ONTOLOGY-BASED MODELS FOR PERSONALIZED E-LEARNING ENVIRONMENT

5th Int. Conference on

Emerging e-learning Technologies and Applications

The High Tatras, Slovakia

September 6-8, 2007

Michal Šimún, Anton Andrejko, Mária Bieliková
Slovak University of Technology, Faculty of Informatics and Information Technologies,
Ilkovičova 3, 842 16 Bratislava, Slovakia
simun.michal@yahoo.com, {andrejko, bielik}@fiit.stuba.sk

Abstract. Students who use educational web-based applications aim at learning knowledge in chosen area. However, their level of knowledge and interests are different. For effective learning it is necessary to provide an individual approach to each student. Educational application should recommend students learning materials that are easily understandable according their level of knowledge and are interesting enough to keep the students' attention. A solution is personalized approach to the student learning process by development of adaptive web-based applications. Personalization requires reasoning on knowledge related to each user and application domain. In this paper we describe a proposal of three main parts of adaptive web-based application – domain model, user model and adaptation model that we designed for the course of programming learning. We employ Semantic Web technologies in order to be able reuse existing educational materials and add a semantic layer responsible for personalization.

Keywords: concept, course, metadata, ontology, domain model, user model, adaptation model

1. INTRODUCTION

Many of the web-based content oriented applications offer mass of information to users with variety of interests, knowledge, and other personal characteristics. This causes users several problems [4]: users can get lost in information space, get information which they can not understand or it is not interesting for them (cognitive overload) or the user is suggested logically incoherent documents.

Described problems are more obvious in the e-learning applications, where students with the different level of knowledge, interests, and goals learn some educational material. A way to improve the effectiveness in information acquisition offers the personalized approach to the user provided in adaptive web-based applications. Adaptive web-based applications reflect some features of the user in the part called user model and employ this model to personalize various visible aspects of the system to the user [4]. To achieve good personalization we need to express structure and semantics of the knowledge representing by documents in the learning domain explicitly.

A part describing document's semantics is called domain model. User characteristics necessary for the personalization are retained in the user model. Adaptation model processes changing of user's characteristics on the basis of user's activities and it provides an adaptation of visible aspects of the system for user.

In this paper we propose models of the adaptive web-based system – domain model, user model and adaptation model. We use the Semantic Web principles for models representation and manipulation. Our models are based on ontological representation (OWL/RDF) that enables not only to represent meta-data but also reasoning in order to provide the best solution for each individual learner. The models are part of the educational portal that we have used also for other domains (job offers and publications). It is based on the layered architectural style of information systems. Here, we assign the domain and the user model into a data layer and the adaptation model into an application layer. Personalization is based on user's characteristics that are stored in the user model. Domain model represents semantics of educational documents.

2. RELATED WORK

The use of explicitly expressed semantics is not a brand new idea in the e-learning domain. There are applications that realize personalization and use ontology representation to capture semantics. The overview of e-learning trends in the context of the Semantic web is in [1]. We describe a few selected applications to point out main drawbacks of existing approaches.

System NetCoach [10] extends ELM-ART that is regarded as one of the first web-based adaptive applications. Domain model of NetCoach consists of fragments – explanation materials and tests questions, where documents presented to the student can have just one fragment. The author can create relations between fragments that express a *structure* of the course (by the *part-of* and *successor* relations), the *prerequisite* relation (fragments that are necessary to know for understanding actual fragment) and *inference* (fragments that are regarded as known when actual fragment is known). More views on user's knowledge are available, e.g. knowledge discovered from user's visit of the concept, user's tests and inference relation between fragments. Personalization is realized by link annotation and documents recommendation. The document is recommended when the user knows every prerequisite of document. Disadvantages are that the user model represents only user's knowledge for every fragment. The relevance or relation role expressing relation strength is not supported in the model.

Domain model of KBS Hyperbook [8] consists of two types of concepts – projects and information related to project. The goal of the recommendation in KBS is to select the most relevant project for a user and advise the user concepts related to this project. System selects the project on the basis of two criteria – project goal distance and fitness. Project goal distance expresses whether the concepts related to the project belong to the user's goal. Fitness expresses a relevance of the project on the basis of user's knowledge about concepts related to the project. In KBS it is impossible to model prerequisites between projects that are important in the recommendation process.

