

Evaluating the Inference Mechanism of Adaptive Learning Systems

Stephan Weibelzahl and Gerhard Weber**

Pedagogical University Freiburg, Kunzenweg 21, 79117 Freiburg, Germany
[weibelza, webergeh]@ph-freiburg.de

Abstract The evaluation of user modeling systems is an important though often neglected area. Evaluating the inference of user properties can help to identify failures in the user model. In this paper we propose two methods to assess the accuracy of the user model. The assumptions about the user might either be compared to an external test, or might be used to predict the users' behavior. Two studies with five adaptive learning courses demonstrate the usefulness of the approach.

1 Evaluation of Adaptive Systems

Empirical evaluations of adaptive systems are rare [11]—e.g., only a quarter of the articles published in *User Modeling and User Adapted Interaction* report significant empirical evaluations. Many of them include a simple evaluation study with small sample sizes (often $N = 1$) and without any statistical method. Several reasons for this absence of significant studies have been identified (e.g., [6,7]). A systematic overview of current evaluation studies can be found in EASy-D, an online database of adaptive systems and related empirical studies¹.

Recently, we proposed an evaluation framework that supports and encourages researchers to evaluate their adaptive system by separating different evaluation steps [11]: evaluation of the input data, evaluation of the inference mechanism, evaluation of the adaptation decision, and evaluation of the interaction.

In this paper we focus on the second step. The inference mechanism is the crucial part of many adaptive systems. We propose two methods that test the accuracy of a user model

- by comparing its assumptions to an external test, and
- by comparing its assumptions to the actually displayed behavior of the learners.

These two methods are applied to an adaptive learning system, called the *HTML-Tutor*, to demonstrate the usefulness of the approach.

** © by Springer-Verlag

¹ <http://www.softwareevaluation.de>

2 Adaptive Learning Systems built with NetCoach

Adaptive learning systems adapt their behavior to individual learner properties such as the user's current knowledge. Opposed to static learning systems that present the same material to every user in the same way and order, adaptive systems consider individual differences in terms of knowledge, experience, preferences, or learning objectives [3] and thus promise to improve the teaching process [8].

NetCoach² is an authoring system that enables authors to develop adaptive web-based learning courses without being required to program source code [10].

2.1 Course Structure and Adaptivity

All NetCoach courses are based on the same structure. Similar to chapters and subchapters in a book, the learning material (i.e., pages with texts, images, animations) is stored in a hierarchical tree-structure of concepts. Learners may navigate through this structure freely. However, the course adapts to each learner individually by suggesting concepts that are suitable to work on next (adaptive curriculum sequencing) and by annotating the links to other concepts (adaptive link annotation).

This functionality is based on two kinds of data: concept relations and test sets (also called test groups) that check the learners knowledge about a concept. Authors may define two kinds of relations between concepts as regards contents. First, a concept might be prerequisite to another, i.e., this concept should be learned before the second concept is presented.

Second, a concept might infer another concept, i.e., the fact that the learner knows concept A implies that she also knows concept B.

The crucial part of NetCoach to assess the learner's knowledge are the so called test sets. A test set consists of a set of weighted test items that are related to a concept. There are forced choice, multiple choice, gap filling, and ranking items. All of them are evaluated online automatically. Users receive points for answering a test item correctly. Mistakes result in a reduction of points. Items are presented randomly (not yet presented items and incorrectly answered items are preferred) until the learner reaches a critical value. Only then the related concept is assumed to be learned completely.

2.2 Inference Mechanism

Adapting to the learner's current knowledge is one of the most important features of NetCoach. If the user completed a test set successfully (either during an introductory test before the content of the concept has been presented or afterwards in a post test) the concept is marked as solved and further inferences about the user's knowledge are drawn based on the inference relations between concepts.

NetCoach summarizes the learner's current knowledge by assigning one of five states to each concept. Table 1 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.

² <http://art.ph-freiburg.de>

Table 1. Possible states of a concept with a test set. 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 set 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.	orange ball with tick
known	the learner marked the concept as known without solving the test set	crossed orange ball

We argue that it is insufficient to assess the current knowledge by looking at the visited pages as most adaptive systems do (e.g., AHA [5] or Interbook [4]). An explicit assessment with sets of test items provides a much more reliable user model and might thus support better and adequate adaptations of the interface.

2.3 Adaptation Decision

Based on this inferred individual learning state NetCoach adapts its interface in two ways. First, links to other concepts are annotated with colored bullets that correspond to the learning state (adaptive link annotation). Table 1 gives an example of the default color configuration, however, authors are free to predefine other colors for each state.

Second, NetCoach suggests a concept that should be learned next and gives warnings if the learner visits a concept with the state *not ready* (adaptive curriculum sequencing).

Note, that for these adaptation techniques the quality of the learning state assessment is crucial. The adaptation decision will only work if the underlying assumptions about the learner are correct. Thus, the two methods to assess the accuracy of the user model are an important prerequisite to a successful adaptation.

