Is the ACT-value a valid estimate for knowledge? An empirical evaluation of the inference mechanism of an adaptive help system

Dorothea Iglezakis

Cath. University Eichstaett-Ingolstadt,
Department for Applied Computer Science,
Ostenstr. 14, 85072 Eichstaett, Germany,
dorothea.iglezakis@ku-eichstaett.de

Abstract. This paper reports the results of an empirical study that evaluates the inference mechanism of an adaptive help system for web-based applications. The help system adapts to a measure for the procedural knowledge that is computed from activity logfiles according to the ACT-theory of Anderson and Lebière [1]. The results of the study show that the ACT-value of procedural knowledge correlates with subjective and objective measures of performance and proves itself as a better estimate of the procedural knowledge than general computer knowledge, a measure often used by other adaptive help systems.

1 Introduction

Help systems are not always as helpful as they should be. In a survey, users of a newly introduced software rated human helpers much higher than the online help, but asked for a better help function [7]. The help of SPSS even worsens the performance of the users [6]. On the other hand, a good online help is an important source of contentedness of customers. Adaptive help systems are one of many approaches to enhance the quality and the helpfulness of help systems. Existing systems either try to adapt the contents of the help by adapting to the knowledge of the user or try to reduce the information by adapting to the goals and plans of the user [5]. The used models of users knowledge are either global stereotypes or overlay models. Global stereotypes are only capable of domains, where the concepts are clearly ordered by difficulty. Overlay models assign a value or a probability to each concept, how well or how probable a user knows this concept. The overlay models entail much more information than the global stereotype models, but also use static knowledge. If a user proves to know a concept, the model assumes this knowledge until the user proves the opposite. None of the adaptive help systems use forgetting in their user models.

Cognitive psychology offers a lot of research about knowledge and forgetting. The adaptive help system evaluated in this paper tries to use the results of this research to offer an alternative way to model the knowledge of the user.

2 CHEetah—An Adaptive Help System for Web-Based Applications

CHEetah is an adaptive help system for web-based applications that adapts to the procedural knowledge of the user. The user model is a long term model that contains a measure of the procedural knowledge of the user for every functionality in the target application. The adaptation happens through showing and hiding different parts of the help items.

2.1 Adaptation target

CHEetah adapts to the procedural knowledge of the user. To model the knowledge of the user as accurately as possible, results from cognitive psychology are used. The ACT-theory from Anderson and Lebière [1] delivers an empirically founded theory[1] about the learning and the forgetting of memory contents. The activation of a memory item corresponds to the accessibility of a concept and is influenced by the frequency of access. Each access of a memory item increases the activation of the corresponding memory trace. Without access, the activation fades over time.

Knowledge items differ in the type of knowledge. Procedural knowledge covers knowledge about actions and activities, whereas declarative knowledge covers knowledge about facts and world knowledge. Especially for help systems that have to explain and to support the execution of tasks, procedural knowledge is more central than declarative knowledge.

A knowledge item in the context of CHEetah is the procedural knowledge about one functionality of the target application. For example, if the target application is the configuration menu of an Internet provider, one knowledge item would be the knowledge about adding a new e-mail address. Declarative knowledge items can function as prerequisites or additional knowledge to the procedural knowledge items, but the procedural knowledge items are the focus of adaptation.

2.2 User Model

The ACT-theory incorporates the strength accumulation equation that computes the activity of a memory item in dependence on the time stamps of the last contacts with this memory item. A memory item in the context of CHEetah is the procedural knowledge about one functionality of the target application. The user model entails a ACT-measure of every functionality of the system for every user. If a user has never used a functionality, the corresponding ACT-value is 0. Otherwise, the ACT-value is computed according to the following equations.

$$a_i^z(T) = \sum_{j=1}^{n} t_j^{-d} \tag{1}$$

[1] See http://act-r.psy.cmu.edu/publications/ for a list of publications about the ACT-theory and evaluations with the ACT-theory

The strength accumulation equation (1) computes the activation a_i of a knowledge trace at the time stamp z in the following way. T is a set of time stamps that contains all contacts with a functionality i. t_j is the time in minutes that passed since the j^{th} contact with the functionality i. The parameter d can be chosen from the interval $]0;1[$ and is dependent on the application. From literature, a $d = 0.5$ is a good estimate for many applications.

