APPLYING COLLABORATIVE TAGGING AND SEMANTIC WEB TECHNOLOGIES FOR PROVIDING RECOMMENDATION IN PROGRAMMING TUTORING SYSTEM

Boban Vesin[1], Aleksandra Klašnja-Milićević[1], Mirjana Ivanović[2], Zoran Budimac[2]

[1]Higher School of Professional Business Studies, Novi Sad, Serbia
{vesinboban,aklasnja}@yahoo.com
[2]Faculty of Science, Department of Mathematics and Informatics, Novi Sad, Serbia
{mira,zjb}@dmi.uns.ac.rs

Abstract

The success of intelligent tutoring system depends on the retrieval of relevant learning material according to the learner's requirements. Therefore, the ultimate goal is development of the system that provides learning materials considering the requirements and understanding capability of the specific learner. In previous research, we implemented tutoring system named Protus 2.0 (PROgramming TUtoring System) that is used for learning basic concepts of Java programming language. It includes the use of an ontology and adaptation rules for knowledge representation. This paper presents the implementation of collaborative tagging technique for content recommendation in Protus 2.0. This technique can be applied in intelligent tutoring systemsfor providing tag-enabled recommendationsof concepts and resources. Learners can tag the content in a collaborative way. The presented system generates repository of entered tags and recommends tags and learning resourcesto specific learner. System also provides browsing capabilities through the ontology concepts and the tags repository.

Keywords- collaborative tagging, ontology, semantic web, Java tutoring system

1 INTRODUCTION

Personalized learning occurs when e-learning systems are designed according to educational experiences that fit the needs, goals, and interests of their learners. Ideally, recommender systems in e-learning environments should assist learners in finding relevant learning actions that perfectly match their profile, at the right time, in the right context, and in the right way, keep them motivated and enable them to complete their learning activities in an effective and efficient way [14]. Personalisation can be achieved using different recommendation techniques [13]. Collaborative filtering recommendation is one of the most successful recommendation techniques to date. However, collaborative filtering recommendation becomes less effective when users have multiple interests, because users have similar taste in one aspect may behave quite different in other aspects. Information got from social tagging websites not only tells what a user likes, but also why he or she likes it. Tagging represents an action of reflection, where the tagger sums up a series of thoughts into one or more summary tags, each of which stands on its own to describe some aspect of the resource based on the tagger's experiences and beliefs [2].

In our previous work, we presented intelligent web-based programming tutoring system – Protus (PRogramming TUtoring System) that implements semantic web technologies for personalisation. This system supports learners by recommending learning resources, online learning activities or optimal browsing pathways, based on their learning style, knowledge level and the browsing history of other learners with similar characteristics.To improve recommendation quality, metadata such as content information of items has typically been used as additional knowledge [7]. With the increasing popularity of the collaborative tagging systems, tags could be interesting and useful information to enhance algorithms for recommender systems. These systems could support learners by recommending tags and learning resources or online learning activities, based on their preferences.

In this paper, we propose modified semantic web architecture for a tag-based recommender system implemented in intelligent web-based programming tutoring system – Protus (PRogramming TUtoring System) that takes into account tags entered by the learner. We first analyse applicability of tag-based recommender systems to e-learning environments, then we present elements ofthe programming tutoring system that will generate repository of entered tags and provide further recommendations.

The rest of the paper is organized as follows. In the Section 2 appropriate related work is analysed and discussed. Section 3 describes the representation of components according to Semantic web technologies within proposed system that will be used to perform tag-based personalisation.Modified ontologies arepresented in Section 4. The implemented user interface that enables tag-based recommendation in Protus is described in Section 5. Section 6 concludes the paper and indicates directions of our further research.

2 RELATED WORK - APPLYING TAG-BASED RECOMMENDER SYSTEMS TO E-LEARNING ENVIRONMENTS

Recommender systems in e-learning environments utilize information about learners and learning activities and recommend items such as papers, web pages, courses, lessons and other learning objects that meet the pedagogical characteristics and interests of learners [5]. Suchrecommender systems could provide recommendations that are based on previous learners' activities or on the learning styles of the learners that are discovered from their navigation patterns. To design an effective recommender system in e-learning environments, it is important to understand specific learners' characteristics [5]: learning goal, prior knowledge, characteristics of learner, learner grouping, rated learning activities, learning paths and learning strategies.

