Web. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 59–61. Vienna, New York: Springer. Available from http://um.org.

Ardissono, L. and Goy, A. (1999). Tailoring the interaction with users in electronic shops. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 35–44. Vienna, New York: Springer.

Arend, U. (1991). Analysing complex tasks with an extended GOMS* model. In Tauber, M. J. and Ackermann, D. (Eds.), *Mental Models and Human Computer Interaction*, volume 2, pages 115–133. North-Holland: Elsevier.

Bares, W. H. and Lester, J. C. (1997). Cinematographic user models for automated realtime camera control in dynamic 3D environments. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 215–226. Vienna, New York: Springer. Available from http://um.org.

Beck, J., Stern, M., and Woolf, B. P. (1997). Using the student model to control problem difficulty. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 277–288. Vienna, New York: Springer. Available from http://um.org.

Benyon, D. R. (1993). Adaptive systems; a solution to usability problems. *User Modelling and User Adapted Interaction*, *3*(1), 65–87.

Benyon, D. R. and Murray, D. M. (1993). Adaptive systems; from intelligent tutoring to autonomous agents. *Knowledge-Based Systems*, *6*(4), 197–219.

Berthold, A. and Jameson, A. (1999). Interpreting symptoms of cognitive load in speech input. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 235–244. Vienna, New York: Springer.

Billsus, D. and Pazzani, M. J. (1999). A hybrid user model for news story classification. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 98–108. Vienna, New York: Springer.

Bohnenberger, T., Jameson, A., Krüger, A., and Butz, A. (2002). User acceptance of a decision-theoretic, location-aware shopping guide. In Gil, Y. and Leake, D. B. (Eds.), *IUI 2002: International Conference on Intelligent User Interfaces*, pages 178–179. New York: ACM.

Borgman, C. (1999). The user's mental model of an information retrieval system: an experiment on a prototype online catalog. *International Journal of Human-Computer Studies*, *51*(2), 435–452.

Botafogo, R. A., Rivlin, E., and Shneiderman, B. (1992). Structural analysis of hypertexts: Identifying hierarchies and useful metrics. *ACM Transactions on Information Systems*, *10*(2), 142–180.

Bovair, S., Kieras, D. E., and Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human Computer Interaction*, *5*, 1–48.

Broadbent, D. E., Fitzgerald, P., and Broadbent, M. H. P. (1986). Implicit and explicit knowledge in the control of complex systems. *British Journal of Psychology*, *77*, 33–50.

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

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

Brusilovsky, P. and de Bra, P. (Eds.). (1999). *Proceedings of the Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*. available at http://wwwis.win.tue.nl/asum99/contents.html.

Brusilovsky, P. and Eklund, J. (1998). A study of user-model based link annotation in educational hypermedia. *Journal of Universal Computer Science, special issue on assessment issues for educational software*, *4*(4), 429–448.

Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. In *Computer Networks and ISDN Systems. Proceedings of the Seventh International World Wide Web Conference, 14-18 April 1998*, volume 30, pages 291–300.

Brusilovsky, P., Karagiannidis, C., and Sampson, D. G. (2001). The benefits of layered evaluation of adaptive applications and services. In Weibelzahl, S., Chin, D. N., and

Weber, G. (Eds.), *Empirical Evaluation of Adaptive Systems. Proceedings of workshop at the Eighth International Conference on User Modeling, UM2001*, pages 1–8, Freiburg.

Buckley, F. and Harary, F. (1990). *Distance in graphs*. Redwood City: Addison-Wesley.

Bueno, D. and David, A. A. (2001). METIORE: A personalized information retrieval system. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 168–177. Berlin: Springer.

Bull, S., Greer, J. E., McCalla, G. I., Kettel, L., and Bowes, J. (2001). User modelling in I-help: What, why, when and how. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 117–126. Berlin: Springer.

Bull, S. and Pain, H. (1995). "Did I say what I think I said, and do you agree with me?" Inspecting and questioning the student model. In Greer, J. E. (Ed.), *Artificial Intelligence in Education, Proceedings of AI-ED'95, Seventh World Conference on Artificial Intelligence in Education, 16-19 August 1995. Washington, DC, AACE*, pages 501–508.

