

# Framework for the Evaluation of Adaptive CBR-Systems

S. Weibelzahl<sup>1</sup> and C. U. Lauer<sup>2</sup>

<sup>1</sup> Pedagogical University Freiburg, Germany  
weibelza@uni-freiburg.de

<sup>2</sup> Competence Center E-Business, University of Trier, Germany  
lauer@cogpsy.uni-trier.de

**Abstract.** Adaptivity is a way of increasing the usability of interactive software. Several systems use CBR as a basic inference mechanism to model the user and to reason about adaptive actions. However, empirical evaluations of adaptive systems are rare. This paper introduces an evaluation framework of six necessary steps for adaptive CBR-systems. The framework is applied to a case-based product recommendation system. Finally, possible experimental designs are presented. Evaluation criteria, both generally applicable as well as criteria that are specific to e-commerce applications, are discussed.

keywords: *evaluation framework, criteria, adaptivity, usability, behavioral complexity*

## 1 Problems and Barriers in the Evaluation of Adaptive CBR-Systems

Making interactive systems adaptive is an emerging field. Many systems have been developed that adapt to the user. E.g., an adaptive product retrieval system might adapt to the user's preferences, an adaptive learning environment might adapt to the learners current knowledge and goals, and an adaptive help system might adapt to the user's current task.

In some of these systems CBR is used as inference mechanism to provide the adaptive features. E.g., PTV (Smyth & Cotter, 1999) adaptively recommends TV programs, DUMBO (Melchior & Tarouco, 1999) supports network maintenance, ELM-PE (Weber, 1995) teaches programming by examples, and CASTLE (Weibelzahl, 1999) recommends vacation homes. These systems aim at optimizing the human-computer interaction. Adaptive features are one possibility of improving usability. However, empirical evaluations of adaptive systems are hard to find. Nevertheless, they are strictly required to justify the enormous efforts of implementation.

Several reasons have been identified to be responsible for this lack (e.g., Eklund, 1999). One structural reason is that computer science has little tradition of empirical research and, thus, evaluations of adaptive systems are usually often required for publication.

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 put to use for further development.

Third, adaptive systems have an inherent property which makes system comparisons difficult. We cannot simply switch off the adaptivity and make a non-adaptive system of it, because adaptivity is an essential part of the 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. Switching off the adaptivity in these systems might result in a rather useless product.

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 (Weibelzahl, Klein, & Weber, submitted; Karagiannidis & Sampson, 2000).

Based on these proposals this paper introduces a framework for the evaluation of adaptivity in case-based systems. Its structure is derived from a more general model for adaptive software (Weibelzahl et al., submitted). By applying this framework to a case-based product recommendation system, we will make clear which steps are required for a complete evaluation.

## 2 Evaluation Framework

The basic idea of our framework is to separate sequential information processing steps of the system. Each of these steps is validated on its own. We argue that only this procedure allows for possible improvements (formative evaluation) and at the same time for concrete measures of system performance (summative evaluation).

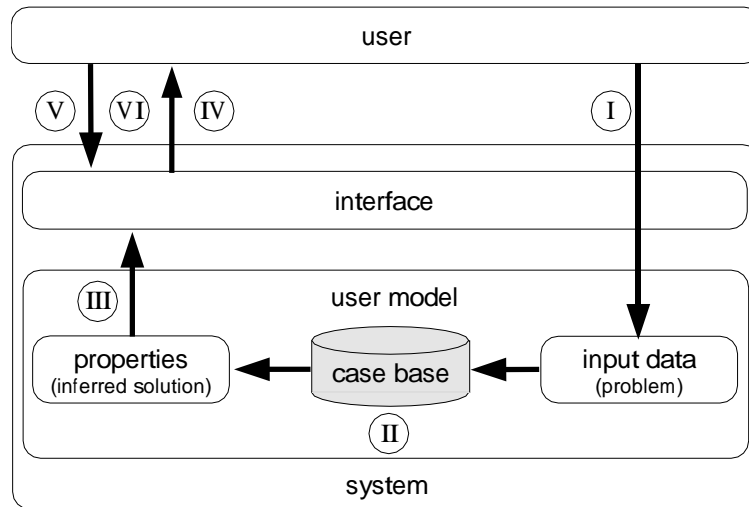
### 2.1 Scope of the Framework

By deriving the framework from a more general software evaluation model it is applicable to all adaptive systems that use CBR to reason about user properties. In our example below users are modeled as cases, with the preferences as problem part and the preferred product as solution part. Users with similar preferences receive similar product recommendations. However, adaptivity can be achieved by using other information containers, e.g., the adaptation, the similarity measures, or the domain model (e.g., Göker & Thompson, 2000). In principle, the proposed framework is applicable to these systems as well, but in this paper we will only refer to systems which infer user properties by CBR. There is no limitation to any domain or application type.

