ALEF: from Application to Platform for Adaptive Collaborative Learning

Mária Bieliková, Marián Šimko, Michal Barla, Jozef Tvarožek, Martin Labaj, Róbert Móro, Ivan Srba and Jakub Ševcech

Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, Ilkovičova, 842 16 Bratislava, Slovakia
{name.surname}@stuba.sk

Abstract. Web 2.0 has had a tremendous impact on education. It facilitates access and availability of learning content in variety of new formats, content creation, learning tailored to students' individual preferences, and collaboration. The range of Web 2.0 tools and features is constantly evolving, with focus on users and ways that enable users to socialize, share and work together on (user-generated) content. In this chapter we present ALEF – Adaptive Learning Framework that responds to the challenges posed on educational systems in Web 2.0 era. Besides its base functionality – to deliver educational content – ALEF particularly focuses on making the learning process more efficient by delivering tailored learning experience via personalized recommendation, and enabling learners to collaborate and actively participate in learning via interactive educational components. Our existing and successfully utilized solution serves as the medium for presenting key concepts that enable realizing Web 2.0 principles in education, namely lightweight models, and three components of framework infrastructure important for constant evolution and inclusion of students directly into the educational process – annotation framework, feedback infrastructure and widgets. These make possible to devise and implement various mechanisms for recommendation and collaboration – we also present selected methods for personalized recommendation and collaboration together with their evaluation in ALEF.

Keywords: personalized recommendation, Web 2.0, collaborative learning, adaptive learning, educational platform

(1) Introduction

Technology has shaped the way people learn for decades. A particularly great influence of technology on learning came with the emergence of the Web in 90s. But it was the next generation of Web, so called Web 2.0, which significantly shifted the existing paradigm of learning.

In general, Web 2.0 made the experience more interactive, empowering users with easy-to-use tools. It enabled user-based authoring of content (by utilizing blogs and wikis) and facilitated organization and sharing of knowledge (by annotating and tagging content, discussing content). It also simplifies collaboration and interaction between users. Users in web-based systems are no longer only content consumers, they have become content creators themselves and indeed they have started to actively contribute to the Web's content as envisioned by Berners-Lee (2005).

An important implication is that Web 2.0 reflected into improved user experience during learning in web-based educational environments. A user – *learner* – gains more competences that result into greater autonomy for the learner. The traditional role of a teacher changes and distinction between teacher and student blurs (Downes 2005).

Together with the increasing popularity and spread of the Web, we witness significant growth of educational materials available online. In order to allow effective learning techniques for adaptive navigation and content presentation adaptive web-based educational systems were devised almost two decades ago (Beaumont and Brusilovsky 1995). A common example of adaptive navigation is *recommendation* of learning objects. The recommendation methods tailor the presented content to a particular learner and/or support a learner by providing adaptive navigation. Most current adaptive web-based educational systems attempt to be more intelligent by advancing towards activities traditionally executed by human teachers – such as providing personal advices to students (Brusilovsky and Peylo 2003).

We see both collaboration and adaptation as key concepts facilitating learning in current web-based educational systems. Opportunities introduced by emergence of Web 2.0 imposed new requirements for adaptive web-based learning that should respond for constant change and inclusion students directly into educational process. The requirements shifted to the following criteria (Šimko et al. 2010):

- Extensible personalization and course adaptation based on comprehensive user model, which allows for simultaneous use of *different adaptive techniques* (such as recommendation) to enhance student's learning experience.
- Student *active participation* in learning process with the ability to collaborate, interact and create content by means of the 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.
- *Domain modeling* that allows (i) automation of domain model creation, and (ii) collaborative social aspect and the need to modify or alter domain model by students themselves.

In order to address the challenges posed on educational systems in Web 2.0 era and beyond, we developed ALEF – Adaptive LEarning Framework (Šimko et al. 2010). We have followed up on the prior research on adaptive learning at the Slovak University of Technology including adaptive learning applications ALEA (Kostelník and Bieliková 2003) and FLIP (Vozár and Bieliková 2008). ALEF now constitutes both a framework for adaptive collaborative educational systems and an instantiated system created primarily for research purposes, but used successfully in educational process at the Slovak University of Technology. After several years of research, ALEF became a base for various autonomous components, some of which present standalone applications, so now ALEF can be viewed rather as a platform for adaptive collaborative web-based learning.

The ALEF platform offers recommendation on various levels. The recommendation is not only on the level of course parts as a whole (learning objects), but also content outside of the integrated course material is recommended through annotation with information gathered from external sources. Content and information within the learning objects is recommended through summarizations.

In this chapter we present Adaptive Learning Framework ALEF. We focus on recommendation and collaboration in ALEF, which aims at delivering tailored learning experience via personalized recommendation, and enabling learners to collaborate and actively participate in learning via interactive educational components. We present not only functionality realized in ALEF, but also an infrastructure for providing this functionality, which facilitates personalized recommendation and active collaboration – domain model, user model and unique framework components: annotation framework, feedback infrastructure and widgets. Core part of this chapter discusses recommendation, which is performed in ALEF on several levels – on the learning objects level and on the content of learning objects where we provide also summarization which recommends particular parts of learning objects for effective repeating. Next, we present a concept of implicit and explicit collaboration in ALEF. This part is related to the recommendation as during collaboration several decision points where recommendation is useful concept. We conclude this chapter with short summarization and future directions.

(2) Related Work

Adaptive and intelligent web-based educational systems address the new challenges related to impact of Web 2.0 on education in various ways. The same way as a good teacher adapts instruction to individual student's needs the adaptive and intelligent web-based educational system provide adaptive features (e.g., adaptive content presentation and navigation support) and intelligent features (e.g., problem solving support and solution analysis). The emergence of Web 2.0 technologies with its focus on user also changed user expectations. Users now expect that a learning system adapts according to their previous interactions, they expect to be able to actively participate in communities, collaborate and share their work.

Consequently, modern adaptive and intelligent web-based educational systems incorporate collaborative aspects such as knowledge sharing and organization (e.g., annotation and tagging of learning content, discussion forums), synchronous and asynchronous group work, and user-oriented content authoring (e.g., wikis).

User participation via Web 2.0 tools that enable creation, rating and sharing learning content drives the emergence of learning networks (Koper et al. 2005), which provide methods and technology for supporting personal competence development of lifelong learning, typically in an informal setting. Learning networks are structured around tags and ratings, which are often only sparsely provided by users, raising additional strain on recommendation methods in this setting. TEN-competence project is the largest EU-driven initiative that studies bottom-up approaches of knowledge creation and sharing.

There are two possible ways to take when building a modern adaptive learning system: (1) integrate adaptive features into an existing Learning Management System (LMS) such as Moodle, or (2) design and build an adaptive learning system from scratch. Some authors argue that the adoption rate of adaptive technologies in learning remains low mostly due to limited feature set of existing adaptive learning systems (Meccawy et al. 2008). The learning systems are usually experimental prototypes designed and developed from scratch and not used beyond the university departments of their authors. Consequently, Meccawy et al. propose the WHURLE 2.0 framework that integrates Moodle's Web 2.0 social aspects with adaptation features. Their design follows the typical service-oriented architecture of other adaptive learning systems such as the distributed architecture of KnowledgeTree proposed by Brusilovsky (2004). KnowledgeTree architecture is based on distributed reusable learning activities that are provided by distributed activity servers, while other types of servers provide the remaining services which are required in every adaptive learning system: domain modeling, student modeling, and adaptation engine. The service-oriented architectures facilitate reusability of learning content and learning analytics across different services provided by the learning system.

Modern adaptive and intelligent web-based educational system is expected to provide diverse learning content and services to students. The content can range from non-interactive course material, simple quizzes and exercises to highly interactive synchronous collaborative learning. The basic services include the generic LMS services such as course administration, and automatic quiz/exercise evaluation services. Additional services result from the adaptive and social properties of the learning system. Each bit of the learning content is

1. adapted in various ways (e.g., student's needs, preferences or knowledge, teacher's requirements), and is
2. socially enabled by providing knowledge sharing, group work and user content authoring facilities. These services are typically backed by methods based on artificial intelligence and presented within a user interface that is continuously recording each user action providing back the data for analysis by the adaptation methods. Examples include methods for course material personalization and recommendation according to student's knowledge or time constraints.

Recommendation in education brings about additional requirements compared to methods of generic recommendation such as books or movies recommendation (Manouselis et al. 2010). The typical recommendation scenarios apply (e.g., predicting link relevance, finding good (all) items, recommending sequence of items, finding novel resources, finding peers/helpers) with the additional consideration of relevancy to learning goals and learning context. The recommendation must also account for various pedagogical rules.

Recommendation differs substantially based on the type of corpus used. Closed corpus recommendation systems can take advantage of detailed metadata description and/or ontological representation of the learning objects. Consequently, the recommendation systems can effectively personalize the learning process through adapting the learning content and/or the learning sequence. The recommendation methods can take into account the various learning goals, contexts and pedagogical rules. As examples we can mention an approach for semantic recommendation in education settings called SERS (Santos and Boticario 2011) or XAPOS system (Šaloun et al. 2013).