Carro, R. M., Pulido, E., and Rodríguez, P. (2001). TANGOW: a model for internet based learning. *International Journal of Continuing Engineering Education and Life-Long Learning*, *11*(1-2).

Chin, D. N. (1989). KNOME: Modeling what the user knows in UC. In Kobsa, A. and Wahlster, W. (Eds.), *User Models in Dialog Systems*, pages 74–107. Berlin: Springer.

Chin, D. N. (2001). Empirical evaluation of user models and user-adapted systems. *User Modeling and User-Adapted Interaction*, *11*(1-2), 181–194.

Chin, D. N. and Porage, A. (2001). Acquiring user preferences for product customization. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 95–104. Berlin: Springer.

Chin, J. P., Diehl, V. A., and Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of ACM CHI'88 Conference on Human Factors in Computing*, pages 213–218.

Chiu, B. C., Webb, G. I., and Kuzmycz, M. (1997). A comparison of first-order and zeroth-order induction for Input-Output Agent Modelling. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 347–358. Vienna, New York: Springer. Available from http://um.org.

Chu-Carroll, J. and Brown, M. K. (1998). An evidential model for tracking initiative in collaborative dialogue interactions. *User Modeling and User-Adapted Interaction*, *8*(3-4), 215–253.

Cini, A. and de Lima, J. V. (2002). Adaptivity conditions evaluation for the user of hypermedia presentations built with AHA! In de Bra, P., Brusilovsky, P., and Conejo, R. (Eds.), *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 497–500. Berlin: Springer.

Cohen, W. W. (1996). Learning to classify English text with ILP methods. In Raedt, L. D. (Ed.), *Advances in Inductive Logic Programming*, pages 124–143. IOS Press.

Corbett, A. T. (2001). Cognitive computer tutors: Solving the two-sigma problem. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 137–147, Berlin. Springer.

Corbett, A. T. and Anderson, J. R. (1992). The LISP intelligent tutoring system: Research in skill acquisition. In Larkin, J., Chabay, R., and Scheftic, C. (Eds.), *Computer Assisted Instruction and Intelligent Tutoring Systems: Establishing Communication and Collaboration*, pages 73–110. Hillsdale: Erlbaum.

Corbett, A. T. and Bhatnagar, A. (1997). Student modeling in the ACT programming tutor: Adjusting a procedural learning model with declarative knowledge. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 243–254. Vienna, New York: Springer. Available from http://um.org.

Cronbach, L. J. (1967). How can instruction be adapted to individual differences. In Gagné, R. M. (Ed.), *Learning and individual differences*. Ohio: Columbus.

Crosby, M. E., Iding, M. K., and Chin, D. N. (2001). Visual search and background complexity: Does the forest hide the trees? In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 225–227. Berlin: Springer.

Curtis, B., Sheppard, S. B., Milliman, P., Borst, M. A., and Love, T. (1979). Measuring the Psychological Complexity of Software Maintenance Tasks with the Halstead and McCabe metrics. *IEEE Transactions on Software Engineering, 5*(2), 96–104.

de Bra, P. and Calvi, L. (1998). AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia, 4*, 115–139.

de Bra, P., Houben, G.-J., and Wu, H. (1999). AHAM: A Dexter-based reference model for adaptive hypermedia. In Tochtermann, K., Westbomke, J., Wiil, U. K., and Leggett, J. (Eds.), *Proceedings of the ACM Conference on Hypertext and Hypermedia*, pages 147–156, Darmstadt.

de Haan, G., van der Veer, G. C., and van Vliet, J. C. (1991). Formal modelling techniques in human-computer interaction. *Acta Psychologica, 78*(1-3), 26–76.

Dix, A. J., Finlay, J. E., Abowd, G. D., and Beale, R. (1998). *Human-Computer Interaction*. Harlow, England: Prentice Hall.