The framework aims at evaluating the adaptive features with its underlying CBR processes. Other software evaluation issues such as the system's precision or a verification of the algorithm are excluded.

### 2.2 Model of Adaptivity

**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 that they acquire additional input data. E.g., a system might monitor the interaction by registering the frequency of certain commands. Another more straightforward approach is to ask the user about preferences or interests.



**Fig. 1.** Architecture of adaptive CBR-systems and flow of information.

**Case-Based Retrieval.** In CBR-systems this additional input data is handled as problem or query of a case-based retrieval. Note, that the retrieval might involve several steps or inference processes, however, the basic idea of inferring solutions based on cases will always remain the same.

**User Model.** The outcome of this case-based reasoning process is a set of user properties on an abstract level. Thus, this step is sometimes called upwards inference (Jame-son, 1999). E.g., a system might infer the user’s knowledge about a concept, a preference for a specific product, or certain disabilities. The key point is that these properties are inferred from the input data and not accessed directly.

**Adaptation Decision.** The user properties are now used to adapt the interface. Note, that the used definition of “interface” is not limited to design issues, but also includes presentation strategies, contents, annotations, etc.

E.g., in most cases the appropriate way of adaptation seems to be obvious: 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 some time.

In summary, in the process of adaptation decision the system decides about concrete adaptation steps based on the inferred abstract user properties. A flowchart of the complete model and its information processing steps is shown in figure 1.

### 2.3 Evaluation Steps

Based on the described model of adaptivity we constructed a framework for evaluation. Obviously every information processing step of the model has to be evaluated on its own, before fitting it in a global evaluation process.

The following sections explain, how to evaluate the different steps. Each step is illustrated by the evaluation of a product recommendation system, called CASTLE (Weibelzahl & Weber, 1999). The roman numerals indicate which step of information processing in the model is referred to.

- I. *Correctness of input data acquisition*: To build a user model the system acquires direct or indirect input from the customer (e.g., appearance of specific behavior, utterances, answers, etc.). Is this data received in a reliable and valid way? Is there noise in the data? Does the input answer the complete problem, i.e., is it enough data for further inferences? Empirical test theory provides plenty of relevant interference factors (e.g., social desirability, reactance).

When recommending vacation homes, CASTLE asks the customer about general preferences and needs. We need to know whether this input data is reliable and valid. Reliability is determined by how stable an instrument measures each time it is used under the same conditions with the same subjects. Measuring the same concept twice after a while estimates the reliability of such a questionnaire (retest reliability), because preferences should remain stable, and so should the measurements.

To test the validity, we can compare two or more instruments. E.g., in addition to the questionnaire we could apply a selection task, where subjects have to choose between several objects. Only if other instruments confirm the results we can assume that the input data measures what we expect it measures.

- II. *Correctness of inference*: Based on the input, properties of the user are inferred. Similar to the first step we can check whether these properties are inferred in a reliable and valid way. This can be done by comparing assumptions about the user with reality by assessing the properties in an external way in addition or by asking the user directly whether the inferences are correct. That is, we evaluate the correctness of the property without looking at the usefulness.

E.g., CASTLE infers preferences for specific product configurations. In a simple laboratory experiment subjects might express this preference in an indirect way by choosing between different product configurations. Only if subjects actually select the predicted configuration the case-based inference is regarded as valid. Otherwise the reasoning process has to be improved, e.g., by changing similarity measures, maintaining the case base or changing indexes.

- III. *Appropriateness of adaptation decisions*: During 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.

Given that the properties are correct (as evaluated in the previous step), are the adaptation processes permissible, necessary, and sufficient to solve the user's problem or to assist the user? Considering the fact that usually several other decisions are possible, is the chosen decision the optimal one?