If the user model already specifies an ACT-value from a computation on a time stamp z_0 in the past, the value is updated at the time stamp z according to the following equation 2:

$$a_i^z(a_i^{z_0}, T) = a_i^{z_0} * (z - z_0)^{-d} + \sum_{j=1}^{n} t_j^{-d} \tag{2}$$

$a_i^{z_o}$ is the old value, T again the set of contacts with the functionality i between the old time stamp z_0 and the actual time stamp z.

The strength accumulation equation builds on the general activity function of the ACT-theory that we used in [5], but omits the logarithm. Therefore, the strength accumulation equation avoids the problem of negative ACT-values for very old experiences.

2.3 Input Data

The ACT-theory needs the time stamps of every contact with a memory item—in our context a functionality of the target application—to compute the ACT value. One contact with a functionality is a complete execution of this functionality. As CHEetah concentrates on web-based applications, the input data of CHEetah are logfiles of the web server on which the target application runs. We use an extended logfile format in XML that saves for every visited page the user-ID, the page name and the used parameters.

From these data, CHEetah recognizes the execution of functionalities through predefined processes. Each process specifies one possible execution of a functionality in form of a regular expression. The output of the process recognition is a process ID, a start time-stamp and an end time-stamp of every successful execution of a functionality of the target application. A contact with the functionality in the sense of the ACT-theory is therefore the end time-stamp of a successful execution of a corresponding process.

2.4 Adaptation

The adaptation of CHEetah happens through online assembling of different components of help items. This form of adaptation has proven successful in other adaptive help systems such as PUSH [4] and EPIAIM [2].

The main advantage lies in the simplification of writing and maintaining the help items. As Höök [4] states, it is very difficult to write and to maintain different versions of the same help items. The online assembling of the help items from standardized components like EPIAIM or PUSH is therefore more promising.

The help items of CHEetah are written in an XML format that bases on the Docbook [8] format. Each help item corresponds to one functionality of the target application. The adaptation happens through online assembling of the different components to one help item, dependent of the actual activation of the target process and its subordinate concepts and processes.

2.5 Summary

CHEetah has a concept of knowledge that is oriented on the results of cognitive psychology. The most important difference to other adaptive help systems is the fact, that CHEetah incorporates forgetting. If a user does not use a concept for a longer time, the knowledge about this concept is no longer available but nevertheless not vanished. The key concept of CHEetah is to show the user only these parts of the help information that are most valuable for him at this specific moment. This knowledge concept does not only allow to help novices, but also differentiates between different grades of expertise.A user that has forgotten the use of a functionality needs a different kind of help than a user that has never used this function before.

3 Empirical study

Weibelzahl [9] developed a framework for evaluations of adaptive systems that recommends evaluations on every step of the system:

- Evaluation of input data
- Evaluation of the inference mechanism
- Evaluation of adaptation decisions
- Evaluation of total interaction

In my opinion, an evaluation of the input data mechanism is not necessary, because there is no inference or uncertainty in this step. CHEetah uses the logfiles of the webserver that protocols the page visits of every user.

In contrast, the computation of the ACT-values as a measure of procedural knowledge needs evaluation. Therefore, we conducted an empirical study to answer the following questions:

- Is the ACT-value a reliable and valid measure of procedural knowledge?
- Does the ACT-value correspond with subjective and objective measures of the knowledge of the user?
- Is the ACT-value a better estimate of procedural knowledge than general computer experience?

To answer these questions, a web-based questionnaire was conducted. Participants of the study were 16 employees of the hmd software-AG, 6 male and 10 female. The target application for these evaluation was WebTime, a web-based calendar and task organization tool. The usage of WebTime in the hmd software-AG was logged over the period of 11 months. From these logfiles, the ACT-value of two different functionalities from the target application was computed. The functionality NewAppointment is a very basic task used very often by all participants. The functionality NewTodo was used more rarely and by a lower number of participants.