Draier, T. and Gallinari, P. (2001). Characterizing sequences of user actions for access log analysis. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 228–230. Berlin: Springer.

Draper, S. W., Brown, M. I., Henderson, F. P., and McAteer, E. (1996). Integrative evaluation: an emerging role for classroom studies of CAL. *Computers and Education, 26*(1-3), 17–32.

Edmonds, E. A. (1987). Adapation, response and knowledge. *Knowledge-Based Systems, 1*(1). Editorial.

Eklund, J. (1999). *A Study of Adaptive Link Annotation in Educational Hypermedia*. PhD thesis, University of Sydney.

Eklund, J. and Brusilovsky, P. (1998). The value of adaptivity in hypermedia learning environments: A short review of empirical evidence. In Brusilovsky, P. and de Bra, P. (Eds.), *Proceedings of Second Adaptive Hypertext and Hypermedia Workshop at the Ninth ACM International Hypertext Conference Hypertext'98, Pittsburgh, PA, June 20, 1998*, Computing Science Reports, Report No. 98/12, pages 13–19. Eindhoven: Eindhoven University of Technology.

Encarnação, L. M. and Stoev, S. L. (1999). Application-independent intelligent user support system exploiting action-sequence based user modeling. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 245–254. Vienna, New York: Springer.

Fischer, G. and Ye, Y. (2001). Personalizing delivered information in a software reuse environment. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 178–187. Berlin: Springer.

Goren-Bar, D., Kuflik, T., Lev, D., and Shova, P. (2001). Automating personal categorization using artificial neural networks. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 188–198. Berlin: Springer.

Grant, S. and Mayes, T. (1991). Cognitive Task Analysis? In Weir, G. R. S. and Alty, J. L. (Eds.), *Human-Computer Interaction and Complex Systems*, pages 147–167. San Diego: Academic Press.

Green, N. and Carberry, S. (1999). A computational mechanism for initiative in answer generation. *User Modeling and User-Adapted Interaction*, *9*(1-2), 93–132.

Greer, J. E., McCalla, G. I., Collins, J., Kumar, V. S., Meagher, P., and Vassileva, J. (1998). Supporting peer help and collaboration in distributed workplace environments. *International Journal of Artificial Intelligence in Education*, *9*, 159–177.

Guzmán, E. and Conejo, R. (2002). Simultaneous evaluation of multiple topics in SIETTE. In Cerri, S. A., Gouardères, G., and Paraguaçu, F. (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems, ITS 2002*, volume 2363 of *LNCS*, pages 739–748. Berlin: Springer.

Henze, N. and Nejdl, W. (1999). Adaptivity in the KBS Hyperbook System. In *2nd Workshop on User Modeling and Adaptive Systems on the WWW, May 11th, Toronto, Canada*. Held in conjunction with the World Wide Web Conference (WWW 8), and the Seventh International Conference on User Modeling (UM 99).

Henze, N., Nejdl, W., and Wolpers, M. (1999). Modeling constructivist teaching functionality and structure in the KBS hyperbook system. In *Proceedings of the Computer Supported Collaborative Learning Conference (CSCL'99)*, Stanford.

Herder, E. (2002). Structure-based adaptation to the mobile context. In Henze, N., Kókai, G., Schröder, O., and Zeidler, J. (Eds.), *Personalization for the Mobile World. Proceedings of the German Workshop on Adaptivity and User Modeling in interactive Systems, ABIS 2002*, pages 22–26, Hannover.

Höök, K. (2000). Steps to take before intelligent user interfaces become real. *Interacting With Computers*, *12*(4), 409–426.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI*, pages 256–265. San Francisco: Morgan Kaufmann.

Horvitz, E. and Paek, T. (2001). Harnessing models of user's goals to mediate clarification dialog in spoken language systems. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*. Berlin: Springer.

Howard, S. and Murray, M. D. (1987). A taxonomy of evaluation techniques for HCI. In Bullinger, H.-J. and Shackel, B. (Eds.), *Human-Computer Interaction Interact'87*, pages 453–459.