E.g., CASTLE uses the knowledge about product preferences to recommend products with the same (if available) or similar configurations. However, as mentioned above it would also be possible to inform the customer that a recommendation is available, but he/she may continue to browse the non-adaptive catalog. To decide which adaptation decision is optimal requires the definition of criteria. See section 3 for a discussion of adequate approaches.

- IV. *Change of system behavior when the system adapts*: Adaptation certainly has an impact on system behavior. In which way does system behavior change in comparison to the normal division of labor? If system behavior remains the same when comparing different users, the adaptation technique is probably not optimal, because a non-adaptive system could provide the same. If two property sets result in the same adaptation it is not necessary to discern these sets.

This evaluation step might also uncover problems that are related to the frequency of user models. Inferred properties that are theoretically correct but never occur in real interactions are not useful.

E.g., CASTLE recommends a list of appropriate products. Observing the system in interaction with real customers might identify an invariance of recommendation: if customers do not differ as much as expected this might result in the situation that the same product is recommended to all customers. A non-adaptive system would have achieved the same effect with less efforts.

- V. *Change in user behavior when system adapts*: Does the user change his/her behavior when the system adapts in comparison to the normal division of labor? In which way? Behavior changes provide valuable hints at changes in cognitive states, and thus at the system's usability.

E.g., in CASTLE, we compared the behavioral complexity of customers under two conditions: an adaptive and a non-adaptive version of CASTLE (Weibelzahl & Weber, 2000). Section 3.1 gives details of this evaluation approach.

- VI. *Change and quality of total interaction*: The main question is to the usability. How is the interaction quality? Does it change? Is the user satisfied? The point is that this last evaluation step can only be interpreted correctly if all the previous steps have been completed. Especially in the case of finding no difference between an adaptive and a non-adaptive system the previous steps provide hints at shortcomings.

In CASTLE a variety of methods was applied to evaluate the system's usability and customer satisfaction, including laboratory and field studies, with both control group and repeated measurement designs (Weibelzahl, Bergmann, & Weber, 2000; Lauer, 2000). See section 3.2 for a description of these evaluations.

## 2.4 Total Evaluation Procedure

The evaluation steps are interdependent, because the evaluation of a step requires a positive evaluation in the previous step. E.g., an evaluation might find that the inferred user properties are not in accordance with reality. This result can be interpreted in two ways. Either the case-based retrieval failed or the query was incorrect. But by evaluating each step on its own, our framework can clear this ambiguity. Fitting this in the example from before, this means, if an independent evaluation proves the quality of the query,

the missing accordance of the inferred user properties is caused by the case-based retrieval.

In some systems one of the steps is redundant. E.g., a system might acquire the number of mouse-clicks as input. This data is probably perfectly reliable and valid. In this case an empirical evaluation of this step is not required.

In summary, while step I through V are part of a formative evaluation that provides hints for shortcomings, step VI represents a global summative evaluation of the complete system.

### 3 Current Criteria

The previous section introduced a model of adaptivity and according evaluation steps. For the steps III to VI we only outlined *what* has to be evaluated, but not *how*. Thus, in this sections we describe and discuss criteria and experimental designs used for the evaluation of CASTLE and for the evaluation of similar systems.

The criteria can be classified in two categories. One category are *general criteria*, which are applicable to (almost) all adaptive systems, no matter of the domain or the underlying inference mechanism. For this category we will introduce behavioral complexity as an example. The other category of criteria is specific to product recommendation and e-commerce.

#### 3.1 General Criteria

Current evaluations of adaptive systems apply a variety of criteria for the usability of adaptive systems, including traditional criteria from human-computer interaction research such as usability questionnaires and newly developed criteria such as behavioral complexity.

**Traditional Usability Criteria.** Most evaluations apply objective criteria, e.g., duration of interaction, number of navigation or dialogue steps, number of errors, or frequency of assistance access. These measures are easy to access and do not distort the user behavior. However, most objective criteria are difficult to interpret. E.g., a reduction of interaction duration might be caused by either a more easy to handle interface or by annoyed users who tried to minimize the interaction as much as possible.