In [6] the educational course consists of a set of concepts. Each course has defined objectives and necessary resources. The set of concepts is ordered in the hierarchy using predefined concept's attributes *nextConcept*, *previousConcept*, *hasRequisite* and *isPrerequisiteFor*. The proposed model provides general concepts and uses class *Resource* to define semantics of the concepts. The relations between concepts do not have defined the strength or the relevance.

The domain model described in this paper is an extension of the ontology proposed in [5]. We extended, in particular, test questions, explanations and relations in domain model and changed its structure. The personalized approach is covered by the adaptation model that is responsible for the recommendation of learning documents. As the user is able to understand just some of knowledge provided by current educational applications and is not interested in every learning material, it is important to advise the user, which document is suitable for next studying. This is possible by representation of semantics of learning materials, interconnections between knowledge to be able reason appropriate sequence for particular user.

3. DOMAIN MODEL

Domain model represents semantics of educational materials. We divided domain model into two interconnected parts – *educational knowledge items space* and *concept space*. Knowledge item space consists of knowledge items – topics and keywords. For our domain of learning programming, the topics are chosen on the basis of the ACM Computing Classification System for Software (`www.acm.org/class/1998/D.html`). The second part of the domain model consists of concepts that represent characteristics of educational material. Concepts are assigned to knowledge items from knowledge item space by defined relations.

<u>Knowledge item space</u>

It represents a set of knowledge items that divide all educational courses into separated parts. Authors of a course define knowledge items and relations. Prerequisite relation between knowledge items determines which item(s) is necessary to know on defined level of knowledge (specified by relevance of the relation) to understand the others.

Prerequisites are connected using logical conjunctions AND or OR, which allows the author to express whether a student should know *one* (OR) or *every* prerequisite knowledge item (AND). An example of knowledge item space is shown in the Fig. 1. Knowledge items and prerequisite relations between them create acyclic directed graph.

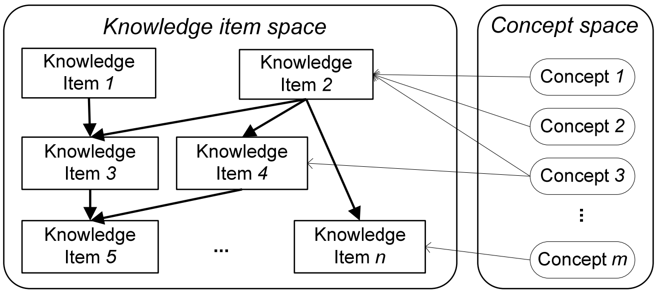

Fig. 1. Domain model

<u>Concept space</u>

Concept space describes characteristics of educational material that enable several alternatives of information fragments for the knowledge (e.g. difficulty level of the concept). Fragments are built into documents that are part of an educational course.

Every concept has general characteristics defined in the *Concept* class. Every concept has assigned *DomainAttribute* that represents properties of the concept in chosen domain. General characteristics of every attribute are defined in the *Attribute* class, which also contains relevance – *hasRelevanceAttribute*. We defined attributes specified programming language for the concept (*LanguageAttribute*), level of difficulty of the concept in the domain of learning programming (*DifficultyAttribute*). Every concept can have assigned text fragments, whose position in the concept is determined by the *IndexAttribute* attribute. Fig. 2 depicts proposed definition of the concept.

The educational courses are defined in the *Course* class and are represented in the hierarchical structure as a tree, where root of the course tree is assigned to the course by property *hasRootConcept*. Concepts consist of *IndexedFragment*, which represents logically separated part of the concept. Every fragment is connected to its assigned concept by concept's property *hasFragment*.

In our proposal we distinguish four types of programming concept's fragment – *Note*, *Solution* (represents solution of programming example), *Definition* (text defining a problem in the concept or explanation text) and *Hint* (gives the user an advice to solve the example). It was inspired by existing system for learning programming ALEA – predecessor of currently developed semantic web portal for e-learning [3].

The educational concept *ProgrammingConcept* (see Fig. 3) encapsulates all the types of the concepts necessary in the programming course. Every such a concept can have assigned definition fragments (*hasDefinition*) that occur in the educational document.