3 Comparison of User Model to an External Test

The first proposed method to test the accuracy of a user model is to compare the assumptions in the user model to an external test. For instance, the assumption of a product recommendation system [2] about the most preferred product could be tested by actually letting the customer choose between several products. The user model of a system that adapts to the user’s keyboard skills [9] might be tested by assessing the skills

externally with a valid diagnostic instrument for motor disabilities. In an adaptive learning system it is possible to compare the assumed knowledge of the learner to the results of an external knowledge test that is known to have external validity. We evaluated the congruence of the user models in the *HTML-Tutor* and an extended external assessment. The *HTML-Tutor* is a NetCoach course that introduces to HTML and publishing on the internet. It consists of 138 concepts, 48 test sets, and 125 test items.

3.1 Method

We assessed 32 students who took part in one of three compact seminars between April 2001 and April 2002 at the Pedagogical University Freiburg. The 10 male and 22 female students had been studying for 0 to 9 semesters ($\bar{x} = 3.55$). 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. For instance, the line `` had to be changed to ``.

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. That is, in a qualitative analysis for each concept we decided whether the learner has complete knowledge about it. The test collects different data types (source code generation, open question, 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 evaluation 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.

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

3.2 Results

We found, that most assumptions were in congruence with test performance (see Table 2). 131 concepts were assumed to be either *solved* (i.e., the learner completed the test

Table 2. Number of congruent and incongruent cases. The results of 32 participants working on a subset of 48 concepts with test sets were observed. These $32 \times 48 = 1536$ cases are categorized in terms of the assumptions about the learners' knowledge in the user model of the *HTML-Tutor* and in terms of the results of the 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. The external test indicates whether a concept is *known* or *unknown*. Otherwise the case is categorized as *no information*.

		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

set 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 test (1244 out of 1536 cases) results from the fact that the time for such a test is limited. Compared to the 20 hours of teaching one hour of testing can obviously assess only a few selected aspects.

However, we identified nine incongruencies, i.e., there were nine cases where the system assumed that a concept is already *solved*, though the external assessment indicated that the concept is *unknown* (see Figure 1). These incongruencies were caused by three out of the 11 explored concepts. For all of these three concepts we were able to show that the test set did not measure the same kind of knowledge as the external test did. The nine cases are distributed across the concepts in the following way: in five cases the external test indicated that the learners do not encode German umlauts correctly (compared to 8 congruent cases in chapter 2.5). Nevertheless 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: three students were able to answer test items on the structure of HTML pages (chapter 1.6), but when they had to work on their own in the test, they just skipped the header of the page which resulted in an incorrect page. Nevertheless we identified 20 congruent cases for the same concept. Finally, the third concept (chapter 2.3) with an incongruence introduces line breaks. One student encoded them correctly when answering the test items, but sometimes forgot to do so when generating a page. 18 students behaved in a congruent way in regard to this concept.

To get a quantitative measure of how close the relation of user model and external assessment is, it is possible to compute a χ^2 -contingency-coefficient [1] based on the data given in table 2. A contingency measure of $C_{corr} = .24$ suggests that the two as-

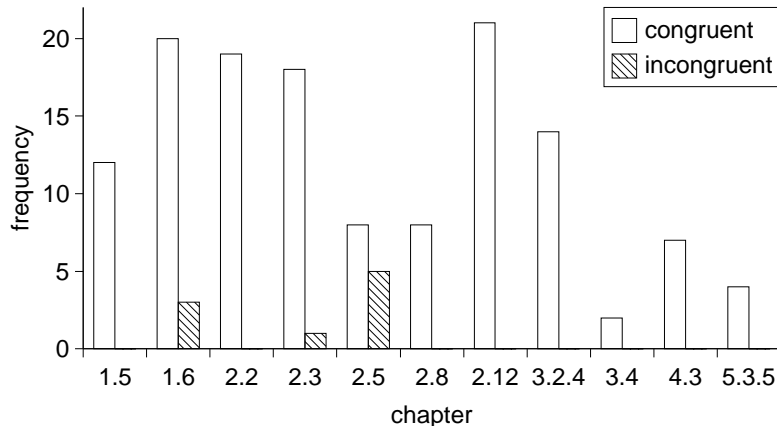


Figure 1. Number of congruent and incongruent cases categorized by chapters.

assessment methods are related, but do not measure the same. We propose 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.

4 Comparison of User Model to the User Behavior

In a second evaluation study we compared the assumptions in the user model to the actual displayed behavior of the learners. The user model of NetCoach courses contains information not only about known concepts but also whether the learner is *ready* or *not ready* to work on a concept (see Table 1). Users should perform worse on concepts that have the status *not ready* in the user model.