3.1 Design

The study wants to show if the ACT-value is a valid measure of the procedural knowledge of the user. The procedural knowledge according to one functionality is operationalized through subjective and objective measures of performance of this functionality.

Variables Therefore, the dependent variables were the subjective and the objective measure of the performance of the participants in the two different functionalities of the target application. Independent variables were the ACT-values of the participants of the two functionalities and—as a control variable— the general computer expertise of the participants.

Operationalization For the subjective performance measures the participants rated their expertise of the two functionalities on a six-step rating scale.

For the objective measurements, the participants were asked to perform the two functionalities on two examples. The performance of the users was logged and afterward rated on correctness and time. The correctness was rated on the following four-step scale:

- The task was successfully executed
- The task was executed with errors
- The task was started, but not finished
- The task was not started

Time was measured through the difference between the first step of the functionality and the final step of the functionality.

Procedural knowledge was represented by the ACT-values, computed from the log-files of the last eleven months according to equation 1.

Measure	*M*	*SD*	Min	Max
subj. performance NA	3.23	1.54	0	5
obj. performance NA	3.00	0.61	1	4
ACT value task NA	0.89	1.19	0.02	6.12
subj. performance NT	2.20	1.62	0	4
obj. performance NT	2.89	0.80	1	4
ACT value NT	0.02	0.04	0	0.17
$comp_h$	0.51	1.02	−1.25	3
$comp_t$	4.26	1.91	1	6

Table 1. Mean values and standard deviation for performance measures and procedural knowledge for the tasks NewAppointment (NA) and NewTodo (NT) and computer experience

General computer experience was rated through two different measures, a scale for helplessness with the computer ($comp_h$) and a guttmann scale [3] for activities ($comp_t$)[2].

Table 1 shows the mean values and standard deviation for the different variables.

3.2 Results

To get the relationship between the dependent and independent variables, correlations between subjective and objective measures of performance and the ACT-values and the correlations between the performance measures and general computer expertise were computed. Tables 2 and 3 show the results.

Correlation between	**r**	**Significance**
ACT — subjective expertise	0.19	n.s.
ACT — objective performance	0.09	n.s.
ACT — time	−0.06	n.s.
$Comp_h$ — subjective expertise	−0.32	n.s.
$Comp_h$ — objective performance	−0.59	$p < 0.01$
$Comp_h$ — time	−0.27	n.s.
$Comp_t$ — subjective expertise	−0.27	n.s.
$Comp_t$ — objective performance	−0.48	$p < 0.05$
$Comp_t$ — time	−0.27	n.s.

Table 2. Results of functionality NewAppointment

For the functionality NewAppointment, none of the correlations between ACT-value and performance measures is significant. The correlation between objective performance and ACT-value is near zero, but the correlation between subjective expertise and ACT-value shows an expected trend, though not very high. The correlations between general computer expertise and performance measures are higher, but in an unexpected direction. The correlation tendentially shows that the higher the general computer expertise, the lower is the subjective and objective performance.

For the functionality NewTodo, the results are more as expected. The correlation between subjective performance and ACT-value is significant ($p < 0.05$) and with a value of $r = .66$ relatively high. So the higher the ACT-value, the better the participant rates his expertise in the corresponding task. The correlation between objective expertise and ACT-value is not significant, but shows the right direction and has a reasonable value of $r = .30$. The correlation between ACT-value and time shows similar results. The higher the ACT-value, the faster the participants executed the task. The corresponding correlation of $r = -.23$ was also not significant.

[2] The questionnaire with the scales can be found at http://o14-0er-inf1.ku-eichstaett.de/hmd/befragung/index.php. Log in with the user ”testuser” and the password ”testpass”

Correlation between	**r**	**significance**
ACT — subjective expertise	**0.66**	$p < 0.05$
ACT — objective expertise	0.30	n.s.
ACT — time	−0.23	n.s.
$Comp_h$ — subjective expertise	−0.25	n.s.
$Comp_h$ — objective expertise	−0.15	n.s.
$Comp_h$ — time	0.34	n.s.
$Comp_t$ — subjective expertise	−0.36	n.s.
$Comp_t$ — objective expertise	−0.16	n.s.
$Comp_t$ — time	0.30	n.s.