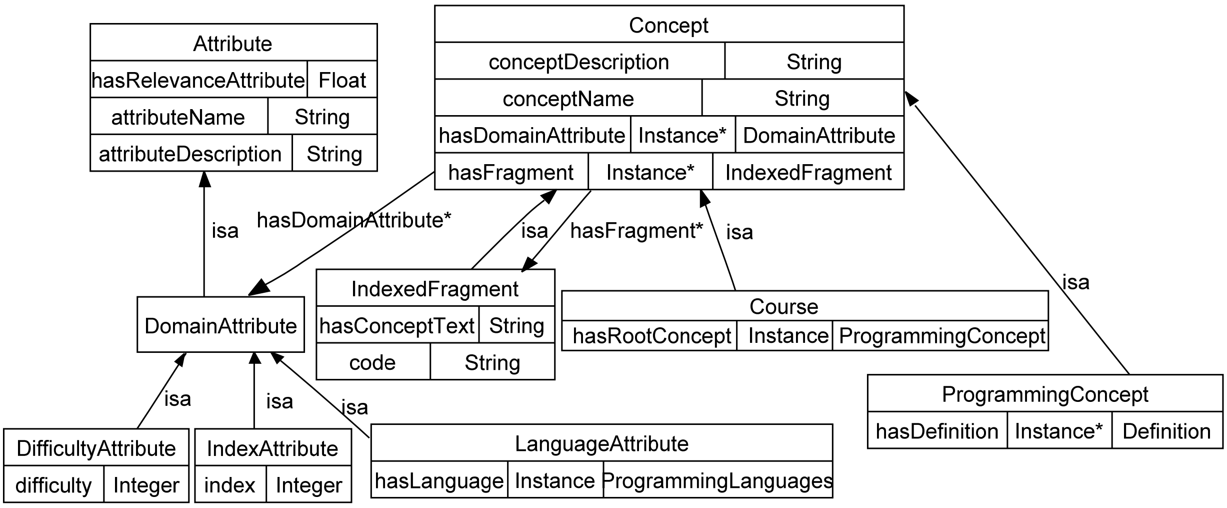

Fig. 2. Definition of the concept

We model following types of the *ProgrammingConcept*:

- *Exercise* – exercise in programming language according to the concept's *LanguageAttribute*. Exercise consists of the fragment's definition, hint, and solution. The exercises are grouped by *ProgrammingExercises* entity.
- *Explanation* – explanation of some problem.
- *Template* – general scheme of programming.
- *TestQuestion* – one test question used to examine student's knowledge. Test questions are grouped by entity *ProgrammingTest* and create test.

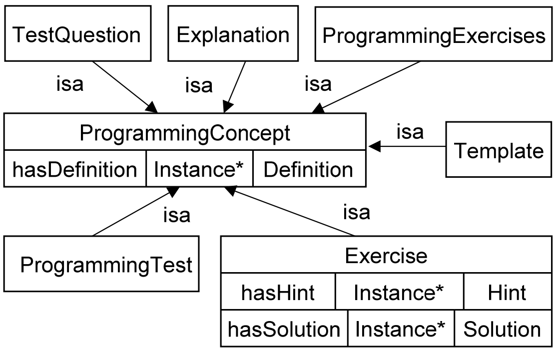

Fig. 3. Programming concept

Every defined concept is connected to knowledge items in the knowledge item space (Fig. 1) by *isRelatedTo* relation with assigned relation's relevance. The semantics of the relation depends on the type of the concept in the relation. Concepts can explain, exercise or test part of chosen knowledge item that is determined by the relevance of *isRelatedTo* relation.

4. USER MODEL

We proposed the user model and divided it into *general part* (domain independent) that can be common for user models in different domains and *domain specific part*. Both parts have assigned defined characteristics that allow keeping track on changes of the attributes – the count of attribute's updates, time of the last update, and identification of the source of change.

Domain specific user characteristics are inherited from *StudentAttribute* entity (see Fig. 4). Here, we distinguish three types of student's attributes, namely *StudentActivity* – activity while using the system, *StudentInterest* – estimation of user's interest for concepts and knowledge items, *StudentKnowledge* – student's knowledge of domain model parts. User model characteristics express relation between user model and domain model – defined concepts or knowledge items; therefore, our model belongs to overlay user models. Student's knowledge (interest) in domain model is defined as probability that student acquired some level of knowledge (interest).