According to these learners' characteristics, which serve as guidelines for framework design and platform implementation of good recommender systems in e-learning environment, we considered some collaborative tagging systems for extending capabilities of traditional recommendation method. The increasing number of users providing information about themselves through collaborative tagging activities caused the appearance of tag-based profiling approaches, which assume that users expose their preferences for certain contents through tag assignments. Tag-based recommender systems [9] analyze tags, discover preferences of a given user and provide suggestions for the user which items could be interesting. The main advantage of the tag-based recommenders is that user preferences and interests are expressed by used tags of the given person. Therefore, these recommenders provide more accurate and personalized recommendations.The innovation with respect to the e-learning environments lies in their ability to find appropriate content on the web, and capability to personalize and adjust this content based on the system's examination of its learners and the collected tags given by the learners and domain experts [11]. These systems also have ability to promote the learning performance of individual learners.

An innovative architecture for a recommender system dedicated to the e-learning environments ispresented in [6]. This system simultaneously takes advantage of collaborative taggingand concept maps. By mapping the tags and concepts completed by a learner, incomprehensible facts of his/her knowledge are identified. In the proposed algorithm the similarity of concept maps and tags beinglabelled by users are computed to achieve the best suggestion to learner. Authors described the architecture of an automatic recommendation system for learning environments that considers the profiles of the learners containing learner's tags and concept maps. Original algorithm for recommender systems that utilizes collaborativefiltering and uses the learner's tags and concept maps as its input has also been proposed.

Authors in [2]outline their experiences with applying collaborative tagging in e-learning systems to supplement more traditional metadata gathering approaches. This paper takes a broad look at tagging within e-learning.It first looks at the implications for tagging within thedomain through an analysis of tags students provided whenclassifying learning objects. Next, two e-learning systems and their interfaces for applying tagging are described. Both systems contain collaborative tagging features and emphasize applying tags within learning content.Both ofthese applications focus on contextualizing tagging through fine-grained annotation of learning content; the first in typical webpages, and the second in multimedia displays.

Online learning system – QSIA, an active recommender system for Questions Sharing and Interactive Assignmentsis described in [12]. It is designed to enhance knowledge sharing among learners. Authorslaid out some of the theoretical background for social, open-rating mechanisms in online learning systems. Authors also indicatedthat social recommendations are critical for the exploitation of the value associated with recommendation.The ability to support Semantic web technologies for tag-based recommendation is the main innovation of our system.

3 PERSONALISATION OF PROGRAMMING TUTORING SYSTEM USING TAG-BASED RECOMMENDER SYSTEMS

Protus is a tutoring system designed to help learners in learning essentials of programming languages. In spite of the fact that this system is designed and implemented as a general tutoring system for different programming languages, the first completely implemented and tested version was for an introductory Java programming course [9]. Java was chosen because it is a clear example of an object-oriented language and therefore suitable for teaching the concepts of object-orientation. The main purpose of the Protus system is to recommend useful and interesting materials to e-learners based on their different backgrounds, preferences, learning purposes and other meaningful attributes [8]. Protus system consists of five functional components: domain module, learner model, application module, adaptation module and session monitor.

The adaptation module provides personalization based on recommender systems. The proposed framework for building automatic recommendations is composed of three modules (Fig. 1).

3.1 A learner-system interaction module

A learner-system interaction module pre-processes data to build learner models. The data about learners' activities (like sequential patterns, visited pages, test results and grades earned) are collected within this module. The pages for learners' registration, theory sessions, tutorials, examples and tests are extended with background processing of the input data.

3.2 An off-line module

An off-line module uses learner models on the fly to recognize learners' goals and content profiles. The first time that learners use the Protus 2.0, system asks them to fill the questionnaire that contain the index of learning styles questions to calculate their own learning styles [15]. After appropriate learning style is determined for each learner, learning content is filtered, depending on the status of the course and learner's affiliation.

3.3 A recommendation engine

A recommendation engine produces a list of recommended resources and actions. This list contains resources that should be presented to learner and recommended options that should be annotated for current learner (communication, testing, navigation, etc.). The list of recommendations is sent to alter learner–system interaction within a new session. Recommendations cannot be made for the whole pool of learners, because even for learners with similar learning interests, their ability to solve a task can vary due to variations in their knowledge level. In our approach, we perform a data clustering technique as a first step to cluster learners based on their learning styles. These clusters are used to identify coherent choices of learning activities. Then, a recommendation list can be created according to the learners' and experts' tags for each generated cluster based on the user-centric tag model which produces more accurate recommendations then existing state-of-the-art algorithms [16]. To create a tag in Protus, the learner simply enters arbitrary keywords in the appropriate text field for the current resource.

