ALEF: Web 2.0 Principles in Learning and Collaboration

Mária Bieliková, Marián Šimko, Michal Barla, Daniela Chudá, Pavel Michlík, Martin Labaj, Vladimír Mihál, Maroš Unčík

Abstract: *In this paper we present ALEF, a framework for adaptive web-based learning 2.0 that integrates adaptive learning with key concepts of emerging Web 2.0. We outline framework architecture and describe its basic features. We introduce basic modules built on the framework as a part of a prototype implementation for learning lisp programming.*

Key words: *Adaptive Web-based Learning, Framework, Collaboration.*

INTRODUCTION AND RELATED WORK

Web-based educational systems of today continue in addressing the key concepts of Web 2.0. The "2.0" learning environments allow users to create and edit the content, to annotate it and post comments on the top of it and to support social learning (e.g., Moodle [3]). However, current (*non-adaptive*) learning environments do not take into account differences among students and provide "one-size-fits-all" instead of personalized learning experience. This is due to the fact that underlying models of a virtual learning environment often miss explicit and rich metadata descriptions used for inferring by adaptation engines. The complexity of adaptive course authoring is one of the major drawbacks and obstacles when integrating adaptive features into any learning environment [8].

On the other hand, we are not aware that *adaptive* educational systems leverage Web 2.0 advantages to a large extent; especially when integrating and/or combining different approaches to adaptation or when considering social aspects and collaboration. The authors of KnowledgeTree identified the main drawbacks of adaptive web-based educational systems (lack of integration of multiple aspects of learning into one system, poor reuse support) and proposed a distributed architecture which allows a student to reach various interactive learning activities from one single point [1]. However, a drawback of the whole approach is that the activities within a system are still kept separated – the system provides a student with all relevant links pointing to distinct services needs to make a decision whether he or she wants to study the explanatory materials, take some questions and quizzes or practice the acquired knowledge on several exercises.

Related issues were addressed also in [7]. Similarly to our work, authors identified that adaptive educational web-based systems should follow Web 2.0 principles in order to be successful. The authors proposed a framework consisting of five web services which integrate into and extend Moodle LMS by personalization and adaptation features and take advantage of its Web 2.0 presentation layer which provide collaborative parts of the course (activities, blog etc.). The weak point of the solution is that student modelling process as well as whole personalization is separated from collaborative processes – the collaboration within Moodle environment cannot directly influence neither any of the five web services nor their underlying models and thus drive the behaviour of the system.

In order to overcome the aforementioned drawbacks, we integrate all available learning and supporting activities into one single framework, which makes them easily accessible from any point of the learning flow. While following the emergence of Web 2.0 features, we develop ALEF, Adaptive LEarning Framework for creating adaptive and interactive web-based learning systems. The conception of ALEF shifts adaptive learning towards Web 2.0 by building on three key principles:

- *Domain modeling* with respect to possibility to automate certain domain model parts creation, and collaborative social aspect and the need to modify or alter domain model by students themselves.

- *Extensible personalization and course adaptation* based on the comprehensive user model, which allows for simultaneous employment of different adaptive techniques to enhance the student's learning experience.
- *Student active participation in learning process* with the ability to collaborate, interact and create content by means of *read-write web* vision. In particular, we exploit different types of annotations as a suitable way to allow for rich interactions on the top of the presented content.

ALEF supports including exercises into the educational course together with their personalized recommendation. Moreover, we developed methods for analyzing submitted solutions of exercises for similarity and style (both texts in formal programming language and texts in natural language, particularly Slovak texts) [2]. It can serve both for improving learning and for plagiarism detection.

In this paper we present an ALEF implementation for learning lisp programming. We focus on description of particular modules – building blocks of adaptive web-based learning 2.0. In particular, we introduce *content recommender*, *sidebar navigator*, *annotation creator* and *collaborative question creator*.