Hübscher, R. and Puntambekar, S. (2002). Adaptive navigation for learners in hypermedia is scaffolded navigation. In de Bra, P., Brusilovsky, P., and Conejo, R. (Eds.), *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 184–192. Berlin: Springer.

International Standards Organisation (1998). *ISO 9241. Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability* (1 ed.).

Jameson, A. (1996). Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction, 5*, 193–251.

Jameson, A. (1999). User-adaptive systems: An integrative overview. Tutorial presented at the Seventh International Conference on User Modeling, Banff, Canada, June 20th 1999.

Jameson, A. (2001). *Systems That Adapt to Their Users: An Integrative Perspective*. Saarbrücken: Saarland University.

Jameson, A., Paris, C., and Tasso, C. (Eds.). (1997). *User Modeling: Proceedings of the Sixth International Conference, UM97*. Vienna, New York: Springer.

Jameson, A. and Schwarzkopf, E. (2002). Pros and cons of controllability: An empirical study. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 193–202. Berlin: Springer.

Jörding, T. (1999). Temporary user modeling for adaptive product presentation in the web. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 333–334. Vienna, New York: Springer.

Juvina, I., Trausan-Matu, S., Iosif, G., van der Veer, G. C., Marhan, A.-M., and Chisalita, C. (2002). Analysis of web browsing behavior - a great potential for psychological research. In Pribeanu, C. and Vanderdonckt, J. (Eds.), *Task Models and Diagrams for User Interface Design: Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design - TAMODIA 2002, 18-19 July 2002, Bucharest, Romania*, pages 170–178. INFOREC Publishing House Bucharest.

Karagiannidis, C. and Sampson, D. G. (2000). Layered evaluation of adaptive applications and services. In Brusilovsky, P. and Stock, C. S. O. (Eds.), *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2000, Trento, Italy*, pages 343–346. Berlin: Springer.

Kay, J. (1995). The UM toolkit for cooperative user models. *User Models and User Adapted Interaction, 4*(3), 149–196.

Kay, J. (Ed.). (1999). *User Modeling: Proceedings of the Seventh International Conference, UM99*. Vienna, New York: Springer.

Kieras, D. E. and Polson, P. G. (1999). An approach to the formal analysis of user complexity. *International Journal of Human-Computer Studies*, *51*(2), 405–434.

Kim, S., Hall, W., and Keane, A. (2001). Using document structure for personal ontologies and user modeling. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 240–242. Berlin: Springer.

Klein, B. (2000). *Didaktisches Design hypermedialer Lernumgebungen: Die adaptive Lernumgebung "incops" zur Einführung in die Kognitionspsychologie*. Marburg: Tectum.

Koch, N. and Wirsing, M. (2002). The Munich Reference Model for adaptive hypermedia applications. In de Bra, P., Brusilovsky, P., and Conejo, R. (Eds.), *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 213–222. Berlin: Springer.

Krogsæter, M., Oppermann, R., and Thomas, C. G. (1994). A user interface integrating adaptability and adaptivity. In Oppermann, R. (Ed.), *Adaptive User Support*, pages 97–125. Hillsdale: Lawrence Erlbaum.

Lesh, N., Rich, C., and Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 23–32. Vienna, New York: Springer.

Linden, G., Hanks, S., and Lesh, N. (1997). Interactive assessment of user preference models: The Automated Travel Assistant. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 67–78. Vienna, New York: Springer. Available from http://um.org.

Lippitsch, S., Weibelzahl, S., and Weber, G. (2002). Improving structural salience by content adaptation with summaries. In Henze, N., Kókai, G., Schröder, O., and Zeidler, J. (Eds.), *Personalization for the Mobile World. Proceedings of the German Workshop on Adaptivity and User Modeling in Interactive Systems, ABIS02*, pages 35–38, Hannover.

Lippitsch, S., Weibelzahl, S., and Weber, G. (2003). Adaptive learning courses in pedagogical psychology. The PSI project and the authoring system NetCoach. *Künstliche Intelligenz, 3/03*.

Litman, D. and Pan, S. (1999). Empirically evaluating an adaptable spoken dialogue system. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 54–64. Vienna, New York: Springer.

