Examples of Adaptive Web-based Educational Systems

Pinter Robert, Szedmina Livia, Sanja Maravic Cisar
Polytechnical College Subotica /Department of Informatics, Subotica, Serbia
probi@vts.su.ac.yu

Abstract—This paper will present some existing and on-line accessible Web-based educational systems, and the adaptation theory which is integrated in those systems.

I. INTRODUCTION

Over the past decade a lot of e-learning system was developed. Only a few of them integrates some form of adaptive technologies. Adaptive Web-based educational systems (Brusilovsky, 1999) emerged as an alternative to the traditional “one-size-fits-all” approach in the development of educational courseware. These systems build a model of the goals, preferences and knowledge of each individual student, and use this model throughout the interaction with the student in order to adapt to the needs of that student.

If we do not pay attention to the learner’s characteristics and requirements while developing an e-learning system and e-curriculum, it could cause the student’s discomfort level to rise to a level which then will hinder or prevent learning or further use of the learning system.

Besides that the problem is recognized, and there are theories which gave some aspect of how to handle this problem of creating adaptive e-learning systems, the concrete practical solutions are very rare. Unfortunately only a few of these adaptive systems are available on the web. Most of them are still in the research phase and cannot be accessed publicly.

This paper will present some existing and on-line accessible systems, and the adaptation theory which is integrated in those systems. The goal is to collect in one place and present those “working” systems, which could help in developing our own adaptive e-learning systems.

II. METHODS AND TECHNIQUES USED IN ADAPTIVE HYPERMEDIA SYSTEMS (AHS)

A method involves the notion of adaptation that can be presented at a conceptual level. Techniques operate on actual information content and on the presentation of hypertext links.

The next picture presents Peter Brusilovsky’s taxonomy of adaptive hypermedia technologies [Brusilovsky, 2001]:

![Adaptive Hypermedia Technologies](image)

Figure 1. The taxonomy of adaptive hypermedia technologies

A. Adaptive presentation

It is important that the information will be presented in different ways, depending on the user’s knowledge, goals or other features. For example there should be a difference in presenting the same subject to a novice and an expert user. The adaptive presentation uses the following methods:

- Additional, prerequisites, and comparative explanations: the goal is to provide additional information, explanations, examples, illustrations etc. for users who appear to need them. At the same time
the system can hide explanations which can confuse the user.

- Explanation variants: the aim of this method is that same information can be presented in different ways. The form of presentation depends on the user’s model.

- Sorting: this method determines the order of presenting the more or less independent subject fragments. For instance, there are students who prefer to see an example before the definition etc.

B. Adaptive navigation support

This part of AHS has the task to adapt the document’s rich hyperlink structure. The goal is to guide the user, reduce orientation problems, and help them to find the relevant while keeping them away from the non-relevant information. Brusilovsky mentions five adaptive navigation support methods:

1. Global guidance
2. Local guidance
3. Global orientation support
4. Local orientation support
5. Managing personalized views

Guidance means that if the user wishes to learn something more about the topic, the system may suggest some relevant pages (global) or suggest the next step to take (local).

Orientation support indicates that the system can present the structure of the hyperspace in global picture or presenting only some part of it. The adaptation is present through marking the relevant links (where to go), the visited links and the links that are still to be avoided. Managing personalized views means that each view may be a list of links to all the pages or sub-parts of the whole hyperspace that are relevant for a particular working goal.

C. Adaptive navigation support techniques

- Direct guidance: the system marks one of the links which is supposed to be the “best to follow”. The system also can generate additional links.

- Adaptive link sorting: depending on the user model, the system can sort the links on the page. For example: the more relevant the link, the higher up on the page it will be.

- Adaptive link hiding: the system prevents the use of (still) non-relevant links. This can be done by hiding, disabling or removing the links.

- Adaptive link annotation: With these techniques, the system can indicate the relevance of a link. We can use different colors to distinguish the “highly-relevant”, “somewhat relevant” or “non-relevant” links.

- Adaptive link generation includes three cases: discovering new useful links between pages and adding them permanently to the set of existing links; generating links for similarity-based navigation between items; and dynamic recommendation of relevant links.

- Map adaptation: the system can reorganize the overall link structure of the hyperspace.