A correct interpretation often requires the consideration of additional subjective criteria, such as the user's preference for one of two versions, or a standardized usability questionnaire. When referring to these subjective (cognitive) states the picture of the interaction between the user and the adaptive system becomes much clearer.

Thus, the evaluation of CASTLE considered both objective and subjective criteria. Customers who interacted with the adaptive version of CASTLE needed less time to find a suitable vacation home and were more satisfied (as indicated by a questionnaire) than those who used a non-adaptive version.

Yet, a methodological problem arose in this and several other empirical evaluations: Adaptivity effects are often small, compared to the huge variance that is generated by

the individual differences. Even with statistical methods it is difficult to extract the effects that are caused by adaptivity from the enormous background noise.

Faced with this problem when evaluating CASTLE we developed two approaches. First, a different experimental design and criteria that are more specific to the domain yielded interesting results (see section 3.2). Second, we derived a new criterion from theoretical considerations, called behavioral complexity. This measure seems to be more sensitive to adaptivity effects than the criteria described above.

**Behavioral Complexity as Criterion.** Adaptive systems change the division of labor (Jameson, 1999). They take over routine tasks or perform actions that have not been initiated explicitly—actions, to smooth and improve the interaction. The adaptive behavior reduces interaction complexity. This makes it easier for the users to reach their goals, which are either given or set themselves, because less knowledge about the system is required.

The interaction process can be seen as a state-transition network. The system changes its current state when the user initiates an action. E.g., 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.

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 & Fjeld, 1998). They will return to a previous state if they realize that the chosen transition did not result in the effect they wanted.

By exploring the system the users can increase their knowledge about the system's functionality, 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.

In a laboratory experiment (Weibelzahl & Weber, 2000), CASTLE was tested to reduce the users' behavioral complexity. This result is even more important if we take into consideration that several traditional criteria did not indicate a difference between the adaptive and the non-adaptive version of CASTLE.

### 3.2 Criteria for the evaluation of product recommendation systems

Adaptive systems in e-commerce are designed to improve the customer satisfaction, i.e., the satisfaction with the service should increase and the system should fulfill the customers' expectations (Liljander & Strandvik, 1993; Groß-Engelmann, 1999).

Several measurement methods have been proposed (Homburg & Werner, 1998). The easiest way is to ask the customer about service quality. E.g., customers might express their subjective impressions in an interview or a questionnaire about the support they received by the system, whether their wishes have been considered by the system, or the service in general.

In addition to this service specific questions, it is also possible to ask about the satisfaction in general. A typical rating scale, that was also used in the CASTLE evaluation, is: “All in all—when I think of the system—I am ... absolutely satisfied ... absolutely dissatisfied”; the customer may rate between 1 (satisfied) and 9 (dissatisfied).

Customers who are satisfied probably intend to use the system again. Thus, an interesting question refers to this intention of future behavior.

Finally, customers might compare the “virtual” interaction with a real sales talk. Users might perceive the system as either too offensive (e.g., very many recommendations) or passive (e.g., few recommendations, no reasons for recommendation given).

The evaluation of CASTLE considered all of the above (Lauer, 2000). In a repeated measurement design people interacted with CASTLE and a non-adaptive version of CASTLE in a random order. After an exploration phase for each version they had to rate the system in reference to several aspects, including their overall satisfaction with the system, the amount of efforts needed to find a suitable product, the system’s usability and appearance, the service quality, their intention to use the system again, and their satisfaction in comparison to a real sales talk.

We found that the overall satisfaction, as well as the satisfaction with the service and the navigational features improved considerably, while the perceived efforts of interaction were reduced. The empirical design reduced error variance which in turn alleviated statistical inferences.

## 4 Conclusion

The necessary empirical evaluation of adaptive software has been widely neglected over a long period. We claim that the described framework can alleviate some of the problems of classical evaluation procedures. The goal of our framework is to help researchers to avoid pitfalls and problems concerning the evaluation of adaptivity in CBR systems. Evaluations containing all steps of our framework will allow to emphasize the advantages of adaptivity and to point out possible improvement.