Open corpus recommendation, on the other hand, does not require preexisting metadata descriptions. The objects are often preprocessed with automatic metadata extraction methods, and the recommendation itself typically relies on collaborative filtering methods that are robust to noisy input. The recommendation results improve when more user/item data is provided over the course of the recommendation systems lifetime.

Personal learning environments enable even more personalized experience by providing facilities to build and personalize their own learning environment. The concept of PLEs and recommendation has been extensively studied in the ROLE project, approaches for recommendation specific to personal learning environments are outlined by Mödritscher (2010).

Adaptive and intelligent web-based educational systems are based upon domain and user models. User model often follows overlay student modeling that represents student knowledge and other characteristics on top of domain model. Several reference models for adaptive web-based systems have been proposed, such as Adaptive Hypermedia Application Model (AHAM) (de Bra et al. 1999), Munich reference model (Koch and Wirsching 2002), and LAOS (Cristea and de Mooij 2003) and its social extension SLAOS (Cristea et al. 2011). When considering domain modeling in these reference models, they often suffer from tight coupling between conceptual description of subject domain and content. Also, support for Web 2.0 paradigm on the level of domain modeling is limited in these models. Although there are attempts to incorporate social collaborative aspects (e.g., content annotations, tagging, rating, commenting) into adaptive web-based systems at abstract level, it has limitations in extendibility of interaction and collaboration in domain model and ability to support interaction and collaboration on top of user-generated entities (Cristea et al. 2011).

(3) Adaptive Learning Framework ALEF

ALEF's primary goal is to provide an infrastructure for developing adaptive collaborative educational web-based systems (Šimko et al. 2010). Besides its base functionality – to deliver educational content – it particularly focuses on making the learning process more efficient by (1) delivering *tailored* learning experience via recommendation/personalization, and (2) enabling learners to *collaborate* and *actively participate* in learning via interactive educational components. To facilitate both aims ALEF's architecture incorporates two core models and framework components:

- *domain model* – rich yet lightweight domain model semantically describes resources within a course,
- *user model* – overlay user model represents current state of user's knowledge and goals,
- *framework components* – extendable components such as annotations framework and widgets provide fundamental functionality related to adaptive web-based systems.

Models can be used easily in any learning domain, and together with extendable framework components they allow developers to build custom framework extensions, that is, shifting the notion of ALEF from a framework for adaptive web-based educational systems towards a modern web-based educational platform.

Overview of different framework components together with their close connection to domain and user model is displayed on Fig. 1. Individual models and frameworks are discussed in more details in the following sections.

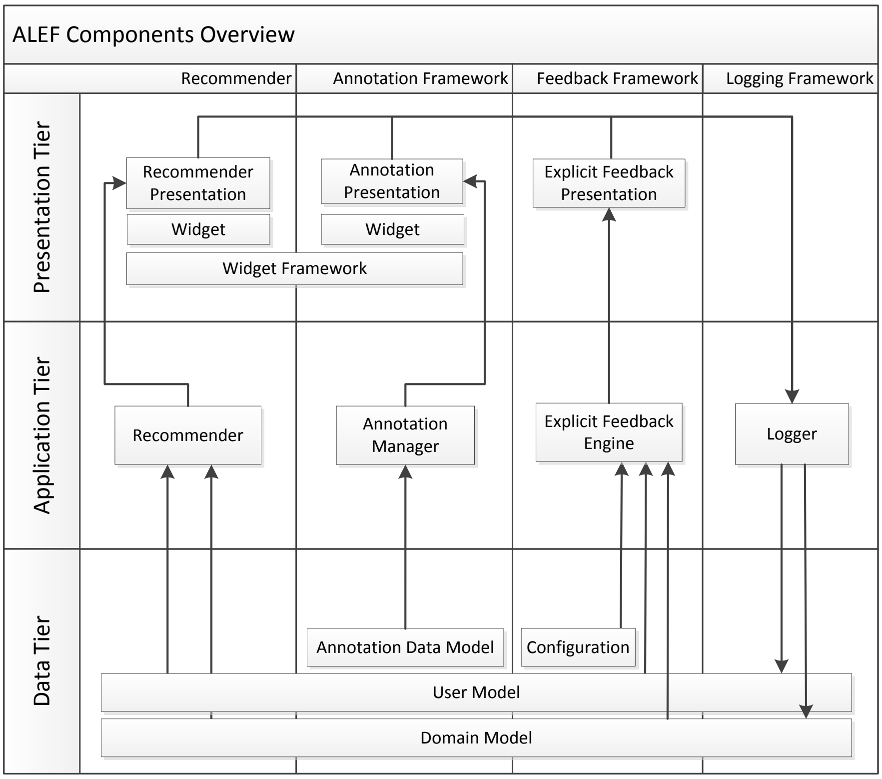

Fig. 1 ALEF components overview – three tiers architecture: data, application and presentation tier. Particular framework components spread across all of these tiers.

(3.1) Domain Model

In domain modeling, ALEF leverages the so-called lightweight semantics and proposed a lightweight domain model for adaptive web-based educational courses (Šimko 2012). We consider modern educational courses to consist of educational content[1] authored by teachers, and user-generated content (e.g., comments, tags) provided by students. The various types of user-generated content are represented uniformly as *annotations* – an abstraction representing user-generated content in ALEF.

Learning resources are not described using complex domain descriptions such as ontologies, instead, resources are described by *domain relevant terms*. The terms, relationships between terms, and their associations to resources constitute the core domain conceptualization that forms a basis for user modeling and is utilized by the adaptation engine. We take advantage of multilayer design that explicitly differentiates between resources, their abstractions and semantic descriptions (see Fig. 2) and clearly separates content from conceptualization.

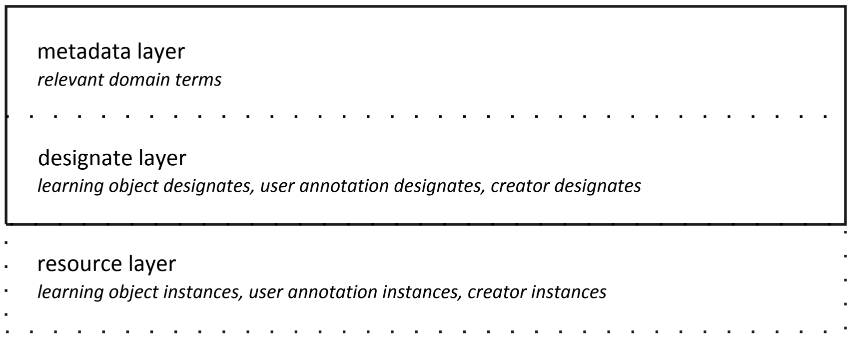

Fig. 2 Domain model scheme: metadata layer over designate layer. Resource instances are not a part of domain model (solid line).

Domain model consists of:

- designate layer, and
- metadata layer.

These two layers represent a conceptual abstraction over resource instances (both learning objects and annotations) that are created and modified by content authors. Resource instances form the actual learning content presented to learners (e.g., a learning object *Recursion basics* in a programming course).

[1] A basic component for education delivery is a learning object. For learning object we adopt a broader definition by IEEE, which defines a learning object as any entity, digital or non-digital, that may be used for learning, education or training" (IEEE, 2002).

Designate layer is further divided into *resource designates* and *creator designates*. Designate layer represents an abstraction of resources (learning objects, annotations) and their creators, and is crucial for ensuring reusability and extendibility in terms of content resource's lower level representation. The concept of resource creators was introduced to domain model since in social and interactive environment it is important to explicitly model creator relations to both resources and metadata. In the social and interactive environment, different creators produce content (educational content, annotations and metadata descriptions) with various degree of "reliability", which must be taken into account by algorithms later in the processing chain when accessing domain model elements (e.g., for recommendation of learning objects or annotations filtering).

Metadata layer is formed by domain relevant terms – easy to create descriptions that are related to particular domain topics (that are not explicitly represented in domain model). It is important to note that relevant domain terms do not represent concepts in strict ontological definition, cf. (Cimiano 2006). They rather represent lexical reference to non-explicit topics or concepts, which form the domain model. Examples of relevant domain terms in the domain of programming involve *recursion*, *cycle* or *comment*.

Learning content is comprised of various types of learning objects such as explanations, exercises and questions. These elements are interconnected via various types of relationships that represent different forms of relatedness between domain model elements. In ALEF's domain model, we distinguish three (high level) types of element relationships:

- relationship between designates,
- relationship between designates and relevant domain terms,
- relationship between relevant domain terms.

Relationships between resource designates typically reflect relationships between resource instances (e.g., hypertext links or hierarchical book-like structure of learning objects), or creators and resources (authorship relation).

Relationships between resource designates and relevant domain terms represent lightweight semantic descriptions of resources. Such relationships arrange relevant domain terms in a lightweight semantic structure that is necessary to perform reasoning tasks. We refer to all these types of relationships as resource-metadata relationships. Note that each relationship type can be assigned arbitrary attributes, e.g., a relation weight.

A basic example of a relationship between resource and metadata is the relationship that associates resources with relevant domain terms representing its content. Examples of relationships between relevant domain terms include similarity relationship (e.g., *recursion* is-similar-to *cycle*), composition relationship (*comment* is-part-of *program*), and hierarchical relationship (*printf* is-a *function*).