Luckin, R. and du Boulay, B. (1999). Capability, potential and collaborative assistance. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 139–148. Vienna, New York: Springer.

Magnini, B. and Strapparava, C. (2001). Improving user modeling with content-based techniques. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 74–83. Berlin: Springer.

Marinilli, M., Micarelli, A., and Sciarrone, F. (1999). A case-based approach to adaptive information filtering for the WWW. In Brusilovsky, P. and De Bra, P. (Eds.), *Proceedings of the Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, pages 81–87. available at http://wwwis.win.tue.nl/asum99/.

Mark, M. A. and Greer, J. E. (1993). Evaluation methodologies for intelligent tutoring systems. *Journal of Artificial Intelligence and Education*, *4*(2/3), 129–153.

McCabe, T. (1976). A complexity measure. In *IEEE Transactions on Software Engineering, SE-2*, pages 308–320.

McDonald, S. and Stevenson, R. J. (1998). Effects of text structure and prior knowledge of the learner on navigation in hypertext. *Human Factors*, *40*(1), 18–27.

McGrath, J. E. (1995). Methodology matters: Doing research in the behavioral and social sciences. In Baecker, R. M., Grudin, J., Buxton, W. A. S., and Greenberg, S. (Eds.), *Readings in Human-Computer Interaction* (2 ed.)., pages 152–169. San Francisco, CA: Morgan Kaufmann.

Mitrovic, A. (2001). Investigating students' self-assessment skills. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 247–250. Berlin: Springer.

Müller, C., Großmann-Hutter, B., Jameson, A., Rummer, R., and Wittig, F. (2001). Recognizing time pressure and cognitive load on the basis of speech: An experimental study. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 24–33. Berlin: Springer. Available from http://dfki.de/∼jameson/abs/MuellerGJ+01.html.

Murray, T. (2000). Evaluating the need for intelligence in an adaptive hypermedia system. In Gauthier, G., Frasson, C., and VanLehn, K. (Eds.), *Intelligent Tutoring Systems, Proceedings of the 5th International Conference, ITS 2000, Montreal, Canada*, pages 373–382. Berlin: Springer.

Murray, T., Shen, T., Piemonte, J., Condit, C., and Thibedeau, J. (2000). Adaptivity in the MetaLinks hyper-book authoring framework. In *Workshop Proceedings of Adaptive and Intelligent Web-Based Education Systems workshop at ITS 2000*.

Nielsen, J. (1989). The matters that really matter for hypertext usability. In *Proceedings of the ACM Hypertext'89 Conference, Pittsburgh*, pages 239–248.

Nielsen, J. (1993). *Usability Engineering*. San Diego: Morgan Kaufmann.

Noh, S. and Gmytrasiewicz, P. J. (1997). Agent modeling in antiair defense: A case study. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 389–400. Vienna, New York: Springer. Available from http://um.org.

Norman, D. A. (1983). Some observations on mental models. In Gentner, D. and Stevens, A. L. (Eds.), *Mental Models*, pages 7–14. Hillsdale: Lawrence Erlbaum.

Ohene-Djan, J. (2002). Ownership transfer via personalisation as a value-adding strategy for web-based education. In Brusilovsky, P., Henze, N., and Millán, E. (Eds.), *Proceedings of the workshop on adaptive systems for web-based education, held at the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 27–41. Universidad de Málaga.

Oppermann, R. (1994). Adaptively supported adaptability. *International Journal of Human Computer Studies*, *40*(3), 455–472.

Paramythis, A., Totter, A., and Stephanidis, C. (2001). A modular approach to the evaluation of adaptive user interfaces. In Weibelzahl, S., Chin, D. N., and Weber, G. (Eds.), *Empirical Evaluation of Adaptive Systems. Proceedings of workshop at the Eighth International Conference on User Modeling, UM2001*, pages 9–24, Freiburg.

Paris, C., Wan, S., Wilkinson, R., and Wu, M. (2001). Generating personal travel guides - and who wants them? In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 151–153. Berlin: Springer.