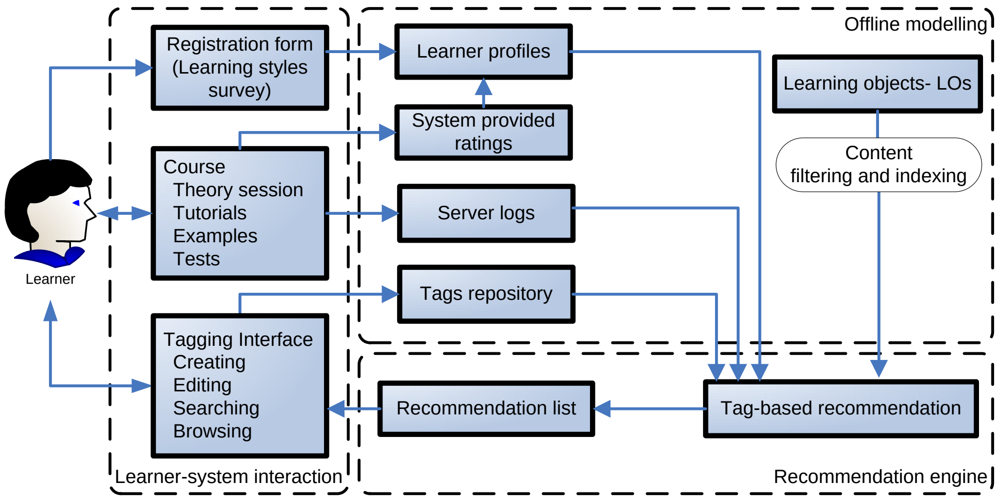

Fig. 1. The recommendation component of Protus system

To evaluate our system, we plan to carry out some experiments on an educational dataset on 1st year undergraduate learners. Involved learners will be programming beginners that successfully passed the basic computer literacy course at previous semester. The research will be focused on appropriate selection of collaborative tagging techniques that could lead to applying the best results in terms of increasing motivation in learning process and understanding of the learning content. Personalized and the most likely preferred recommendations can be estimated to an active learner as a result. This recommendation will be in accordance with the learner's interests, his/her learning style and previously acquired knowledge.

4 ADDAPTEDONTOLOGIES OF PROTUS

Complete ontology architecture of Protus 2.0 was described in our previous work [15]. In order to provide new functionalities to existing tutoring system in form of the tag-based recommendation, appropriate changes must be incorporated in *Task Ontology*, *Learner Model Ontology*and *Teaching Strategy Ontology*of the Protus 2.0.

4.1 Task Ontology

The complete Java course in Protus 2.0 contains several Concepts (lessons) [15]. Therefore, Java course contains: an introductory lesson, syntax, loop statements, execution control, etc. Each concept can be assigned any number of different resources (text files, images, animations, etc.).All resources are assigned depending on their resource type. Therefore, we have: theory, examples, assignments, exercises, syntax rules, etc. System should keep track of defined roles of every specific resource. For example, some resources represent the crucial information, while the others just represent a means to provide additional information or a comparison.

Task ontology is a vocabulary for describing problem solving structure of all existing types of domain independent tasks [4]. *Task Ontology*defines additional roles of each object within domain model and relations between them. This ontology does not describe the content taught by the learning material. Instead, each class of the ontology stands for a particular instructional role for a learning concept.

*Task Ontology*shows the role of specific resource from *Domain Ontology*. The role of the specific concept can be changed by the previous learners with their tags. Learners should be allowed to tag specific resource in order to present influence that specific resourcehad on them [10].

An excerpt of the modified *Task Ontology*of resources in Protus 2.0 is depicted in Fig.2.adapted to allow tagging. The ontology represents learning material grouped by the resources. The class *Concept* is used to presenta unit of knowledge that is represented by some *Resource***.**

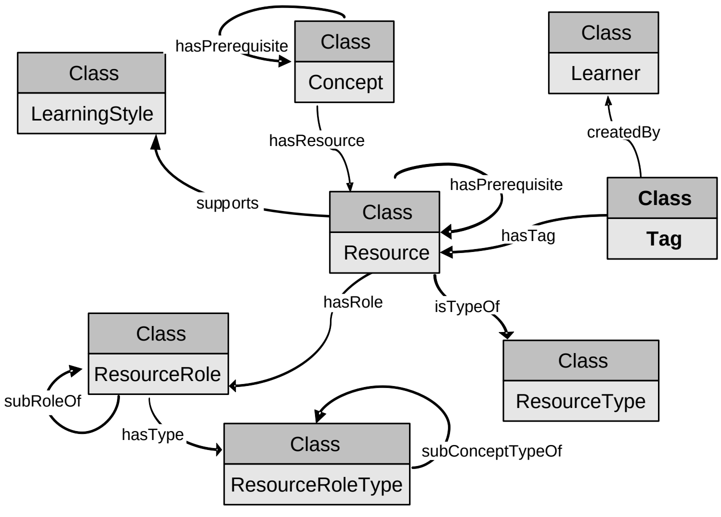

Fig. 2. An excerpt of Task Ontology of resources

Details about resources are kept in *Resource* class instances. Each instance of *Resource* class contains basic information on individual resources, which are used for the subsequent selection of appropriate resources in the process of personalization. Specific type and role is determined for every resource. List of tags for specific resources and their authors are kept in *Tag* class.