We describe these building blocks by dividing them according to the activities present during an interaction with the learning environment. We recognize two activity flow types: learning and collaborating/creating. Within the learning flow (Fig.1, solid line), a *personalizer* module plays the key role. Based on a *user model*, it accesses the *domain model* in order to select and personalize educational material to be presented in *presenter*. A *collaborative adaptive content creator* is a basic element of collaborating/creating flow (Fig.1, dashed line). It enriches presentation from the presenter in order to allow users to assign various types of annotations such as tags or comments to learning objects. The advantage (and the contribution to the state-of-the-art) of ALEF is the possibility to incorporate several personalizers or collaborative adaptive content creators at once, causing educational system to be truly integrated interactive learning environment.

PERSONALIZERS IN E-LEARNING

Content recommender

The objective of adaptive navigation is to help the student to choose the best topics or learning objects to focus on, in order to maximize the learning efficiency. As a key factor we consider the limited time for learning, which might be not sufficient for mastering all required *concepts* (i.e., domain knowledge elements) perfectly. Our strategy is to cover all concepts to some extent rather than to master only a few concepts perfectly.

To achieve this, we estimate potential gain of knowledge at the end of a learning session by considering current learning speed (determined from gain of knowledge from the start of the learning session) and remaining time. Estimated gain of knowledge is divided between the concepts so that the final estimated knowledge levels of concepts respect the importance of a particular concept given by a course author (i.e., a teacher).

To make a recommendation for a student, each learning object is evaluated and assigned a real number as its appropriateness for a student.

Three criteria are used for each learning object evaluation.

- Concept appropriateness for a student. A decision whether a student should learn the concepts covered by a particular learning object. It depends on a teacher-given concept importance, a student's knowledge level of the concept and concept.
- Learning object difficulty appropriateness. A learning object difficulty should match a student's knowledge level to prevent the student from being uninterested or

discouraged. This also maximizes the information value of the student's feedback [6], thus improving student knowledge modelling.

- Time period since the student's last access the learning object. Recommending recently viewed learning objects is suppressed.

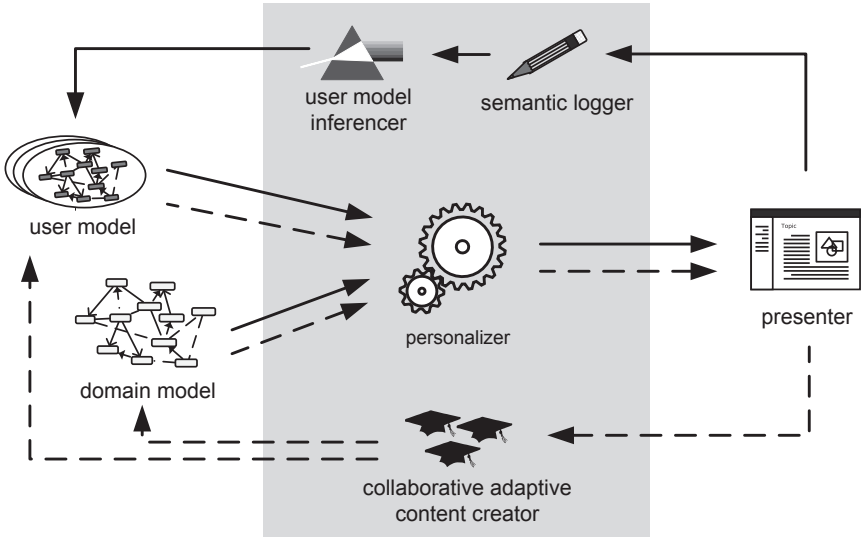

Fig.1. Activity flows within the Adaptive Web-based learning 2.0 environment

These criteria are orthogonal and for an ideal recommendation all of them are supposed to be met. Therefore, the final appropriateness of a learning object for the student is defined as the lowest value from the three partial results. After evaluating all learning objects in particular educational course, a predefined number of learning objects with largest appropriateness values are recommended to the student.