Table 3. Results of functionality NewTodo

The correlations between general computer expertise and performance showed again unexpected results. All correlations showed an unexpected direction. The higher the general computer expertise, the lower the subjective and objective performance of the users.

3.3 Discussion

The results were not as clear as desired, but nevertheless show interesting effects. The ACT-value seems to predict better the subjective expertise of a user than the objective performance. But all correlations were in the expected directions. The low results for the task NewAppointment can be explained by a ceiling effect. As this task is often used by all users, every participant mastered this task without problems. In contrast, the results for the task NewTodo show, that the ACT-value is capable to predict the subjective expertise quite well, independent of the general computer expertise. The unexpected results can be explained by the unusual composition of participants. The participants with high computer expertise used the target application less than the participant with low expertise. Therefore, it is not so unexpected that the participants with high computer expertise performed worse than the participants with low computer expertise. Unfortunately, this fact complicates the interpretation of the positive correlations between ACT-value and performance measures. As the values of these correlations stay nearly the same for partial correlations, where the influence of general computer expertise is deducted, the general statement lasts.

All in all, the results are promising. The procedural knowledge represented by the ACT-value seems to be a better performance estimate than general computer expertise, especially for subjective performance. The results are better for tasks that are used infrequently than for tasks that are used very often. But infrequently used tasks are much more the focus of a help system, than task that are used very often. The low number of participants can explain the lack of significance of most of the correlations, but also limits the power of the results. The modeling of the user's knowledge through ACT-values seems to go in the right direction. Further evaluations will show, if help systems that adapt to ACT-values successfully support the user in a helpful and non-frustrating way.

3.4 Further Work

According to the framework described in section 3, the next steps are the evaluation of the adaptation decisions of the system and the evaluation of the total interaction between user and system. The evaluation of the adaptation decisions is accomplished already. The first results show that people with different ACT-values prefer different components of the help items. The evaluation of the total interaction between user and system is in the planning phase.

4 Acknowledgments

I am grateful to the management and especially the employees of the hmd software-AG, who participated in this study and allowed me to protocol their actions in WebTime. It is not easy to find long-term participants, so I am very thankful for their cooperation. I am also thankful to Jörg Desel, Ioannis Iglezakis, and Richard Mote for patient support and fruitful discussions.

References

1. John R. Anderson and Christian Lebiere. *The atomic Components of Thought*. Lawrence Erlbaum Associates, Mahwah, NJ, 1998.
2. Fiorella de Rosis, Bernardina de Carolis, and Sebastiano Pizzutilo. User-tailored hypermedia explanations. In *Adaptive hypertext and hypermedia - Workshop held in conjunction with UM'94*, 1994.
3. L. Guttmann. A basis for scaling qualitative data. *American Sociological Review*, 9:139–150, 1944.
4. Kristina Höök. *A Glass Box Approach to Adaptive Hypermedia*. PhD thesis, Stockholm University, Swedish Institute of Computer Science, 1996.
5. Dorothea Iglezakis. Adaptive help for webbased applications. In *Adaptve Hypermedia and Adaptive Web-based systems*, pages 304–307, 2004.
6. Mark Neerincx and Paul de Greef. How to aid non-experts. In *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, pages 165–171, New York,NY,USA, 1993. ACM, ACM Press. SPSS help even worsens performance of the user.
7. Yvonne Waern, Nils Malmsten, Lars Oestreicher, Ann Hjalmarsson, and Anita Gidlöf-Gunnarsson. Office automation and users' need for support. *Behaviour & Information Technology*, 10(6):501–514, 1991.
8. Norman Walsh and Leonard Muellner. *DocBook: The Definitive Guide*. O'Reilly, 1999.
9. Stephan Weibelzahl. *Evaluation of Adaptive Systems*. PhD thesis, University of Freiburg, Germany, 2003.