When considering domain model in general, it is important to point to the issue of domain model authoring. Conceptual description of even a small domain typically contains hundreds of concepts – domain knowledge elements – and thousands of relationships. Providing conceptual descriptions manually is a very demanding task that teachers (adaptive content authors) can accomplish only with difficulties. ALEF benefits from the proposed lightweight domain modeling, which open doors for methods that can automatically create lightweight semantic descriptions, while preserving acceptable quality of personalization.

We devised such methods and showed that *automated* creation of domain model – ranging from relevant domain terms extraction (Šimko and Bieliková 2009a) to various types of relationships discovery (Šimko and Bieliková 2009b; Šimko and Bieliková 2012) – is to a great extent comparable to manual creation in terms of quality of produced domain descriptions as well as their suitability for learning object recommendation (Michlík and Bieliková 2010). Though our methods do not replace a teacher (which is hardly possible), but they can be used with advantage to support her/him when authoring adaptive courses.

(3.2) User Model

User model employed in ALEF is based on principles of overlay user/student modeling, that is, it adds several user-related layers on top of the domain model. The most basic layer is used to store interaction of users with domain elements, and contains mainly information about:

- which learning object student visited, how much time he/she spent reading it
- which questions and exercises student solved and how successful he/she was (e.g., whether he/she answered correctly right away or was forced to request a hint or did not manage to answer correctly despite of the hint provided)
- which additional resources (via annotations) student interacted with

This layer is basically representing the students' interaction history. On top of this layer sits an additional one that is used to store student characteristics (mainly knowledge of domain concepts (relevant domain terms) related to relevant learning objects).

Each such characteristic apart from its value (scalar from 0 to 1) and a timestamp contains:

1. *confidence* representing the level of certainty that a student does have this characteristic at this value, and
2. *source* of the characteristic (such as "self-reported" in case of a questionnaire, or discovered by a particular user model inference agent).

When a user model is updated, the update spreads through relationships among concepts using standard spreading activation algorithm (Crestani, Lee 2000). This ensures that any gain or loss of knowledge is appropriately distributed to all relevant parts of the overlay model following lightweight representation of domain model.

(3.3) Framework components

ALEF's architecture comprises the following easily reusable and extendable pivotal components:

- Annotation framework,
- Feedback infrastructure,
- Widgets

Annotation framework constitutes a robust framework for creating, updating, accessing and sharing annotations as a fundamental means for educational content enrichment and interaction. *Feedback infrastructure* streamlines and unifies the process of feedback collection and evaluation for various components and methods deployed within the educational system. *Widgets* represent building blocks of user interface. They are active learning and collaboration-supporting components and act as gateways for accessing learning content and annotations.

Annotation Framework

Students get more involved in the educational process through the possibility of adding different kinds of annotations to the content; they can create both new content and metadata. The *annotation framework* is designed to provide means and encourage this kind of participation (Šimko et al. 2011).

ALEF's annotation framework aims to support and standardize interaction with various types of annotations and to ease the development of new annotation types by providing a common software infrastructure. In order to achieve a high degree of reusability and extendibility, content and annotations share common representation within the framework.

Content and annotation are defined as the same entity – *Resource* (see Fig. 3). In this representation, *Resources* can be connected with *Relationships* of various types (e.g., *Annotates*). This allows not only to assign annotations to the content, but even to interconnect annotations with each other. It also allows to easily add a new annotation type by extending the *Annotation* entity, as well as to add a new content type, which is immediately annotatable by existing set of annotation types.

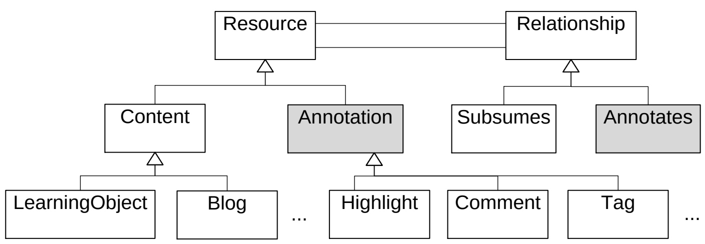

Fig. 3 Extensibility of resource annotations (Šimko et al. 2011).

Every annotation is defined by its *content* and *context*. Content is a piece of textual information added by a student in a form of annotation, such as a comment or URL of an external source. In a special case, it can also be empty (e.g., in case of a highlight). Context represents an association (binding) of the annotation to a learning object and in some cases also to the text, where the annotation has been originally inserted by the student. Whether an annotation has been assigned to the learning object as a whole or only to a specific fragment of its text (i.e., a word, a phrase, or a paragraph) differentiates two distinct types of annotations on the conceptual level: *per-text-annotations* and *per-content-annotations*.

Students can access annotations and navigate among them using both content and context: by context (*access-by-context*), i.e., directly in the text, where the annotation has been assigned in case of the per-text-annotations; by content (*access-by-content*), i.e., separately from the text (usually using a specialized *widget*).

We designed four distinctive user interface elements as a means for creating and accessing both the content and the context information of annotations:

- in-text interaction and presentation,
- sidebar,
- annotation browsers,
- annotation filter.

In order to create (using in-text pop-up menu, see Fig. 4, left), access and remove (by hovering the mouse over the text, see Fig. 4, right) per-text-annotations, students use *in-text interaction and presentation*, which represents the fastest access to annotations with no significant interruption of the learning process.

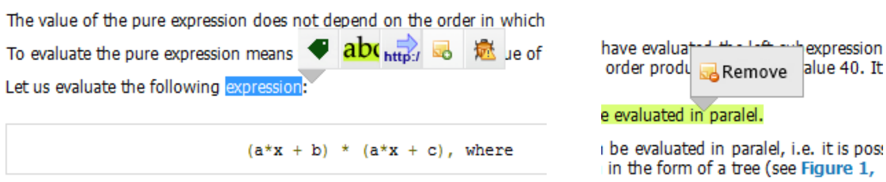

Fig. 4 In-text pop-up menu for creating a new annotation (left): tag, highlight, external source, comment and error report (icons in order from left to right); and removing an existing one (right); content in Slovak.

Sidebar represents another type of access-by-context navigation element. Annotations which are contextually close, i.e., were inserted in close proximity within each other in the text, are grouped into regions visualized on the sidebar. Hovering over a region shows a list of inserted annotations and highlights them in the text. Hovering over a particular annotation shows a tooltip with annotation's content; it also enables students to edit or remove the selected annotation (see Fig. 5).

Access-by-content navigation is provided by the *annotation browser*, which lists all annotations (of a specific type) related to the currently displayed learning object. Thus, students can interact with annotations regardless of their position in the text; however, selection or interaction with an annotation inside the browser invokes in-text visualization to indicate context of the annotation, if any. Annotation browsers are implemented as widgets located on the side of the screen, not distracting students from the main text in the central part. We provide more detailed description and examples of specific annotation browsers thereinafter in section on Implicit Feedback.

Annotation filter allows users (students as well as teachers) to select types of annotations to be displayed in the text as well as on the sidebar. Users can therefore focus their attention on selected types of information, resulting in more effective navigation among annotations. The filter contributes to adaptability of the learning environment towards learners' preferences and actual needs.

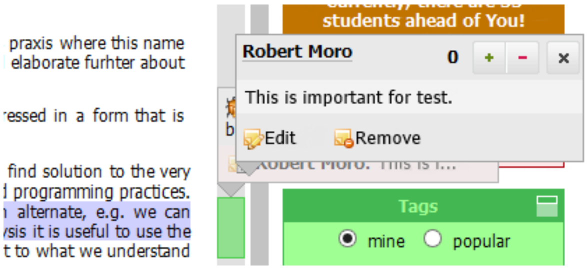

Fig. 5 Sidebar with a region of grouped annotations and a tooltip showing detail of a selected annotation with other interactive elements; content in Slovak.

Fig. 6 Annotation filter with all the annotation types set to be displayed.

Feedback infrastructure

The feedback infrastructure in ALEF was devised to alleviate two tasks: (1) monitoring student actions and building models, especially the user model, through the *logging framework*, and (2) evaluating personalization methods through *evaluation feedback engine*.

Logging framework ALEF combines many experimental methods implemented through multiple components. User (student) feedback gathered in any part of the educational system (e.g., commenting on a selected part of the learning content through annotations, solving an exercise) implies student's knowledge and interests. Therefore both implicit actions and explicit ratings from students are integrated into a common user model layer storing interaction with domain elements. This is ensured through logging framework, which acts as a proxy intercepting any action made by a student: (1) *before it is processed* – when it is being sent to the framework, e.g., the student clicked a button to evaluate the solution, (2) *after it was processed*, e.g., the solution was evaluated as correct and additional information to be logged can be included while processing, this is then collected by the logging framework.

Vast numbers of basic relationships to domain elements are created from both pre-processing and post-processing logging, including both implicit feedback, e.g., a student has looked on a fragment of a learning object (Labaj 2011), and explicit feedback, e.g., a student has rated difficulty of an exercise (Fig. 7) or rated learning object usefulness through a personalized rating scale. Advantage of this centralized logging pipeline is that one particular type of activity is always evaluated uniformly, regardless of a component which triggered the activity.