Park, I. and Hannafin, M. J. (1993). Empirically-based guidelines for the design of interactive multimedia. *Educational Technology Research & Development*, *41*(3), 63–85.

Partl, H. (1999). *HTML-Einführung, Hypertext Markup Language die Sprache des World Wide Web*. ZID BOKU Wien.

Pitkow, J. E. and Pirolli, P. L. T. (1999). Mining longest repeated subsequences to predict world wide web surfing. In *Second USENIX Symposium on Internet Technologies and Systems*, pages 139–150, Berkeley. USENIX Association.

Pohl, W. (1997). LaboUr—machine learning for user modeling. In Smith, M. J., Salvendy, G., and Koubek, R. J. (Eds.), *Design of Computing systems: Social and Ergonomic Considerations. Proceedings of the Seventh International Conference on Human-Computer Interaction*, volume B, pages 27–30. Amsterdam: Elsevier.

Pohl, W. (1998). User-adapted interaction, user modeling, and machine learning. In Timm, U. J. and Rössel, M. (Eds.), *Proceedings of the Sixth German Workshop on Adaptivity and User Modeling in Interactive Systems, ABIS98*, Erlangen.

Rasmussen, J. (1997). Merging paradigms: Decision making, management, and cognitive control. In Flin, R., Salas, E., Strub, M. E., and Marting, L. (Eds.), *Decision making under stress: Emerging paradigms and applications*, pages 67–85. Aldershot: Ashgate.

Rauterberg, M. (1992). A method of a quantitative measurement of cognitive complexity. In van der Veer, G. C., Tauber, M., Bagnara, S., and Antalovits, M. (Eds.), *Human-Computer Interaction: Tasks and Organisation*, pages 295–307. Rom: CUD.

Rauterberg, M. and Fjeld, M. (1998). Task analysis in human-computer interaction — supporting action regulation theory by simulation. *Zeitschrift für Arbeitswissenschaft*, *52*(3), 152–161.

Reisner, P. (1984). Formal grammar as a tool for analyzing ease of use. In Thomas, J. C. and Schneider, M. L. (Eds.), *Human Factors in Computing Systems*, pages 53–78. Norwood: Ablex.

Rich, E. A. (1983). Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies*, *18*, 199–214.

Rossi, P. H. and Freeman, H. E. (1993). *Evaluation*. Beverly Hills: Sage.

Runkel, P. J. and McGrath, J. E. (1972). *Research on Human Behavior: A systematic guide to method*. New York: Holt, Rinehart & Winston.

Sanrach, C. and Grandbastien, M. (2000). ECSAIWeb: A web-based authoring system to create adaptive learning systems. In Brusilovsky, P., Stock, O., and Strapparava, C. (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems. International Conference, AH 2000*, pages 214–226. Berlin: Springer.

Schoppek, W. (2002). Examples, rules, and strategies in the control of dynamic systems. *Cognitive Science Quaterly*, *2*(1), 63–92.

Scriven, M. (1967). The methodology of evaluation. In Tyler, R. W., Gagné, R. M., and Scriven, M. (Eds.), *Perspectives of curriculum evaluation*, American Educational Research Association Monograph Series on Evaluation, pages 39–83. Chicago: Rand McNally.

Segal, R. B. and Kephart, J. O. (1999). Mailcat: An intelligent assistant for organizing e-mail. In *Proceedings of the Third International Conference on Autonomous Agents*, pages 276–282.

Semerano, G., Ferilli, S., Fanizzi, N., and Abbattista, F. (2001). Learning interaction models in a digital library service. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 44–53. Berlin: Springer.

Shin, E. C., Schallert, D. L., and Savenye, W. C. (1994). Effects of learner control, advisement, and prior knowledge on young students' learning in a hypertext environment. *Educational Technology Research & Development*, *42*(1), 33–46.

Sison, R. C., Numao, M., and Shimura, M. (1998). Discovering error classes from discrepancies in novice behaviors via multistrategy conceptual clustering. *User Modeling and User-Adapted Interaction*, *8*(1-2), 103–129.