## References

- 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.
- Eklund, J. (1999). *A study of adaptive link annotation in educational hypermedia*. PhD thesis, University of Sydney.
- Göker, M., & Thompson, C. (2000). Personalized conversational case-based recommendation. In E. Blanzieri & L. Portinale (Eds.), *Advances in Case-Based Reasoning. Proceedings of the 5th European Workshop on Case Based Reasoning*. Berlin: Springer.
- Groß-Engelmann, M. (1999). *Kundenzufriedenheit als psychologisches Konstrukt: Bestandsaufnahme und emotionstheoretische Erweiterung bestehender Erklärungs- und Meßmodelle*. Lohmar: Eul.

- Homburg, C., & Werner, H. (1998). Messung und Management von Kundenzufriedenheit. *Marktforschung und Management*, 42(4), 131–135.
- Höök, K. (2000). Steps to take before intelligent user interfaces become real. *Interacting With Computers*, 12(4), 409–426.
- Jameson, A. (1999). *User-adaptive systems: An integrative overview*. (Tutorial presented at the Seventh International Conference on User Modeling, Banff, Canada, June 20th 1999)
- Karagiannidis, C., & Sampson, D. G. (2000). Layered evaluation of adaptive applications and services. In P. Brusilovsky & C. S. Oliviero Stock (Eds.), *Proceedings on International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2000, Trento, Italy* (p. 343-346). Berlin: Springer.
- Lauer, C. U. (2000). *Fallbasierte Verkaufsunterstützung im Internet: Adaption und Evaluation des CASTLE-Systems*. Unpublished master's thesis, University of Trier. (available at <http://psychologie.uni-trier.de:8000/people/ulauer/diploma.pdf>)
- Liljander, V., & Strandvik, T. (1993). Estimating zones of tolerance in perceived service quality and perceived service value. *International Journal of Service Industry Management*, 4(2), 6–28.
- Melchior, C., & Tarouco, L. M. (1999). Fault management in computer networks using case-based reasoning: DUMBO system. In K.-D. Althoff, R. Berkman, & K. Branting (Eds.), *Case-based Reasoning Research and Development, Proceedings of the Third International Conference on Case-Based Reasoning ICCBR99* (pp. 510–524). Berlin: Springer.
- Rauterberg, M., & Fjeld, M. (1998). Task analysis in human-computer interaction — supporting action regulation theory by simulation. *Zeitschrift für Arbeitswissenschaft*(3), 152.
- Smyth, B., & Cotter, P. (1999). Surfing the digital wave — generating personalized TV listings using collaborative, case-based recommendation. In K.-D. Althoff, R. Berkman, & K. Branting (Eds.), *Case-based Reasoning Research and Development, Proceedings of the Third International Conference on Case-Based Reasoning ICCBR99* (pp. 561–571). Berlin: Springer.
- Weber, G. (1995). Examples and reminders in a case-based help system. In J.-P. Haton, M. Keane, & M. Manago (Eds.), *Advances in case-based reasoning. Selected Papers of the Second European Workshop, EWCBR-94, Chantilly, France* (pp. 165–177). Berlin: Springer.
- Weibelzahl, S. (1999). *Conception, implementation, and evaluation of a case based learning system for sales support in the internet*. Unpublished master's thesis, University of Trier. (available at <http://www.iig.uni-freiburg.de/cognition/members/stephan/literatur/weibelzahl.pdf>)
- Weibelzahl, S., Bergmann, R., & Weber, G. (2000). Towards an empirical evaluation of CBR approaches to product recommendation - in electronic shops. In M. Göker (Ed.), *Proceedings of the 8th German Workshop on Case Based Reasoning, GWCBR2000* (pp. 3–12). Lämmerbüchel. (available at <http://www.wagr.informatik.uni-kl.de/~gwcb2k/program.html>)
- Weibelzahl, S., Klein, B., & Weber, G. (submitted). Criteria for the evaluation of adaptive systems. *User Modeling and User Adapted Interaction*.

- Weibelzahl, S., & Weber, G. (1999). Benutzermodellierung von Kundenwünschen durch Fallbasiertes Schließen. In T. Jörding (Ed.), *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, ABIS-99*. Magdeburg. (available at <http://www-mmt.inf.tu-dresden.de/joerding/abis99/proceedings.html>)
- Weibelzahl, S., & Weber, G. (2000). Evaluation adaptiver Systeme und Verhaltenskomplexität. In M. E. Müller (Ed.), *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, ABIS-2000*. Osnabrück.