Sidebar navigator

Sidebar navigator focuses on tracing student's navigation within the learning object presentation page while learning. Main means of gathering and interpreting the user navigation are based on observation of user actions carried out by keyboard and mouse. Based on collected usage data from different users and taking into the account the social network of a particular student, it personalizes the visualization of certain learning object parts (e.g., by emphasizing mostly visited text). Basic concept of the *sidebar navigator* is similar to the *Read Wear* based on an idea of virtual "wearing-off" of scrollbar [4] – the more time the users spend on a particular portion of a long text (or other visual content), the more will the content become worn-off on this section.

The time spent on learning object portions indicates interesting or problematic textual fragments. Accordingly, the user model is updated by increasing user interest in concepts assigned to those fragments. Portions interpreted as problematic for a student can be planned into learning flow again later, or can be passed to a teacher for a review. By examining the read wear, frequently skipped fragments of the learning object can be visually dimmed or even left out. Fragments within the learning object can be reorganized considering frequent sequences of movement (scrolling). Dimming and reorganizing can significantly optimize students' workflows through learning object.

Data collected by the *sidebar navigator*, especially read wear data, also allow augmentation of social collaboration. Apart from the collaboration through a contribution to the content and respective collaborative adaptive content creators, direct collaboration

based on user communication (i.e., chat) can take advantage of read wear, navigation history and other interest indicators. In this way students have clues which parts took their friends or classmates some time, possibly interesting them or being problematic for them and therefore they are encouraged to communicate and collaborate.

COLLABORATING AND CREATING

Annotations creator

Annotations creator allows for inserting annotations to any resource within a course. Annotations are aimed to obtain fast and targeted feedback from students as well as their contribution to the learning content. We currently focus on comments created by students, which are inserted to user defined places in the learning content.

A student inserts a comment by selecting a portion of text in a learning object (a word or phrase) and writes a comment. The comment is associated with a particular position and text and is displayed within the learning object. Additionally, students can reply to the comment, creating *linked comments*. Many responses to the comment will result in an interactive discussion right 'above' the learning object. Students are also allowed to mark comments as incorrect or inappropriate. This kind of feedback helps the *annotation creator* to filter out confusing or misleading information.

Annotations placed within a learning content in particular positions provide students with another valuable benefit – information with added-value related to the content is collected and available in a convenient place. An example of such information is a link to a website with interesting learning material for further reading.

Moreover, annotations inserted by students can be also considered as a source of implicit feedback. Since every student's comment is associated with a location in the learning object and a small portion of the text, it is possible to associate every comment with a related concept. We can detect interactions between students and concepts that can result into the user model refinements. We have currently identified four kinds of interaction with comments: creation of a comment, marking a comment as incorrect, replying to a comment and deleting one's own comment. We explore the possibilities of user modelling using interactions of students with comments. Our aim is to discover relations between interactions with annotations and student's results.

Collaborative question creator

Collaborative question creator builds on the top of *annotation creator's* mechanisms and allows for creation of specialized annotations holding questions. During the learning, students are encouraged to create testing questions, which are displayed within a page. This way, students themselves become the creators of the educational content.

Collaboratively created questions are usually related to those content parts that are the most important, unclear, or controversial. Adding questions to those parts provides students with further learning possibility and thus supports the overall learning process.

Although the questions are added by students, our goal is to have questions whose quality is near to those provided by domain experts (teachers). The evaluation of question quality is based on the explicit feedback of students in conjunction with actions performed by the students (implicit feedback). It is also dependent on teacher's evaluation. The user explicit feedback is evaluated by taking into account the student model.

The entire method is divided into four processes (not necessary sequenced in the following order): *(i)* adding a question, *(ii)* answering a question, *(iii)* rating a student, *(iv)* rating a question. The first two processes are short-term (from user's point of view). The third and fourth processes are long-term processes as it takes some time to gather required amount of ratings.

According to these processes a student model holds facts such as a student creates, answer or rates explicitly a question. The rating of a question quality is similar to a rating of student's knowledge. It derives from the explicit rating of questions coming from students and implicit rating, based on the student's actions in the system.