Smyth, B. and Cotter, P. (2002). The plight of the navigator: solving the navigation problem for wireless portals. In de Bra, P., Brusilovsky, P., and Conejo, R. (Eds.), *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 328–337. Berlin: Springer.

Specht, M. (1998). Empirical evaluation of adaptive annotation in hypermedia. In Ottmann, T. and Tomek, I. (Eds.), *Proceedings of the 10th World Conference on Educational Telecommunications, ED-MEDIA & ED-Telecom '98, Freiburg, Germany*, pages 1327–1332, Charlottesville, VA. AACE.

Specht, M. and Kobsa, A. (1999). Interaction of domain expertise and interface design in adaptive educational hypermedia. In Brusilovsky, P. and De Bra, P. (Eds.), *Proceedings of the Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, pages 89–93. available at http://wwwis.win.tue.nl/asum99/.

Spooner, R. I. W. and Edwards, A. D. N. (1997). User modelling for error recovery: A spelling checker for dyslexic users. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 147–157. Vienna, New York: Springer. Available from http://um.org.

Strachan, L., Anderson, J., Sneesby, M., and Evans, M. (1997). Pragmatic user modelling in a commercial software system. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 189–200. Vienna, New York: Springer. Available from http://um.org.

Theo, G. (2001). Getting the right information to the right person. In Bauer, M., Vassileva, J., and Gmytrasiewicz, P. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 257–259. Berlin: Springer.

Thimbleby, A. (1990). *User Interface Design*. Workingham: Addison Wesley.

Totterdell, P. A. and Boyle, E. (1990). The evaluation of adaptive systems. In Browne, D., Totterdell, P., and Norman, M. (Eds.), *Adaptive User Interfaces*, pages 161–194. London: Academic Press.

Totterdell, P. A. and Rautenbach, P. (1990). Adaptation as a problem of design. In Browne, D., Totterdell, P., and Norman, M. (Eds.), *Adaptive User Interfaces*, pages 61–84. London: Academic Press.

Trewin, S. and Pain, H. (1997). Dynamic modelling of keyboard skills: Supporting users with motor disabilities. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 135–146. Vienna, New York: Springer. Available from http://um.org.

Villamañe, M., Gutiérrez, J., Arruabarrena, R., Pérez, T. A., López-Cudrado, J., Sanz, S., Sanz, S., and Vadillo, J. A. (2001). Use and evaluation of HEZINET; A system for basque language learning. In Lee, C. H., Lajoie, S., Mizoguchi, R., Yoo, Y. D., and du Boulay, B. (Eds.), *Proceedings of the Ninth International Conference on Computers in Education/ScoolNet, 2001 (ICCE-2001), Nov. 12-15, Seoul Korea.*, pages 93–101.

Virvou, M. and du Boulay, B. (1999). Human plausible reasoning for intelligent help. *User Modeling and User-Adapted Interaction*, *9*(4), 323–377.

Vogt, C. C., Cottrell, G. W., Belew, R. K., and Bartell, B. T. (1999). User lenses – achieving 100% precision on frequently asked questions. In Kay, J. (Ed.), *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 86–96. Vienna, New York: Springer.

Weber, G. (1999). Adaptive learning systems in the World Wide Web. In Kay, J. (Ed.), *User modeling: Proceedings of the Seventh International Conference, UM99*, pages 371–378. Vienna: Springer.

Weber, G., Kuhl, H.-C., and Weibelzahl, S. (2001). Developing adaptive internet based courses with the authoring system NetCoach. In Reich, S., Tzagarakis, M. M., and de Bra, P. (Eds.), *Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 226–238. Berlin: Springer.

Weber, G. and Möllenberg, A. (1995). ELM programming environment: A tutoring system for LISP beginners. In Wender, K. F., Schmalhofer, F., and Böcker, H.-D. (Eds.), *Cognition and computer programming*, pages 373–408. Norwood, NJ: Ablex Publishing Corporation.