5. ADAPTATION MODEL

Adaptation model is responsible for building and updating user model characteristics and for personalization of the application to the user on the other side [9]. In most educational applications recommendation is realized only on the basis of user's knowledge.

Concepts that do not fulfill all prerequisites are usually not recommended. Recommended concepts are determined on the basis of chosen metrics – user's knowledge, goal. We have proposed an improvement of this process dividing the process of concept selection for recommendation into two phases. First phase is preprocessing of information from the domain model for particular course. Afterwards, the most relevant document for the user is selected based on appropriate knowledge item selection and related concepts selection from the domain model.

<u>Preprocessing of domain model</u>
To be able recommend relevant document we need to find the most appropriate knowledge item and select related concepts. When students want to study selected knowledge item but they did not fulfill required prerequisite, the knowledge item preceding the prerequisite is recommended, including direct or indirect prerequisite. *Direct prerequisite* of knowledge item *A* involves knowledge items connected to *A* with direction to *A*.

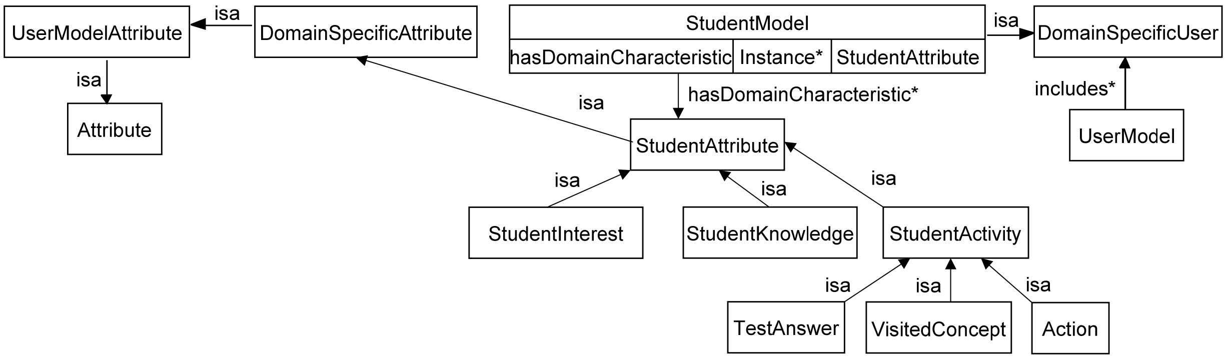

Fig. 4. Domain specific part of user model

Indirect prerequisites of knowledge item *A* consist of items connected to the direct or indirect prerequisite of *A*. On the other hand, in the recommendation it is necessary to select for item *B* all items, where *B* is direct or indirect prerequisite – direct or indirect use of *B*. Direct use of item *B* involves all items connected to item *B* with direction from *B*. Indirect use item *B* involves items connected to direct or indirect use of *B* (used items) with direction from used items.

For this purpose we transform the domain model into the layered model (see Fig. 5). On every layer occur knowledge items that have prerequisite from upper layers. When a knowledge item belongs to layer *n*, then the item has prerequisites just from layer 1 to layer *n-1*. Layer 1 consists of knowledge items that do not have any prerequisite from designed course. Every knowledge item has also related concepts from concept space. Described transformation is always possible, because condition of correctly designed knowledge item space is modeled as acyclic directed graph.

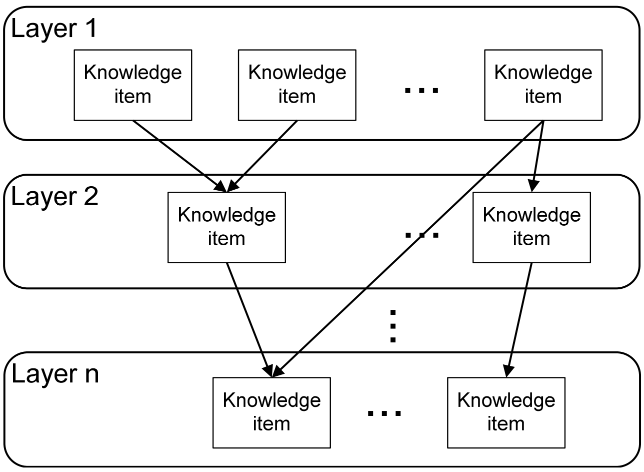

Fig. 5. Layered model of domain model

<u>Selecting relevant document</u>
The relevant document represents educational material that the user wants to read and also is able to understand. This process consists of two steps:

1. *Selecting a knowledge item* – the most appropriate knowledge from the knowledge item space are selected for the student.
2. *Selecting a concept* – concepts related to knowledge items (selected in the step 1) are ordered by relevance for the student.