If any needed feedback or a new type of feedback from a new component is not yet logged, it can be easily added by creating a declarative description (processed by the logging framework) describing which relationship is to be created from which actions.

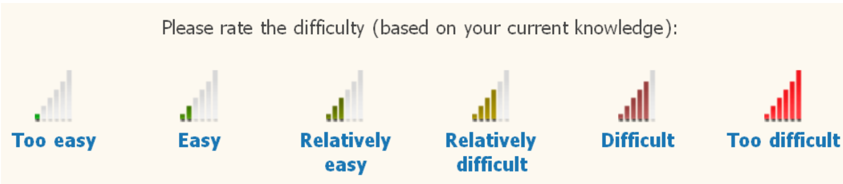

Fig. 7 Difficulty rating options displayed after student has finished a question or an exercise.

Evaluation feedback engine ALEF has been served and serves as a test bed for many experimental methods and often needs to collect diverse types of feedback. A rule-based *explicit feedback engine* was designed and developed in order to provide flexible feedback options. Besides generic question facility, it allows to display personalized questions instantiated from question templates.

An example of a typical problem in adaptive systems is when users do not use a newly added adaptive tool. If a recommended learning object is visited, we can easily evaluate the recommendation method on whether the student liked it or not and what knowledge did they gain after using it. However, when the recommendation facility is not used at all, we can only guess what is wrong with it (were the recommendations completely wrong so a student did not click on any recommended item or they simply overlooked them?). The evaluation feedback engine can help resolve such issues. Consider a scenario, which we realized in ALEF: a student is considering visiting a learning object from a list of recommendations presented to him (i.e., he looks on it, which we detect via gaze sensors) but then decides to use the menu instead. Right after the student makes the user interface action (mouse click on the menu item) the engine can ask a question about why the student chose the menu item rather than the recommendation. Gaze, as one of the indicators, is estimated either through mouse movement or through a commodity webcam by analyzing shape of the eye in the image feed in the browser. Using a simple low-resolution camera source or mouse movements brings smaller or larger errors to the gaze estimation, however these sources are sufficient for estimation of widget usage.

By personalizing the evaluation questions to the current context and asking them in appropriate situations, the engine can collect feedback of a better quality compared to when e.g., handing out predefined questionnaires after a learning session. More, we gather such feedback even when the students learn from remote locations such as at home.

The evaluation feedback engine allows to both declaratively describe situations in which a question should be displayed (and our logging framework allows for detection of various situations), and describe questions based on templates evaluated within student's current context.

Widgets

Annotation browsers, navigational and other components providing specific functionality (e.g., presentation of current student's score) are implemented within ALEF in the form of *widgets*. The main goal of our widget framework is to provide modular approach for designing and implementation of various functions for supporting students during their learning process and expose them in a uniform manner within the user interface. The widget framework provides standard functionality to all widgets (initialization, display of a widget, asynchronous state transitions, content refreshes, etc.), thus ensuring reusability and extendibility. It means that a developer can focus on design and implementation of widget's primary functions instead of solving various integration issues.

Widget framework is dynamic and flexible. It makes it easy to add a new type of widget, to change its default behavior, etc. It can even provide a gateway towards external systems, e.g., we have applied this widgets' framework to integrate ALEF with the PopCorm extension that is used for collaborative learning.

An example ALEF's user interface consisting of widgets is depicted on Fig. 8. The ALEF user interface is divided into three major parts: *navigational part*, *content part*, and *supporting components* providing specific functionality presented as widgets. Navigation through sets of learning objects is provided by navigational widgets such as hierarchical menu or list of recommendations. Various forms of interactions are enabled by incorporating annotations and interaction/collaboration widgets. Annotations constitute both a means for learners to better organize their own learning space, and also an interface for enrichment (contributions) to the learning content (the implementation of read/write web). We discuss specific widgets' usage (as annotation browsers) in section on Implicit Collaboration.

(4) Recommendation

The objective of recommendation in the domain of adaptive educational systems is to help students choose a topic which is best according to a combination of various factors: student's current state and goals, the actual available materials, as-

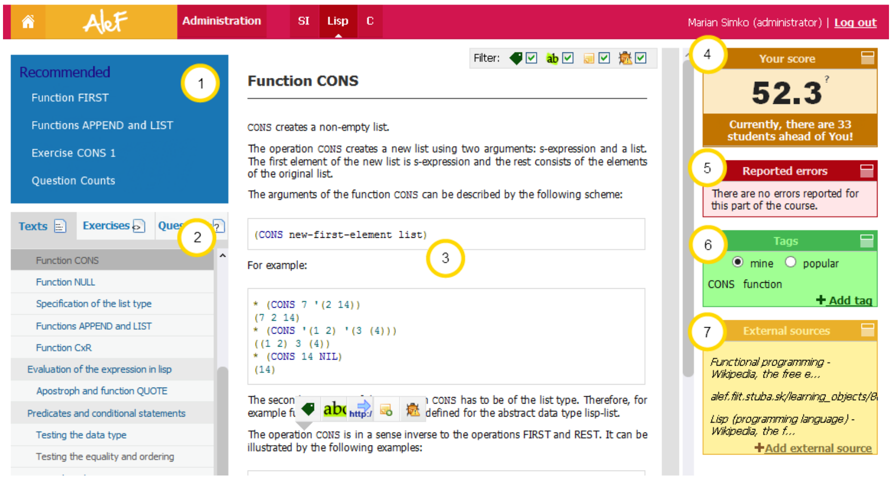

Fig. 8 Screenshot of ALEF user interface. It is divided to three vertical parts (left to right): (i) navigational part containing learning objects recommendations (1) and learning objects hierarchical menu (2); (ii) educational content containing selected learning object (3); and (iii) learning and collaboration supporting widgets: system activity score (4), error reporter (5), tagger (6) and external resource inserter (7); content in Slovak.

signments, etc. Recommendation approaches and presentation can vary according to typical workflow in given system. Are students given possibility to move through the course(s) on their own? Or is the system used as a support for courses being taught offline? Nevertheless, the recommendation should be personalized, as each student has different knowledge in various topics (both prior and during the learning), different learning pace, goals, etc.

The recommendation in ALEF is performed on several levels. First, learning objects (e.g., course material, programming exercise, quiz question) are recommended. Second, the content of learning objects (at any stage of completeness) is not usually everything that is known about a given topic, and students can take advantage of studying about it from external resources, for example information available on the open Web. ALEF recommends such information to students via automatic creation of annotations within learning objects.

Third, students may benefit from personalized summaries of learning objects. In the same way in which students can make use of additional content to the learning objects, other students may need only the most important pieces of information, some overview or quick reference to given learning objects. ALEF recommends the most important or relevant information within learning objects in the form of personalized summarizations.

The described levels are complementing each other. The recommendation of learning objects helps selecting a learning object to focus on. The summarization helps picking the most useful information within the selected learning object, while their augmentation by annotations expands the available information (extending the volume even beyond the scope of authored content). Examples of different recommendations delivered to the user are shown in Table 1.

In general, recommendations in ALEF are made based on information stored and maintained in user and domain models. User characteristics related to domain concepts (such as concept knowledge) represented by relevant domain terms are considered to select appropriate resources that are a subject of recommendation via relationships between domain model's metadata and designate layers. Relationships within metadata layer (connections among relevant domain terms) are typically used to update and spread information about student inferred from her/his actions. However, particular utilization of models depends on a recommendation method used.

Table 1 Examples of recommendations within ALEF platform.

Event description	Method	Example delivery	Example recommendation
Selecting an object	Meta-recommender (time-limited recommendation, sequential walkthrough recommendation, …)	Navigation widget Order of exercises in the menu Selection of exercises/questions placed inside an explanation Link to proceed to next exercise/question	"Try the following exercises now to learn the most in the remaining time (1:31 hour): Lambda REMOVE-IF, Scheme FIND" "Click for next exercise" (After the exercise was finished. It is not indicated which one will be displayed as next.)
Filtering information within the object	Adaptive summarizer	Summarization displayed instead of the object (quick reference) Summarization displayed after the object (repeating)	Collection of sentences from the text of the learning object
Accessing information outside the object	Automated annotation creation	In-text annotations Sidebar	"Construction of software includes transformation of detailed module specifications to program realization." (Term program is annotated with definitions and excerpts from external sources for "computer program", "programming language", and "process".)

(4.1) Recommending Learning Objects

For recommending learning objects, the ALEF supports multiple recommendation methods, which can be easily added. They are selected or combined on-the-fly for a given student by the means of meta-recommender, which is effectively a hybrid recommender system. ALEF uses weighted (recommenders are given weights and results are combined), switching (a recommender is selected for the given user at the current time), and mixed (multiple recommenders present their results at once) hybridization methods. Also where multiple domain or user models exist, a recommender can operate on any of them and the model to serve as a source for recommendation to the given user is personalized.

The methods supported within ALEF include traditional approaches found in adaptive web-based systems. One recommendation approach offered in ALEF – the sequential course walkthrough – is based on traditional recommendation principles of content similarity. Current student's learning interests are considered and based on them, several learning objects (LOs) are recommended in order to both (1) advance current subject further through course advancement in Explanations, and (2) refine knowledge being gained in the current subject through exercising in Exercises and Questions.