For students' motivation we created simple game. Students gain points, which determine their ranking among their peers. Students should discover a strategy of balance between adding questions and rate questions similarly to others' ratings. We evaluated our approach as a part of the Functional and Logic Programming course last academic year where we considered a reward in form of bonus points included in the overall assessment.

CONCLUSIONS AND FUTURE WORK

We presented basic concepts of Adaptive Web-based Learning 2.0 reflected into ALEF, the framework for adaptive learning. The framework builds upon domain modelling designed to simplify automated course metadata creation [8] with respect to collaborative social aspect of creation. It follows up previous works in this domain [5, 9], while allowing for an extensible personalization and course adaptation based on comprehensive user model and supporting students active participation in the learning process with the ability to collaborate, interact and create content by means of read-write web vision.

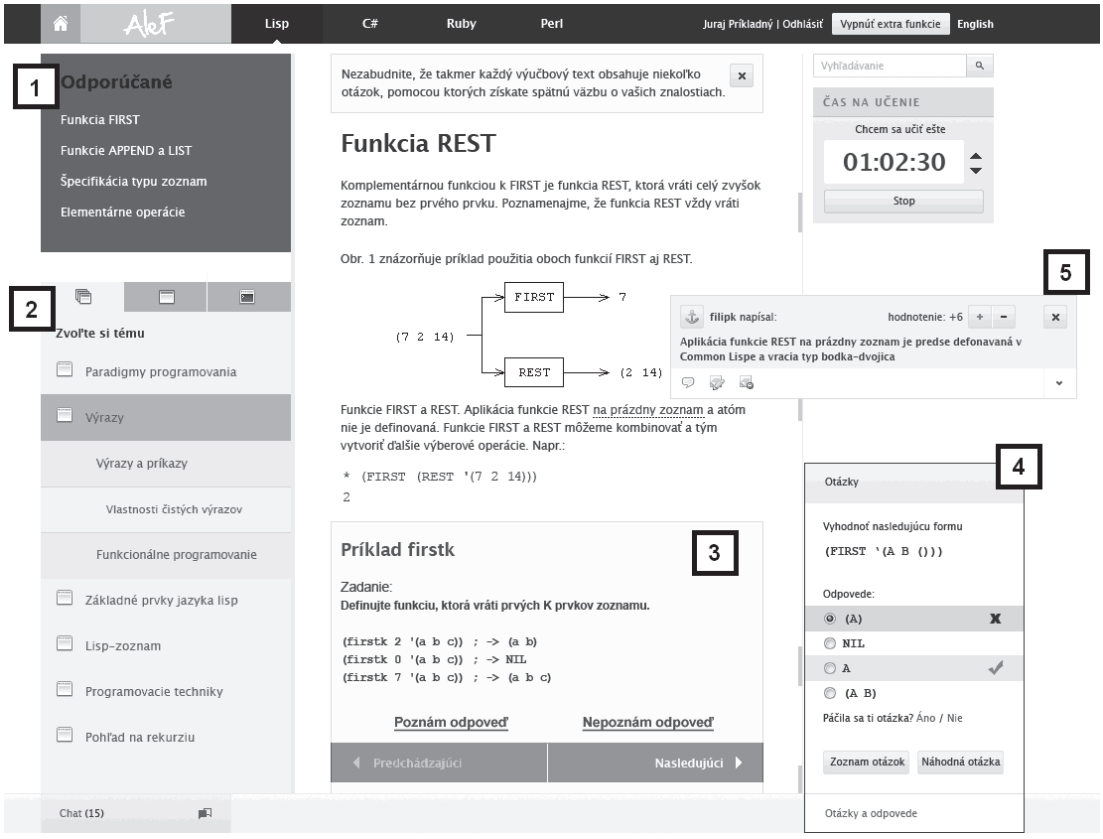

Fig. 2. Screenshot of ALEF user interface for learning lisp programming (in Slovak). Recommendations coming from *content recommender* are presented either in a separate box (1) above the menu or can be embedded within the main content in the form of interactive examples (3). *Sidebar navigator* visually emphasizes more often read text (2). *Collaboratively created questions* related to current learning object are visualized on-demand in a pop-up widget (4). Displayed content is enriched by adding different types of *annotations*, accessed by hovering the mouse over the underlined sections of text (5).