In the first step we look for items, where all prerequisites are fulfilled and mostly correlate with user's preferences (attributes that define the relevance of knowledge item for the user). User's preferences are expressed by local and global preferences. Here multicriteria search of the best object in domain space methods can be used [7].

Defined attributes for knowledge items in our domain are *interest*, *knowledge* and *count of accesses* (read) of a knowledge item. Local preferences represent values of attributes used in chosen knowledge item and are explicitly expressed by fuzzy function. Global preference defines relevance of attributes and its combination in evaluation function of knowledge items. Evaluation of knowledge item is expressed by aggregation function.

An example of aggregation function, where variable *U* stands for the user and *KI* stands for the knowledge item:

$$\text{recommendationValue}(U, KI) = 0{,}5 * \text{interest}(U, KI) + 0{,}3 * \text{knowledge}(U, KI) + 0{,}2 * \text{access}(U, KI)$$

Fuzzy functions *interest*(*U*, *KI*), *knowledge*(*U*, *KI*) and *access*(*U*, *KI*) represent local preference of these attributes that express similarity between required value of attribute and value of this attribute stored in the user model.

We proposed the process of selecting relevant knowledge items as follows:

1. *Find Start item, i.e. from the knowledge item space, where searching will start.*
2. *Create Lists of relevant and actual items. Assign Start item to actual items.*
3. *<u>IF</u> List of actual items is empty*
   <u>THEN</u> Select most relevant items from List of relevant items and finish with recommending them.
4. *Take first knowledge item KI from actual items*
5. *<u>IF</u> Prerequisites of the KI item are not fulfilled*
   <u>THEN</u> Assign not fulfilled prerequisites to List of actual items and continue in step 3.
6. *Evaluate actual item by relevance value.*
7. *<u>IF</u> actual item can not be assigned to List of relevant items*
   <u>THEN</u> Continue in step 3
   <u>ELSE</u> Assign actual item to List of relevant items
8. *Gather items that directly use actual item and assign them to List of actual items. Continue in step 3.*

Knowledge items are selected from the layered domain model. Firstly the initial knowledge item should be chosen to start the process (Start Item). When the user had read some document from the concept space and it is regarded as interesting for the user, then the start item represents concept's related knowledge items. If none of related items are interesting for the user, then the start item represents the most interesting item in knowledge item space. If the user did not read any document, then knowledge item from first layer of layered model is chosen. Start items are assigned to the list of actual items. The *List* of actual items contains items, which relevance will be probed for the user. If the list is not empty, first item is removed and marked as actual item. The student is able to understand actual item only when prerequisites of item are fulfilled.

Prerequisites items are stored in the layered model in upper layers of actual item. If these prerequisites are not fulfilled, then prerequisites items are moved into the list of actual items. Otherwise, actual item is evaluated by relevance value that expresses relevance of the item for the user. Relevance value is expressed by equations as follows:

$$\text{relevanceValue (KI, SP, U)} = \text{relevance(KI, SP)} * \text{recommendationValue(KI, U)},$$

where

$$\text{relevance}(\text{KI}, \text{SP}) = \prod_{i=\text{SP}}^{\text{KI}-1} \text{relevance}(i, i+1)$$

In equations *KI* stands for knowledge item, whose relevance is explored, *SP* stands for start point that led to *KI* and *U* is the user. Relevance expresses importance of the knowledge item for start point and is acquired as multiplication of relations relevancies between start point and actual *KI*.

Actual item is assigned into the list of relevant items if relevance value is greater than defined threshold. The process of selecting relevant items will be moved to items that directly use actual item by inserting these items into list of actual items – these items are stored in layered model in lower layers. If relevance of actual item was already explored and its relevance value was greater, then the choosing process will not be moved to lower layers, because this path was already explored from other start point with greater relevance value. The process of choosing relevant knowledge item is finished, when the list of actual items is empty. Then the most relevant items from the list of relevant items will be chosen on the basis of item's relevance value.