ALEF was used to experiment with a novel time-limited recommendation. The time spent learning is very important in the domain of learning – it is a form of currency that students "pay" for selecting items (here, learning objects) and of which each student has only a limited amount available. The proposed method for time-limited exercise and question recommendation is briefly described below.

The basic function of the time-limited recommendation is to help students in selecting appropriate assignments for learning the most and meeting the learning goals. In this method, assignments (Questions and Exercises) are recommended based on the student's knowledge of related topics, the target topics (e.g., knowledge required for a mid-term exam), together with time that the student has available for learning. The time limit is either determined externally by time remaining to an event (e.g., exam), or a student can allocate his/hers own available time. The learning targets were set by the domain expert (a teacher giving the exam), but can be also self-imposed by the students or set automatically.

The recommendation itself supports the recommendation task *find good items* (recommend a list of *N* most suitable assignments). Exercises are each composed of a task definition, a hint and a sample solution. Students can take various paths through the exercise (requesting the hint or not) and even when the student does not solve the exercise completely, he/she may or may not understand the sample solution. This is also being considered in the recommendation. Each available assignment in the course is assigned a scalar value of *appropriateness* computed from three criteria on a given assignment.

- *Appropriateness of related domain terms for the student*. Following three criteria apply to all domain terms related to the assignment which is being evaluated: (1) The domain term must be a member of the learning targets. (2) The student's knowledge of the domain term must be less than the estimated optimal value. The estimated optimal knowledge levels suppress further overlearning of domain terms which are already mastered at a satisfying level, to allow better learning of other domain terms, where current knowledge is still lacking. The optimal level is estimated by student's current progress (increment of knowledge over time) extrapolated to the end of learning session and evaluated against current knowledge level using sigmoid function. (3) The knowledge requirements of domain term prerequisites must be met. That is, some domain terms may be required to be mastered by the student before learning another domain term. These requirements are represented by weighted prerequisite relations in the domain model.

- *Appropriateness of difficulty*. In order to prevent the student from being discouraged, difficulty of the assignment to be recommended should match student's knowledge. Difficulty appropriateness for a given assignment is computed based on a Gaussian function with its peak set at the current aggregated knowledge level of all domain terms related to the assignment. Steepness of the curve is based on difficulty distribution among the assignments.
- *Time passed from last attempt to solve*. In order to prevent repeating the same assignment after a short interval, a time period from previous attempt to solve such assignment is considered. Immediately after visiting an assignment, appropriateness for this parameter drops to zero and gradually returns to 1 over time via hyperbolic function using time from previous attempt and student's feedback on the previous attempt, which determines function steepness.

All of these criteria are supposed to be satisfied; therefore the appropriateness of an assignment is the minimum of the partial values. The assignments to be recommended are selected as those with largest appropriateness.

Evaluation We evaluated the time-limited recommendation method in two experiments using Functional and Logic Programming course. In the first experiment, the students took a pre-test, studied for 60 minutes and then took a post-test. We divided 66 students into three groups: (1) a group with recommendation-based adaptive navigation using automatically generated domain model, (2) a group with recommendation-based navigation using manually created domain model, and (3) a group without adaptive features, all students navigated on their own. In the second experiment, the third group was provided with navigation using random recommendations and a 50-minute learning session was followed by a post-test. Results of the first experiment are shown in Table 2. Both experiments (Michlík and Bieliková 2010) had shown groups with personalized recommendation outperforming control groups (third group).

Table 2 Results of recommendation experiment using time-limited recommendation.

	Pre-test (%)	Post-test (%)	Difference
Group A: recommendation, automatic model	50,2 (±21,2)	70,5 (±15,2)	+20,2 (±15,2)
Group B: recommendation, manual model	42,4 (±21,0)	58,3 (±20,4)	+16,0 (±13,8)
Group C: no adaptive navigation support	48,2 (±25,4)	59,2 (±17,6)	+11,0 (±17,6)

(4.2) Summarization of Learning Objects

Automatic summarization can be useful for students in various scenarios. By providing a short summary containing the main points of a learning object it can help them to navigate in the learning object space; another scenario is revising before an exam by providing a longer summary explaining important concepts contained in learning objects. Thus, it can be framed as a recommendation problem: we want to recommend fragments of a document which are the most relevant (e.g., interesting, useful) for students in a given situation.

Conventional (generic) summarization methods summarize the content of a document without considering differences among users, their needs or characteristics. However, in adaptive learning systems we usually have many information sources that can be used to adapt summaries. We identified these three main sources in the educational system ALEF:

- *Domain conceptualization* – we use information contained in the domain model to extract fragments that explain key concepts of the document more accurately.
- *Knowledge of users* – using the modeled user knowledge, we can filter fragments that explain concepts that are too difficult for a user or those which a user already understands very well (depending on our scenario, whether we want to help users revise what they have already learned or help them find and comprehend concepts which are new for them).
- *User-added annotations* – when a user highlights a fragment of a text (by adding highlight annotation), we assume that the fragment contains information deemed important or interesting by the user; when many users highlight the same (or similar) fragment of text, we assume that the fragment contains important and valuable information in general.

We proposed a method of personalized text summarization based on a method of latent semantic analysis (Gong and Liu 2001; Steinberger and Ježek 2005). Our method consists of the following three steps (Móro and Bieliková 2012):

1. Pre-processing during which terms are extracted from the document and the document's text is segmented to sentences.
2. Construction of a terms-sentences matrix which represents an input to singular value decomposition (Gong and Liu 2001).
3. Selection of sentences; we select sentences with the highest score using approach proposed by Steinberger and Ježek (2005).

In order to adapt summaries we apply information from identified sources during a construction of a terms-sentences matrix, thus constructing a personalized terms-sentences matrix. Instead of a conventional weighting scheme based on tf-idf, we use our proposed weighting scheme based on a linear combination of multiple raters:

$$w(t_{ij}) = \sum_{k} \alpha_k R_k(t_{ij}) \tag{1}$$

where $w(t_{ij})$ is the weight of term t_{ij} in the matrix and α_k is the linear coefficient of rater R_k.

We designed a set of generic and personalized raters which positively or negatively affect the weight of each term. In order to produce baseline generic variants of summarization we designed *Terms frequency rater* and *Terms location rater,* which have been inspired by Luhn (1958) and Edmundson (1969) respectively. Our personalized raters take into account various sources of personalization and adaptation, i.e., *Relevant domain terms rater*, *Knowledge rater,* and *Annotations rater*. They determine which terms are important based on a source of information and assign increased weights to terms from selected sources.

Evaluation The personalized summarizer is integrated with ALEF using a summarization widget based on the existing widget infrastructure. We carried out two experiments on the *Functional and Logic Programming* course. In total, 17 students took part in the first experiment and 27 students in the second.

Students' task was to evaluate a presented summary on a five-point Likert scale. After each summary rating, students were asked follow-up questions to further evaluate quality of the summary (e.g., whether sentences selected for the summary were representative, whether the summary is suitable for revision etc.). Moreover, we selected a comparison group of five students who were presented both variants in random order to decide which variant is better or whether they are content equivalents.

In the first experiment we compared generic summarization to the summarization considering the relevant domain terms (Móro and Bieliková 2012). The summarization considering the relevant domain terms gained on average approximately 7.2 % higher score than the generic variant; it was also evaluated as better or equal by the experts in 69 % of the cases. In the second experiment we compared generic summarization to the summarization considering user-added annotations (Móro 2012). We got results similar to the previous experiment, when the experts evaluated the variant considering the user-added annotations as better in 48 % of the cases as opposed to the 24 % when it was considered as worse.

Our results suggest that considering the relevant domain terms as well as user-added annotations in the summarization process leads to better summaries compared to the generic variant and can be of higher value to the students in the learning process.

(4.3) Recommending Web Resources

ALEF contains a collection of learning objects available for students. However, great amount of quality resources are available on the Web. We were looking for the possibility to enrich content of ALEF by these resources. While reading a document, a student often encounters a word or a phrase, he/she may not understand or may require additional information to understand it sufficiently.

To provide more information about important parts of learning objects in ALEF, we proposed a method for automatically extending the content of learning objects by attaching annotations to selected terms in the text. Such annotations provide further explanations, links to related resources and other types of information retrieved using multiple publicly available services for information retrieval. The method is designed to be able to insert annotations into the text written in Slovak language with a potential to be language independent. It consists of three steps:

1. search for candidate words to attach annotations,
2. search for information to fill the annotations, and
3. adaptation and visualization of annotations.

To find locations to which it is appropriate to assign the annotation, various algorithms for keyword extraction or approaches from the field of natural language processing can be used. However, satisfactory results are currently achieved for English texts only. To overcome this problem it is possible to use machine translation to translate source text into English. Based on our experiments we believe that existing, although far from being perfect translation mechanisms are sufficient for this task, as we attach annotations mainly to nouns and verbs and these are translated correctly in most cases.

To solve the problem of linking extracted keywords from translated text to the original text, we proposed a method for mapping equivalent words between text translations based on a dictionary and comparing words using Levenshtein distance (Ševcech and Bieliková 2011). This method is the key element for annotation acquisition for various languages. We primarily consider Slovak language, which is an inflecting language with many various words forms and represents (considering its syntax) rather large group of languages.