Weber, G. and Specht, M. (1997a). User modeling and adaptive navigation support in WWW-based tutoring systems. In Jameson, A., Paris, C., and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 289–300. Vienna, New York: Springer. Available from http://um.org.

Weber, G. and Specht, M. (1997b). User modeling and adaptive navigation support in WWW-based tutoring systems. In Jameson, A. and Tasso, C. (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 289–300. Vienna: Springer.

Weibelzahl, S. (1999). Conception, implementation, and evaluation of a case based learning system for sales support in the internet. Master's thesis, University of Trier. available at http://home.ph-freiburg.de/weibelza/.

Weibelzahl, S. (2001). Evaluation of adaptive systems. In Bauer, M., Gmytrasiewicz, P. J., and Vassileva, J. (Eds.), *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 292–294. Berlin: Springer.

Weibelzahl, S. and Lauer, C. U. (2001). Framework for the evaluation of adaptive CBR-systems. In Vollrath, I., Schmitt, S., and Reimer, U. (Eds.), *Experience Management as Reuse of Knowledge. Proceedings of the Ninth German Workshop on Case Based Reasoning, GWCBR2001*, pages 254–263. Baden-Baden: Shaker.

Weibelzahl, S., Lippitsch, S., and Weber, G. (2002a). Advantages, opportunities, and limits of empirical evaluations: Evaluating adaptive systems. *Künstliche Intelligenz, 3/02*, 17–20.

Weibelzahl, S., Lippitsch, S., and Weber, G. (2002b). Supporting the authoring of Adaptive Hypermedia with structural information? In Henze, N., Kókai, G., Schröder, O., and Zeidler, J. (Eds.), *Personalization for the Mobile World. Proceedings of the German Workshop on Adaptivity and User Modeling in Interactive Systems, ABIS02*, pages 99–105, Hannover.

Weibelzahl, S. and Weber, G. (1999). Benutzermodellierung von Kundenwünschen durch Fallbasiertes Schließen. In Jörding, T. (Ed.), *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, ABIS-99*, Magdeburg. availabe at http://www-mmt.inf.tu-dresden.de/joerding/abis99/proceedings.html.

Weibelzahl, S. and Weber, G. (2000). Evaluation adaptiver Systeme und Verhaltenskomplexität. In Müller, M. E. (Ed.), *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, ABIS-2000*, Osnabrück.

Weibelzahl, S. and Weber, G. (2001a). A database of empirical evaluations of adaptive systems. In Klinkenberg, R., Rüping, S., Fick, A., Henze, N., Herzog, C., Molitor, R., and Schröder, O. (Eds.), *Proceedings of Workshop Lernen – Lehren – Wissen – Adaptivität*

(LLWA 01); research report in computer science nr. 763, pages 302–306. University of Dortmund.

Weibelzahl, S. and Weber, G. (2001b). Mental models for navigation in adaptive web-sites and behavioral complexity. In Arnold, T. and Herrmann, C. S. (Eds.), *Proceedings of the Fifth Annual Meeting of the German Cognitive Science Society, KogWis 2001*, pages 74–75. Leipzig: Leipziger Universitätsverlag.

Weibelzahl, S. and Weber, G. (2002). Adapting to prior knowledge of learners. In de Bra, P., Brusilovsky, P., and Conejo, R. (Eds.), *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Málaga, Spain, AH2002*, pages 448–451. Berlin: Springer.

Whitefield, A., Wilson, F., and Dowell, J. (1991). A framework for a human factors evaluation. *Behaviour & Information Technology*, *10*(1), 65–79.

Woods, D. D. (1993). The price of flexibility. In *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, pages 19–25, Orlando.

Worthen, B., Sanders, J., and Fitzpatrick, J. (1997). *Program Evaluation* (2nd ed.). New York: Longman.

Wu, H., Houben, G.-J., and de Bra, P. (1999). Authoring support for adaptive hypermedia applications. In *Proceedings of the Ed-Media 99 Conference*, pages 364–369.

Index

Author Index

Hiermit versichere ich, dass ich diese Dissertation selbständig verfaßt und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Stephan Weibelzahl