In the second step we select concepts related to relevant *KI* chosen in first step. Every concept related to relevant *KI* is evaluated by relevance value. The application recommends documents that include the most relevant concepts. Relevance value of concept is expressed by equation:

$$\text{relevanceValue (C, KI, U)} = \text{relevance (C, KI)} * \text{recommendationValue (C, U)},$$

where *C* stands for the concept evaluated by the relevance value, *KI* represents chosen relevant knowledge item while *C* is related to *KI* and *U* is the user; *relevance*(*C*, *KI*) expresses the relevance of the relation between *C* and *KI* (relation between concept space and knowledge item space). Recommendation value is expressed the same way as knowledge items (by global and local preferences) and attributes of aggregation function in recommendation value consists of the count of access to the concept and user's preferred type of concept (test, explanation or exercise).

<u>Example: selecting relevant document</u>

Consider a part of domain model as shown in the Fig 6. The user *U* had read document described by the concept *A*. Afterwards, the next document should be recommended for her. Recommendation is processed as follows:

- the most relevant *KI* from the KIS is chosen;
- the concept assigned to chosen *KI* is selected;
- first, *start point* is selected (in the Fig. 6 it is *KI A*, since concept *A* is assigned to that *KI* and according to user model it is regarded as interested concept for user *U*);
- empty *list of relevant items (RI)* and *list of actual items (AI),* containing *KI A* starting point, are created;
- first item of *AI* is removed (item *A*) and its relevance for *U* is determined; prerequisites are *KI B*, *C* and *D*;
- since we assume (from the user model) that U does not know *KI B* and *C* on the required level, so *KI B* and *C* are inserted into the *AI;*
- first item from *AI* (*KI B*) is removed. *KI B* does not have a prerequisite, i.e. *U* can read it and its relevance is evaluated in the recommendation process.

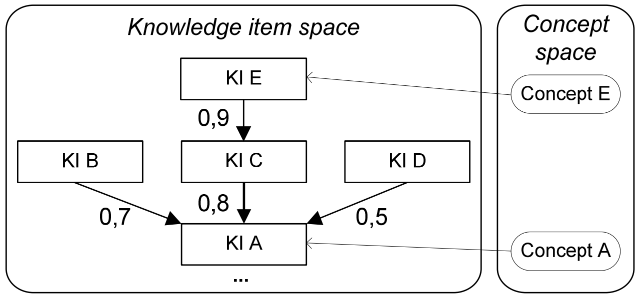

Fig. 6. Example of domain model

Relevance value consists of two parts. First, relevance of path between *KI B* and *KI A* (start point), i.e. 0,7 in Fig. 6. Second, recommendation value is assessed by aggregation function of user preferences for particular *KI*. In the example, the recommendation value for *U* is 0,8 according to the user model. The relevance value is defined as multiplication of this parts, i.e. 0,56. *KI D* is assigned into list of relevant items *RI* with relevance value 0,56. The process follows:

- first item from actual items *AI* (*KI C*) is chosen; prerequisite is *KI E*. Since *KI E* has not been fulfilled for *U*, it is assigned into *AI* and its relevance is explored in the next step.
- *KI E* has no prerequisite, i.e. user *U* is ready to study it.

Relevance of the path between E and starting point is 0,72 (0,8*0,9) and recommendation value is determined to 0,9. Thus, relevance value of KI E for *U* is assigned to 0,648 (0,72*0,9). KI E is inserted into *RI*; list of *AI* is empty, i.e. the most relevant item from *RI* is recommended (KI E).

In the second step of the recommendation process the most relevant concept assigned to the selected *KI* is chosen. Since there is only one concept assigned to KI E (i.e. concept E) this concept will be recommended to the user U.

6. CONCLUSIONS

In general, information systems can be designed according layered model with data, application and presentation layers. E-learning applications that ensure improvement of student's work by personalization should reflect each layer to reach this purpose. The main contribution of this paper is layered structure of the domain model for e-learning based on prerequisite relation and its interpretation in the adaptation model. The definition of semantics of presented documents (domain model) and characteristics of students (user model) are stored in the data layer.