4.1 Method

We collected data from five online courses in different domains that had been developed with the NetCoach authoring system. These courses included the *HTML-Tutor* as well as four introductory courses on psychology such as problem solving (*Problemlösen*), 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 learner’s behavior and whether the learner was prepared. For each test set we specified the minimum number of items that were required to solve this set. The mean proportion of correct answers (\bar{c}) on these first items was computed for those who were

Table 3. 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, respectively the effect size (d) for significant tests

course	N_{users}	\bar{c}_{pre}	$\bar{c}_{\neg pre}$	N_{pre}	$N_{\neg pre}$	σ_{pre}	$\sigma_{\neg pre}$	α	$1 - \beta$	d
Kommunikation	172	.71	.63	1665	125	.43	.47	.04*		.18
P.-wahrnehmung	321	.69	.51	1629	132	.43	.46	.00*		.41
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*		.17
Σ	3501			11770	1183					

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*).

4.2 Results

Table 3 reveals 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. However, the statistical analysis makes obvious that the effect is not in the opposite direction. For *Problemloesen* the sample size was just too small (as indicated by the low test power).

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 sets. While some test sets are easy to solve and therefore the mean proportion of correct answers is high, other test sets are more difficult. However, not all users worked on every concept which decreases the sample size for subsequent conceptwise tests rapidly and statistical significance becomes difficult to reach. Nevertheless, a conceptwise comparison for the *HTML-Tutor* revealed that most of the results are conform with our hypothesis (see Table 4).

Table 4. 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 ($\bar{c}_{pre} > \bar{c}_{-pre}$). While most results were conform with this hypothesis only three of them were statistically significant

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

In summary, the study suggests that learners who do not fulfill all prerequisites for a concept perform worse than those who were prepared for the concept. 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. These results gain even more relevance if we consider the fact that 21,6% of the requested pages in the *HTML-Tutor* ($N = 40607$) are assumed to be *not ready*. As this study suggests that the learners will probably perform worse on a *not ready* page than on a *suggested* or *solved* one, it is reasonable to present a warning or to direct them to the missing prerequisite pages.

5 Conclusion

These two studies describe an approach for the evaluation of adaptive learning systems in general. Other adaptive systems might be evaluated in a similar way. For instance, there are validated tests and assessment methods for various psychological states, traits, preferences, and attitudes. For model dimensions that cannot be assessed in such a way it might be useful to observe whether the user’s behavior is predicted correctly. The approach is certainly limited to explicit user models. Implicit user models as they are sometimes used by machine learning systems are excluded. Moreover, the user model is very straightforward and of low complexity. Defining external tests for complex user models might require higher efforts than the knowledge test that was used in this study.

The results show that it is possible to evaluate the accuracy of the assumptions about the learner. Such evaluations might point to possible (and otherwise undiscoverable) improvements of the inference mechanism. However, a correct user model does not guarantee a successful adaptation, because the actual adaptation decision how to adapt might still be wrong,. e.g., the interface might become confusing by the chosen way of adaptation. Nevertheless, a correct user model is an important prerequisite for the adaptation success.

References

1. Alan Agresti. *An introduction to categorical data analysis*. Wiley, New York, 1996.
2. Liliana Ardissono and Anna Goy. Tailoring the interaction with users in electronic shops. In Judy Kay, editor, *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 35–44. Springer, Vienna, New York, 1999.
3. Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, 2001.
4. Peter Brusilovsky, John Eklund, and Elmar Schwarz. 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, 1998.
5. Paul de Bra and Licia Calvi. AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia*, 4:115–139, 1998.
6. John Eklund. *A Study of Adaptive Link Annotation in Educational Hypermedia*. PhD thesis, University of Sydney, 1999.
7. Kristina Höök. Steps to take before intelligent user interfaces become real. *Interacting With Computers*, 12(4):409–426, 2000.
8. Reinhard Oppermann, Rossen Rashev, and Kinshuk. Adaptability and adaptivity in learning systems. In A. Behrooz, editor, *Knowledge Transfer*, volume II, pages 173–179, 1997.
9. Shari Trewin and Helen Pain. Dynamic modelling of keyboard skills: Supporting users with motor disabilities. In Anthony Jameson, Cécile Paris, and Carlo Tasso, editors, *User Modeling: Proceedings of the Sixth International Conference, UM97*, pages 135–146. Springer, Vienna, New York, 1997. Available from <http://um.org>.
10. Gerhard Weber, Hans-Christian Kuhl, and Stephan Weibelzahl. Developing adaptive internet based courses with the authoring system NetCoach. In S. Reich, M. M. Tzagarakis, and Paul de Bra, editors, *Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 226–238. Springer, Berlin, 2001.
11. Stephan Weibelzahl. Evaluation of adaptive systems. In Mathias Bauer, Piotr J. Gmytrasiewicz, and Julita Vassileva, editors, *User Modeling: Proceedings of the Eighth International Conference, UM2001*, pages 292–294. Springer, Berlin, 2001.