Information for the annotations is retrieved from multiple publicly available services for information retrieval, where the query used to retrieve additional information consists of keywords extracted from the processed documents.

The final step of our method for automatic annotating the content of learning objects is the adaptation of the annotation content and the annotation visualization. For annotation adaptation we used implicit feedback from user interaction with annotations to sort annotation elements by their relevance for users. For finding the relevance of annotation elements we considered clicks on these elements as indication that an element is more relevant than other elements within the annotation. The clicks being the edges of a graph with vertices being elements, we apply PageRank algorithm to determine relevance of individual elements. Elements are then sorted according to this relevance. The annotation is visualized in a form of a tooltip that is displayed after clicking on a highlighted word within the text of the learning object.

Evaluation We evaluated automatic annotating within the *Principles of software engineering* course in ALEF. We attached annotations to keywords in every learning object of the course, and the order of the links to related resources in these annotations was adapted according to the implicit feedback created by students while studying materials of this course. The recommendation of web resources was evaluated in two steps: (1) the evaluation of the method for mapping equivalent words between text translations, and (2) the evaluation of the method for information retrieval from multiple sources (Ševcech and Bieliková 2011). Our method for mapping equivalent words while taking into account adjacency of words in sentences, and stemmed lexicon used in the process of mapping achieved precision at 92.46% with recall of 58.79% which gives F-measure of 71.88%.

Quality of added annotations heavily depends on the quality of a particular service for information retrieval. In our experiments we used Google Search, DBpedia, DictService and SlideShare. Relevancy of gathered information ranged from 70% for Google Search to only 26% for the SlideShare service. It can be improved mainly by adding personal context to the process of gathering information to fill the annotation, i.e. including information on users' interest to the query.

(5) Collaboration

Collaboration among students is an important element of learning. Support of effective and successful collaboration during the learning process represents an important concept in ALEF. Collaboration can occur in different forms. The types of collaboration can be divided according to various dimensions: according to the form of mediation (i.e., face-to-face vs. computer-mediated), according to students' perceptions ranging from implicit (indirect) to explicit (direct) collaboration, and according to the formality of education ranging from formal to informal collaboration. ALEF focuses on computer-mediated formal collaboration in learning, and provides support for both implicit and explicit collaboration.

(5.1) Implicit Collaboration

The process of annotating textual content represents an indirect form of collaboration. Students comment fragments of text for future reference, highlight important or interesting parts, report errors in text (factual or grammatical) etc. In doing so they do not help only themselves, they help other students as well: they can read comments inserted by others and respond to them, thus creating a form of discussion thread; they see which parts of texts were deemed important by their peers, can browse and navigate through the popular tags; and corrections made thanks to their error reports are beneficial to all of the students.

ALEF implements the annotation functionality within collaborative adaptive content creator components. The components are implemented using the aforementioned annotation and widget infrastructure. Each annotation widget introduces different goals for collaboration; it implements the whole lifecycle of an annotation type – creation of annotations, accessing (browsing) the annotations within the learning content, editing and optionally removing the annotations.

Tagger The *Tagger* (Móro et al. 2011) is a simple annotation widget that allows assigning user-defined tags to content (i.e., learning object as a whole); a tag can be a word or a multiword phrase. While the motivation behind tagging may differ among users, the result is usually the same: users add tags that describe the content of a learning object, or their opinion (e.g., *important, funny*), or intention (e.g., *todo*, *toread*). Users can assign private tags as well as public anonymous tags, and can navigate through their own set of tags or through popular tags. We encourage this kind of motivation (i.e., better navigation as a result of tagging) by letting students filter exercises and questions using tags accompanied by the autocomplete feature.

Besides providing additional style of navigation within a course, tags can be utilized for maintaining course metadata, as they represent a form of collaborative semantic descriptions and quickly converge into folksonomy – a vocabulary shared by the community (students within the particular course).

Highlights Highlights represent the simplest type of (per-text) annotation. The *Highlighter* aims to mimic common behavior of students when working with the printed text. They can simply select the desired part of text which they deem important or interesting, and choose to highlight it from the in-text pop-up menu without the need to insert any additional content of annotation.

Commentator Sometimes highlighting the text is not enough; a student would like to insert additional information to the selected fragment of text for future reference. For this purpose serves the *Commentator* (Šimko et al. 2011). Students can add private, public, or anonymous public comments to any part of any learning object. It also supports replying to other users' comments, thus resulting into discussion threads on arbitrary topics, typically related to misconceptions or learning problems.

Error reporter The *Error reporter* is a specialized version of the commentator widget (Šimko et al. 2011). It serves for reporting errors (factual or grammatical) found in the text by students. Reported errors are evaluated by a teacher resulting into improved content and thus better learning. This process supports collaboration between students and teacher or course documents maintainer.

External source linker Important feature of ALEF is to let students get involved into the process of learning by finding new potentially interesting and relevant sources of information. This provides the *External source linker* by enabling students to add links (URLs) to these sources (Mihál and Bieliková 2011). There are two ways to add a link: either as a per-text-annotation, using the in-text menu, or through the external source widget. The widget also serves as a means for accessing the inserted sources associated with the current learning object. The sources are displayed sorted according to their quality, with high quality sources (based on other users' ratings) on top of the widget.

Question creator Understanding of an educational material comes with the ability to explain it to others and pose questions about it. This is the motivation behind the *Question creator* widget, which provides students with an interface for creating questions and for answering questions created by their peers (Unčík and Bieliková 2010); thus students themselves become partly authors of the curricula. Students may add questions with five possible types of answers: (1) single choice question, (2) multiple choice question, (3) simple free text answer question, (4) sorting question (the task is to re-order the lines into correct order), and (5) text complement question where user is asked to fill missing words into dedicated fields within the text, e.g., completing missing commands in a program code.

Answers of peer students are automatically evaluated by the question creator and students can thus receive an instant feedback. Students can also rate questions in order to determine the question's perceived quality.

Evaluation We carried out multiple experiments evaluating separate parts of the annotation framework. We analyzed tags added by users and their capability in helping domain experts to create and refine the domain model; we found out that students were able, in a short period of time, to find almost half (49.8 %) of all the concepts in the domain model, while the domain model covered only 17 % of the tags added by the students (Móro et al. 2011). Similarly, we experimented with deriving new relations among learning objects and concepts in the domain model using the external sources added by students (Mihál and Bieliková 2011): the method identified concepts with 74.8 % precision and managed to find also new relations not present in the current domain model. These experiments confirm that annotations created by students can be used as a valuable source of information for domain model construction.

We evaluated the usefulness of our error reporting feature as well (Šimko et al. 2011): 20 % of the most advanced students provided 82 % of all error reports; altogether students found one error per 1.46 learning objects, thus managing to significantly increase the quality of the educational materials for other students.

In the experiment with the Question creator we assessed quality of student-generated questions (Unčík and Bieliková 2010). Results showed that 37.5 % of questions provided by students could have been used directly as new educational material with no need of teacher to intervene. Experiment also assessed automatic recognition of quality questions, which achieved accuracy of 70.1 %.

(5.2) Explicit Collaboration

According to Soller (2007) there are two approaches how to support effective collaboration in adaptive and personalized learning systems. The first approach is focused on collaboration at group level, namely how to support students to exchange information in the appropriate circumstances (group composition, context, level of detail etc.). The second approach is aimed to support collaboration at community level, i.e., how to share and discover common knowledge in online communities. In ALEF the focus is on supporting collaboration at group level, especially the *group formation* and *collaboration support*. Group formation is aimed to offer recommendations to students on how to create successful and effective groups, or alternatively, how to recommend peer help (another student which can participate on common collaboration). Collaboration support is aimed to support students during and after finishing collaboration. It is based on users' and groups' models which are compared with models of ideal collaboration. Based on the mentioned models we can provide students recommendations how to achieve more successful collaboration.

Group Formation

We consider group composition as one of the most important precondition of effective and successful collaboration. There are many existing methods which solve group formation problem such as the jigsaw method of Hinze et al. (2002), particle swarm optimization of Lin et al. (2010), and ontology based methods of Ounnas et al. (2008). They are not suitable for all domains and scenarios. For instance, they are static and do not consider student's actual context and are limited in employing different information sources about students. Also, the methods assume that it is possible to decide which aspects make collaboration really effective and successful. However, this has not been sufficiently determined by current research.

Improving upon the previous approaches to group formation we propose a method which automatically creates small short-term dynamic groups (Srba and Bieliková 2012). Our method can consider any personal or collaborative user's characteristics. Collaborative characteristics can describe students' behavior during collaboration process or relationships between students (Srba and Bieliková 2010).

Our method is inspired by the optimization approach called Group Technology (Selim et al. 1998). Group Technology approach is rooted in optimization in industry area and solves the problem how to effectively produce different parts by set of machines. This problem can be adapted to our educational domain, but we have students instead of machines and students' characteristics instead of parts.