We added to existing approaches explicit division of educational knowledge space into knowledge items space and concept space. Knowledge items space consists of topics occurring in the course. The concept expresses characteristics of educational documents in the course and each concept belongs to knowledge item(s). This enables reusability and various combinations of knowledge items characteristics according context of learning.

The model is reflected by novel method for concept recommendation, which is based on appropriate interpretation of the prerequisite relation which is crucial in learning application domain. First, in the recommendation process is chosen the knowledge item that can be considered as the most relevant from knowledge item space and has fulfilled all prerequisites. Afterwards, the relevant concept from the concept space that is related to selected knowledge item is chosen. Relevance of the knowledge items and concepts is estimated by aggregation function on the basis of user's characteristics – knowledge, interest, read documents.

We evaluate proposed approach on the adaptive portal framework developed for personalized applications based on ontological representation of models [2]. Learning programming is not the first application integrated within this portal, we already built two prototypes for application domain of personalized recommendation of job offers and publications. Our future work is oriented towards authoring proposed models, mainly automatic discovery of relations between the concepts.

7. REFERENCES

[1] Aroyo, L., Dicheva, D.: The New Challenges for E-learning: The Educational Sem. -Web. In Educational Technology & Society. Vol. 7 (2004) 59-69.

[2] Barla, M., Bartalos, P., Sivák, P., Szobi, K., Tvarožek, M., Filkorn, R.: Ontology as an Information Base for Domain Oriented Portal Solutions. In 15th Int. Conf. on Information Systems Development, Hungary (2006).

[3] Bieliková, M.: An adaptive web-based system for learning programming. Int. J. Continuing Engineering Education and Life-Long Learning, Inderscience, Vol. 16, Nos. 1/2 (2006) 122-136.

[4] Bieliková, M., Kuruc, J.: Sharing User Models for Adaptive Hypermedia Applications. In 5th Int. Conf. on Intelligent Systems Design and Applications, ISDA 2005. Wroclaw, Poland. ACM Press (2005) 506-511.

[5] Bieliková, M., Moravčík, M.: Modeling the content of adaptive web-based system using an ontology. In 1st Int. Workshop of Semantic Media Adaptation and Personalization, SMAP 2006. Athens, Greece. IEEE Computer Society Press (2006) 115-120.

[6] Gascuena, J. M., Fernandez-Caballero, A., Gonzalez, P.: Domain Ontology for Personalized E-Learning in Educational Systems. In ICALT 2006. IEEE Computer Press (2006) 456-458.

[7] Gurský, P., Vojtáš, P.: Multicriterial search of best object with support of several users. In Znalosti 2007. Ostrava, Czech republic (2007) 52-63 (in Slovak).

[8] Henze, N., Nejdl, W.: Adaptivity in the KBS Hyperbook System. In Int. Workshop on User Modeling and Adaptive Systems on the WWW, Toronto (1999).

[9] Henze, N., Nejdl, W.: A logical characterization of adaptive educational hypermedia. In New Review of Hypermedia 10 (2004) 515-530.

[10] Sosnovsky, S., Brusilovsky, P.: Layered Evaluation of Topic-Based Adaptation to Student Knowledge. In Workshop on Evaluation of Adaptive Systems at 10th Int. User Modeling Conference (2005) 49-65.

THE AUTHORS

Michal Šimún received his Bachelor degree in 2006 by the Slovak University of Technology. Currently he is a student of Software Engineering master degree programme at the Slovak University of Technology, Faculty of Informatics and Information Technologies. His research interests are in the area of the Semantic Web and personalization.

Anton Andrejko received his Master degree in 2004 by the Technical University in Košice. Currently he is a PhD student at the Slovak University of Technology, Faculty of Informatics and Information Technologies. He works in the area of user modeling and personalization.

Mária Bieliková received her Master degree (with summa cum laude) in 1989 and her Ph.D. degree in 1995 both by the Slovak University of Technology in Bratislava. Since 2005, she has been a full professor, presently at the Institute of Informatics and Software Engineering, Slovak University of Technology. Her research interests are in the areas of software knowledge engineering and web information systems, especially adaptive web-based systems including user modeling.

ACKNOWLEDGEMENT

This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 3/5187/07.