The most existing adaptive hypermedia systems have three main (theoretical) parts where (some of) the above presented techniques are implemented. These parts are:

- An information domain – it is the information content of the application.
- A user profile – each user is represented by features (knowledge, preferences, goals etc.), this will determine the adaptation.
- An adaptation description – describes how the system is going to generate the adaptation.

III. ADAPTATION EXAMPLES

A. cWADEIn and jWADEIn

The evaluation of expression in the C, C++, and Java programming languages is one of the most difficult topics. Students have problems with both understanding the order of execution and the semantics of operators. To help the students the authors developed cWADEIn and jWADEIn. Both systems allow students to explore the process of expression evaluation step-by-step with detailed visualizations and explanations. They also help them to check their understanding of the order of execution and semantics of operators. From research point of view, the goal of the systems is to explore adaptive explanatory visualizations. Either of the systems can work in two modes: exploration and knowledge evaluation.

In the exploration mode the system presents complex visualizations. Color-coding is used to facilitate understanding. Animations are used in many contexts. As the student progresses the speed of those animations increases. Eventually, animations are collapsed. Each visualization step is accompanied by textual explanations. Those explanations are context-sensitive and attempt to shed light on the step at hand as opposed to giving more general information. As the student progresses explanations are being collapsed as well. Eventually, no explanations are shown. In the knowledge evaluation mode the systems checks the student’s knowledge of execution order and semantics of operators.

B. QuizGuide

QuizGuide is an adaptive system that helps students in selecting most relevant self-assessment quizzes. Quizzes
are assigned to topics. QuizGuide uses adaptive annotation, one of the adaptive navigation support technologies, to show every student which topics are currently most important and which require further work.

Figure 3. Student interface of QuizGuide

C. NavEx

Navigation to Examples - is a Web-based adaptive navigation tool for exploring programming examples. Extends traditional performance-based annotation. NavEx uses concept indexing of the examples and provides these information to students.

- What examples the student is ready to browse and what s/he is not.
- What is the student's current progress.
- What examples are the most relevant for the student.
- Progress of the student with relation to the group.

Figure 4. The student interface of NavEx

D. Knowledge Sea II

Knowledge Sea II is an extension of Knowledge Sea project that was designed as a mixed corpus C programming resource that tries to bridge the gap between closed corpus materials in the form of lecture notes and open-corpus materials in the form of the set of the links to online resources for C programming.

Knowledge Sea II is designed to help users navigate from lectures to relevant online tutorials in a map-based horizontal navigation format. Every cell of the KnowledgeSea map includes links to online material that are related to keyword presented on the cell. The adjacent cells present similar material. The most important feature of KnowledgeSea is facilitating the navigation through providing traffic and annotation based social navigation support.

The personalization is helped with the icons on the Map:

- Paper Stack Icon 📚: The paper stack represents the density of documents inside the cell. 3 levels of paper stack icon could be seen on the cells ( 📘, 📚, 📚). The larger number of paper represents the larger number of documents inside each cell.
- Sticky Note on Paper Stack 📚: The sticky note icon on the paper stack represents that at least one document inside the resources of this cell has some written notes by students.

Figure 5. NavEx with Social Navigation

Figure 6. KnowledgeSea II
• User Icon 🔄: The user icon on each cell, represents the number of visits made by the student (user traffic) to any document of that cell. Five different levels of user traffic is represented by user icons with five different shades of blue 🔄, 🔄, 🔄, 🔄. The darker blue is representation of larger number of visits (heavier level of user traffic).

• Background Color: Background color of each cell represents the number of visits made by other students (group traffic) to any document of that cell. The darker the color of the background the larger the number of visits (heavier level of group traffic).

IV. SUMMARY

The goal of this presentation is to presents some existing educational hypermedia systems, which use some form of adaptation to the special needs of the student. Unfortunately, there is only a few of those systems, which can be accessed publicly. For those who want to develop their own Web-based education system, it is important to see the state-of-the-art of this topic.

Because there is no methodology for creating this kind of neither educational systems nor e-curricula, we considered it is worthy to collect and present (without details) those systems, which show some concrete adaptation mechanism.

All presented applications are available from the http://adapt2.sis.pitt.edu/cbum web adress.

REFERENCES

[12]