The method is applied to the same set of students iteratively. The collaboration process is evaluated after each group finishes solving a particular task. This allows us to continually improve the understating of which characteristics should be combined together based on score representing evaluation of how effective and successful collaboration was achieved. In addition, it is possible to automatically determine students' collaborative characteristics.

Evaluation We evaluated the proposed method in two steps. First, we evaluated the preconditions of the method. Then we performed a long-term experiment where we compared the collaboration between groups created by our method and groups created by a reference method (we employed k-means clustering). 106 participants were iteratively assigned to 254 groups. The results of this experiment are displayed in the Table 3. Statistical significance testing yielded a p-value of 0.0048, producing statistically significant results.

Table 3 Group formation experiment results.

Groups created	Avg. score	Feedback	p-value		
			A	B	C
A. By the proposed method	0.459	4.01	N/A	0.0006	0.0071
B. By the reference method (k-means)	0.392	3.55	0.0006	N/A	0.0987
C. Randomly	0.422	3.29	0.0071	0.0987	N/A

Collaboration Support

The collaboration support in ALEF is based on a structured collaborative environment. Students can communicate by means of semi-structured discussion. It provides 18 different types of messages (e.g., propose better solution, accept proposal, ask for explanation, provide explanation). These different message types allow us to automatically identify student's activities.

Recorded activities are used to measure the collaboration by a set of seven dimensions the design of which is rooted in psychology studies: sustaining mutual understanding, information exchanges for problem solving, argumentation and reaching consensus, task and time management, sustaining commitment, shared task alignment and fluidity of collaboration (Burkhardt et al. 2009). We added one more dimension which represents teacher's evaluation of results correctness.

The computed value in each dimension of collaboration is presented to students after finishing each task. Students can self-reflect their collaboration and improve their activities in the subsequent collaboration. In addition, we provide recommendations for each dimension.

The collaborative environment consists of a set of tools which are suitable to solve collaborative tasks, which can be of several types (e.g., group discussion, list advantages/disadvantages, list pros/cons) while each type can be solved with one or more of the available tools: Text editor, Graphical editor, Categorizer, and Semi-structured discussion facility. These tools are available via ALEF's collaborative extension Popular Collaborative Platform (PopCorm). Students are able to ask for collaborative task assignment whenever during their individual study in ALEF. As soon as the collaborative task is assigned to them, they are able to solve this task in PopCorm besides searching for learning objects in ALEF (see Fig. 9).

Text Editor The text editor is an interactive tool which is suitable for collaborative writing of free text. It provides functionality for parallel editing of written text by several users at the same time together with conflict resolution in the case when two users edit the same part of the text. Basic text formatting is sufficient for our purpose.

Graphical Editor The graphical editor provides opportunity to collaborate visually by sketching. Its functionality includes drawing of vector shapes, importing raster images, adding text notes, etc.

Categorizer The categorizer is a tool used to solve tasks involving one or more lists of items. Students are able to dynamically create, edit and remove categories (lists) and their items. In addition, it is possible to rearrange the items in the particular category, and even to move items from one category to another. All these changes are synchronized in real time among all group's members.

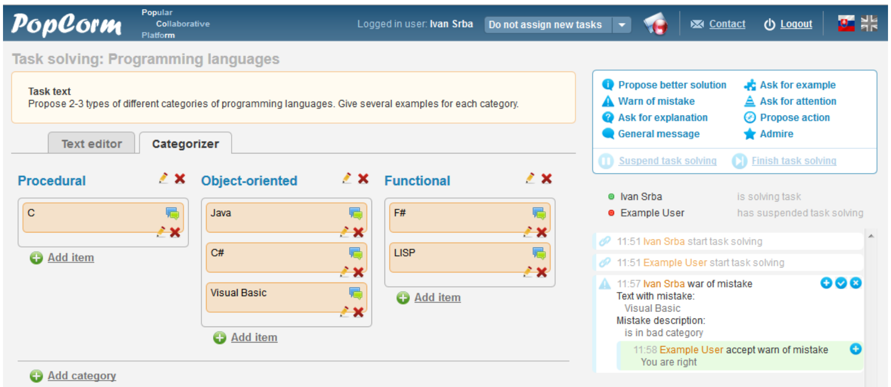

Fig. 9 Screenshot of ALEF's collaborative extension – PopCorm user interface. Categorizer tool is displayed on the left side. Semi-structured discussion is available on the right side.

Semi-structured discussion The semi-structured discussion facility represents a generic communication tool independent of a particular type of task being solved. Discussion is partially structured by employing sentence openers.

(6) Future Directions and Conclusions

Higher demand for web-based learning gave rise to an increase of students who learn online and the expanding amount of educational material available online poses new challenges to web-based educational systems. To meet the challenges and to leverage the trends established by Web 2.0, we devised adaptive learning framework ALEF. ALEF builds on three pillars: (1) extensible personalization and adaptation, (2) student active participation in learning process and collaboration, and the underlying (3) lightweight domain modeling.

Started as a small adaptive learning web application, ALEF became a platform for adaptive collaborative learning, where new methods supporting both education and education research can be easily created, covering wide range of applications with a particular focus on personalization and collaboration.

Modeling of students' knowledge using our logging and feedback infrastructure enabled us to deliver personalized recommendation of learning objects to our students and thus to make the learning process more efficient. Recommenders in ALEF not only ensure that a sequence of learning objects respects all known prerequisites, but also that a minimal required level of knowledge is attained by a student in a given time limit.

Text summarization methods enabled ALEF to recommend relevant fragments of texts to students (as opposed to whole learning objects). It seems promising to also use the summaries to help students revise their knowledge, and to navigate more efficiently in the learning object space.

Recommendation during learning with ALEF reaches beyond the learning content contained within the educational system. Web resources, are automatically linked to the course content, not only provide increased detail on topics and ensure the content is more up-to-date, but also put topics into broader context and thus contribute to overall comprehension of the domain by our learners.

Students' annotations of educational material proved to be useful on various levels: students get more involved into the learning process, quality of learning materials improve over time thanks to student generated error reports, links to external resources etc. and finally, student annotations bring novel insight to the model of the domain, revealing relationships, important parts etc. as well as to the student model, refining student's interest and knowledge. On the other hand, it opens up new research problems of automatic maintenance of annotations in the changing environment and filtering (recommendation) of those quality annotations, which can be helpful to others.

ALEF can furthermore take advantage of implicit collaboration among students when dealing with tasks related to authoring and assessment of particular domain model parts, which cannot be solved easily by computers, such as providing and validating free-text answers to quiz questions.

In contrast with individual learning, explicit collaboration facilitates practicing social and communication skills. As a result, students learn more efficiently and successfully. In fact, we observed improvement mainly for weak students. We believe that recommendation and collaboration together with smart scoring of students' activities are key issues that make ALEF successful.

There are many possibilities how to continue in the research in almost all mentioned areas. Considering recommendation, we still have not fully covered all important decision points where a recommendation can improve learning experience both individual and collaborative. For example, various types of learning objects (explanations, exercises, questions) can be better distinguished thereby the recommendation is tailored for particular learning objects. We incorporated into our recent study on ALEF logs also external sources of valuable information on students such as manual assessment outside ALEF or personality traits, which represent valuable sources for the recommendation.

We already started work on improving interconnection of ALEF with resources on the open Web. We plan to extend our external source linker by direct support of a student exploring the Web for additional learning resources. We proposed client-side use modeler BrUMo[2], which can be used for this task. In particular, we plan employing the BrUMo framework for improving recommendation outside ALEF by considering all the student knowledge already captured in ALEF. We realized BrUMo framework as a browser plugin. It provides low-level mechanisms to index and efficiently represent various user characteristics captured from visited web pages in an efficient manner on the client-side. BrUMo mechanisms are powerful enough to support both collaborative and content-based filtering approaches.

Another promising area is explicit collaboration of students such as providing collaborative support on the community level (in addition to small group level as outlined in this chapter), and improving methods for groups formation and task selection according to the individual user models or group models. Our current research on collaborative validation of question-answer learning objects is also along this line. We propose a method that utilizes students' correctness estimations of answers provided by other students. Our preliminary results show that student estimations are comparable to teacher's evaluations of provided answers.

Not less important than research results is the impact of ALEF for real-world education. ALEF is actively used at the Slovak University of Technology in Bratislava. Currently it supports studies in the three undergraduate courses in Informatics and Computer Engineering: *Functional and logic programming*, *Procedural programming*, and *Principles of software engineering*. Together it contains more than 1,875 learning objects and since summer term 2009/2010, it has successfully served more than 1,000 university students.

Acknowledgements This chapter is based on publications that present partial results of our research in the field of adaptive and intelligent web-based educational systems, all mentioned in References. It was partially supported by the grants

---

[2] Browser-Based User Modeling and Personalization Framework, brumo.fiit.stuba.sk

VEGA 1/0675/11/2011-2014, KEGA 028-025STU-4/2010, APVV-0208-10 and it is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

The authors wish to thank colleagues from the Institute of Informatics and Software Engineering and all students (in particular members of PeWe group, pewe.fiit.stuba.sk) for their invaluable contributions to the work presented in this chapter either by discussions or participating in ALEF implementation and experiments. Special thanks deserve former members of ALEF team Pavol Michlík, Vladimír Mihál and Maroš Unčík for their direct contribution to ALEF design and implementation.