We used the proposed framework to build an environment for learning lisp programming that combines different learning activities (such as learning from explanatory texts, questions or exercises) along with interactive environment of the Web 2.0 (Fig.2). Currently we are working on an evaluation of developed modules.

ACKNOWLEDGMENTS

This work was supported by the Cultural and Educational Grant Agency of SR, grants No. 028-025STU-4/2010 and 345-032STU-4/2010.

REFERENCES

[1] Brusilovsky, P. KnowledgeTree: a distributed architecture for adaptive e-learning. WWW 2004, Posters, New York, NY, USA, ACM, New York, NY, pp. 104–113 (2004).

[2] Chudá, D., Návrat, P. Support for checking plagiarism in e-learning. In Procedia - Social and Behavioral Sciences, Vol. 2, No. 2, Elsevier, pp. 3140–3144 (2010).

[3] Dougiamas, M. Taylor, P. Moodle: Using Learning Communities to Create an Open Source Course Management System. In Proc. of World Conf. on Educational Multimedia, Hypermedia and Telecommunications 2003, AACE , pp. 171–178 (2003).

[4] Hill, W. C., Hollan, J. D., Wroblewski, D., McCandless, T. Edit wear and read wear. In Bauersfeld, P., Bennett, J., Lynch, G. (Eds.) Proc. of the SIGCHI Conf. on Human Factors in Computing Systems CHI '92. ACM, NY, pp. 3–9 (1992).

[5] Kostelník, R. Bieliková, M. Web-Based Environment using Adapted Sequences of Programming Exercises. In: M. Beneš (Ed.) ISIM'03, MARQ Ostrava, pp. 33–40 (2003)

[6] Linacre, J., M. Computer-Adaptive Testing: A Methodology Whose Time Has Come. In MESA Memorandum no. 69 (2000).

[7] Meccawy, M., Blanchfield, P., Ashman, H., Brailsford, T., Moore, A.: WHURLE 2.0: Adaptive Learning Meets Web 2.0. In LNCS 5192, Proc. of the 3rd Int. Conf. on European Conf. on Tech. Enhanced Learning, ECTEL '08. Springer, pp. 274–279 (2008).

[8] Šimko, M., Bieliková, M. Automated Educational Course Metadata Generation Based on Semantics Discovery. In LNCS 5794, Proc. of 4th European Conf. on Technology Enhanced Learning, EC TEL'09. Springer, pp. 99–105 (2009).

[9] Vozár, O., Bieliková, M. Adaptive Test Question Selection for Web-based Educational System. In: Proc. of SMAP'08. CS IEEE Press, pp. 164–169 (2008).

ABOUT THE AUTHORS

Prof. Mária Bieliková, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 473, E-mail: bielik@fiit.stuba.sk

Marián Šimko, doctoral student, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 231, E-mail: simko@fiit.stuba.sk

Michal Barla, doctoral student, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 231, E-mail: barla@fiit.stuba.sk

Dr. Daniela Chudá, Institute of Informatics and Software Engineering, Slovak University of Technology, Phone: +421 2 602 91 496, E-mail: chuda@fiit.stuba.sk

Pavel Michlík, master degree student, Software Engineering, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail: xmichlik@stuba.sk

Martin Labaj, master degree student, Software Engineering, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail:xlabajm@stuba.sk

Vladimír Mihál, master degree student, Software Engineering, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail: xmihalv@stuba.sk

Maroš Unčík, bachelor degree student, Informatics, Institute of Informatics and Software Engineering, Slovak University of Technology, E-mail: xuncik @stuba.sk