References

Beaumont I, Brusilovsky P (1995) Adaptive educational hypermedia: From ideas to real systems. In: Maurer H (ed) Proc. of ED-MEDIA '95 – World Conf. on Educational Multimedia and Hypermedia, Graz, Austria, pp 93–98

Berners-Lee T (2005) Berners-Lee on the read/write web. http://news.bbc.co.uk/2/hi/technology/4132752.stm. Accessed 27 Jan 2013

Brusilovsky P (2004) KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. In: Proc. of the 13th Int. World Wide Web Conf., WWW'04. ACM Press, New York, pp 104–113

Brusilovsky P, Peylo C (2003) Adaptive and Intelligent Web-based Educational Systems. International Journal of Artificial Intelligence in Education 13:156–169

Burkhardt J, Détienne F, Hébert A, Perron L, Safin S, Leclercq P (2009) An approach to assess the quality of collaboration in technology-mediated design situations. In: Proc. of European Conf. on Cognitive Ergonomics: Designing beyond the Product (ECCE '09). VTT Technical Res. Centre of Finland, VTT, Finland

Cristea AI, de Mooij A (2003) LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators. In: Proc. The 12th Int. World Wide Web Conf. WWW'03. Alternate Track on Education, Budapest, Hungary

Cristea AI, Ghali F, Joy M (2011) Social, Personalized Lifelong Learning. In: E-Infrastructures and Tech. for Lifelong Learning (ETLL). IGI Global, p 90–125

Cimiano P (2006) Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer

Crestani F, Lee PL (2000) Searching the web by constrained spreading activation. In: Information Processing and Management, pp. 585–605

De Bra P, Houben GJ, Wu H (1999) AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In: Proc. of ACM Conf. on Hypertext and Hypermedia, pp 147–156

Downes S (2005) E-learning 2.0. ACM eLearn magazine 10:1

Edmundson, HP (1969) New methods in automatic extracting. J. of the ACM 16(2):264–285

Gong Y, Liu X (2001) Generic text summarization using relevance measure and latent semantic analysis. In: Proc. of the 24th ACM SIGIR, pp 19–25

Hinze U, Bischoff M, Blakowski G (2002) Jigsaw Method in the Context of CSCL. In: Proc. of World Conf. on Educational Multimedia, Hypermedia and Telecommunications, pp 789–794

IEEE (2002) LTSC Draft Standard for Learning Object Metadata. IEEE Standard 1484.12.1

Koch N, Wirsing M (2002) The Munich Reference Model for Adaptive Hypermedia Applications. In: Adaptive Hypermedia and Adaptive Web-Based Systems. Lecture Notes in Computer Science, vol 2347. Springer, Heidelberg, pp 213–222

Koper R, Rusman E, Sloep P (2005) Effective Learning Networks. Lifelong Learning in Europe 1, 18–27

Kostelník R, Bieliková M (2003) Web-Based Environment using Adapted Sequences of Programming Exercises. In: Proc. of Inf. Systems Implementation and Modelling, ISIM 2003. MARQ Ostrava, Brno, pp 33–40

Labaj M (2011) Web-Based Learning Support based on Implicit Feedback. Information Sciences and Technologies Bulletin of the ACM Slovakia 3(2):76–78

Lin Y, Huang Y, Cheng S (2010) An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. Comput Educ 55(4):1483–1493

Luhn HP (1958) The automatic creation of literature abstracts. IBM J. of Research Development 2(2):159–165

Manouselis N, Drachsler H, Vuorikari R, Hummel H, Koper R (2010) Recommender systems in Technology Enhanced Learning. In: Rokach L, Shapira B, Kantor P, Ricci F (eds) Recommender Systems Handbook: A Complete Guide for Research Scientists & Practitioners, Springer, pp 387–415

Meccawy M et al (2008) WHURLE 2.0: Adaptive Learning Meets Web 2.0. In: EC-TEL '08 Proc. of the 3rd European Conf. on Tech. Enhanced Learning: Times of Convergence. Lecture Notes in Computer Science, vol 5192. Springer, Heidelberg, pp 274–279

Michlík P, Bieliková M (2010) Exercises Recommending for Limited Time Learning. Procedia Computer Science 1(2):2821–2828

Mihál V, Bieliková M (2011) Domain Model Relations Discovering in Educational Texts based on User Created Annotations. In: Proc. of 14th Int. Conf. on Interactive Collaborative Learning and 11th Int. Conf. Virtual University. IEEE, Piscataway, pp 542–547

Móro R, Bieliková M (2012) Personalized Text Summarization Based on Important Terms Identification. In: Proc. of DEXA 2012 Workshops (TIR 2012), 23rd Int. Workshop on Database and Expert Systems Applications. IEEE Computer Society, Los Alamitos, pp 131–135

Móro R (2012) Combinations of Different Raters for Text Summatization. Information Sciences and Technologies Bulletin of the ACM Slovakia 4(2):56–58

Móro R, Srba I, Unčík M, Bieliková M, Šimko M (2011) Towards Collaborative Metadata Enrichment for Adaptive Web-Based Learning. In: Proc. of IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology – Workshops. IEEE Computer Society, Los Alamitos, pp 106–109

Mödritscher F (2010) Towards a recommender strategy for personal learning environments. Procedia Computer Science 1(2):2775–2782.

Ounnas A, Davis H, Millard D (2008) A Framework for Semantic Group Formation. In: Proc. of the 2008 Eighth IEEE Int. Conf. on Advanced Learning Technologies (ICALT '08). IEEE Computer Society, Washington, pp 34–38

Santos O C, Boticario J G (2011) Requirements for Semantic Educational Recommender Systems in Formal E-Learning Scenarios. Algorithms. 4(2):131–154

Selim HM, Askin RG, Vakharia AJ (1998) Cell formation in group technology: Review, evaluation and directions for future research. Computers & Industrial Engineering 34(1):3–20

Soller A (2007) Adaptive support for distributed collaboration. In: Brusilovsky P, Kobsa A, Nejdl W (eds) The adaptive web. Lecture Notes in Computer Science, vol 4321. Springer, Heidelberg, pp 573–595

Srba I, Bielikova M (2010) Tracing Strength of Relationships in Social Networks. In: 2010 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology. IEEE, pp 13–16

Srba I, Bieliková M (2012) Encouragement of Collaborative Learning Based on Dynamic Groups. In: Proc. of EC TEL 2012, 7th European Conf. on Technology Enhanced Learning. Lecture Notes in Computer Science, vol 7563. Springer, pp 432–437

Steinberger J, Ježek K (2005) Text summarization and singular value decomposition. In: Proc. Advances in Information Systems (ADVIS 04). Lecture Notes in Computer Science, vol 3261. Springer, Berlin, pp 245–254

Šaloun P, Velart Z, Nekula J (2013) Towards Automated Navigation over Multilingual Content. In Semantic Hyper/Multimedia Adaptation, Studies in Computational Intelligence, Volume 418, Springer, pp 203–229

Ševcech J, Bieliková M (2011) Automatic Annotation of Non-English Web Content. In: Proc. of IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology – Workshops. IEEE Computer Society, Los Alamitos, pp 281–283

Šimko M (2012) Automated Acquisition of Domain Model for Adaptive Collaborative Web-Based Learning. Information Sciences and Technologies Bulletin of the ACM Slovakia 4(2):1–9

Šimko M, Barla M, Bieliková M (2010) ALEF: A Framework for Adaptive Web-Based Learning 2.0. In: Proc. of IFIP Advances in Information and Communication Technology, vol 324/2010. Springer, pp 367–378

Šimko M, Barla M, Mihál V, Unčík M, Bieliková M (2011) Supporting Collaborative Web-Based Education via Annotations. In: ED-MEDIA 2011: Proc. of World Conf. on Educational Multimedia, Hypermedia & Telecommunications. AACE: Association for the Advancement of Computing in Education. pp 2576–2585

Šimko M, Bieliková M (2009a) Automated Educational Course Metadata Generation Based on Semantics Discovery. In: Proc. of European Conf. on Technology Enhanced Learning, EC TEL 2009, Nice, France. Lecture Notes in Computer Science, vol 5794. Springer, pp 99–105

Šimko M, Bieliková M (2009b) Automatic Concept Relationships Discovery for an Adaptive E-Course. In: Barnes T et al (eds) Proc. of Educational Data Mining 2009: 2nd Int. Conf. on Educational Data Mining, EDM 2009, Cordoba, Spain, pp 171–179

Šimko M, Bieliková M (2012) Discovering Hierarchical Relationships in Educational Content. In: Proc. of 11th Int. Conf. on Web-based Learning, ICWL 2012. Lecture notes in computer science, vol 7558. Springer, pp 132–141

Unčík M, Bieliková M (2010) Annotating Educational Content by Questions Created by Learners. In: Proc. of SMAP 2010. IEEE Signal Processing Society, pp 13–18

Vozár O, Bieliková M (2008) Adaptive Test Question Selection for Web-based Educational System. In: Proc. of SMAP 2008 – 3rd Int. Workshop on Semantic Media Adaptation and Personalization, Prague, CR. CS IEEE Press, pp 164–169