Like in [3], concepts and resources are related by the *hasResource* property. Concepts can be arranged by the *hasPrerequisite* property. The *hasPrerequisite* property is proposed for navigational purposes. It allows pointing out concepts that must be known before starting to study a concept, and the concepts for which it is a prerequisite. Concept will not be covered unless that the prerequisite condition is satisfied.

4.2 Ontology for Learner Observation

At run time, learner interacts with a tutoring system. These interactions can be used to draw conclusions about possible his/her interests, goals, tasks, knowledge, etc [15]. System must collect tags made by learners in order to generate further recommendation.*Ontology for Learner Observations*should therefore provide a structure that will keep track of learner interaction and generated tags. Fig. 3. depicts such modified ontology as a part of *Learner Model Ontology*. Data from *Learner Model Ontology*is maintained according to a class *Interaction*. *Interaction* is based on actions taken by specific learner, during specific learning session presented by class *Session*. *Interaction* implies a *Resource* learned from the experience, which is represented by *resourceUsed* property. In previous version of this ontology, class *Interaction*was connectedwith *Concept* class with *ConceptUsed* property. Changes were made due to necessity for adding resource-specific tags instead of concept-specific. This ontology is responsible for updating the *Learner Model Ontology*.

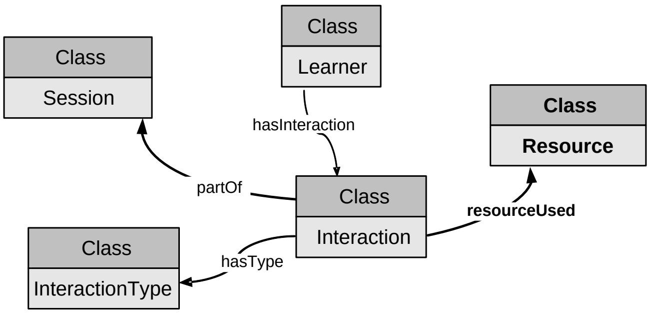

Fig. 3. Ontology for Learner Observation and Modelling in Protus

Personalization in e-learning systems includes authoring of learner models and applying different adaptation strategies and techniques to individual learners based on the data from these models [1]. The major goal of learning systems is to support a given pedagogical strategy [3]. In this scope, pedagogical ontologies can be associated with reasoning mechanisms and rules to enforce a given strategy. Often this strategy consists of selecting or computing a specific navigation sequenceof resources. Thus, formal semantics are required here to enable such computation.

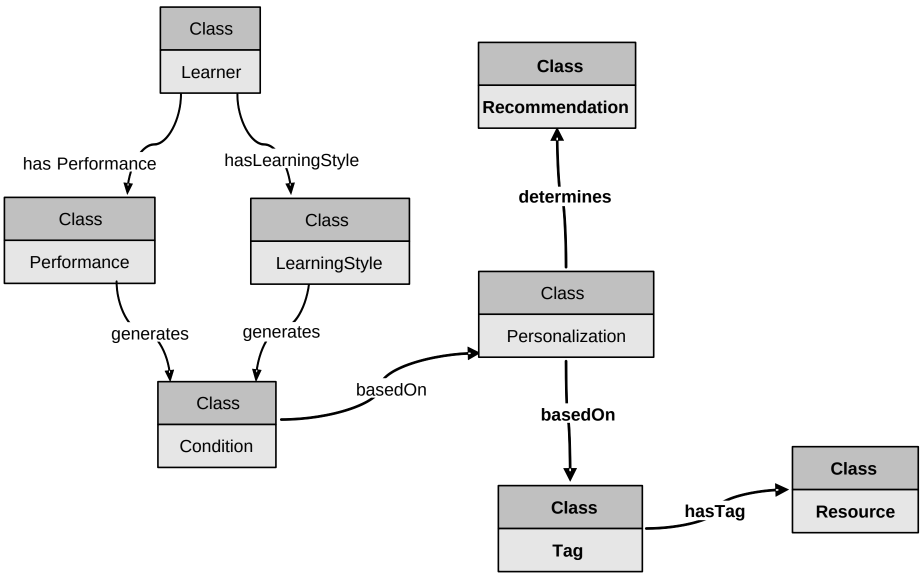

Fig. 4. Teaching Strategy Ontology of Protus

Fig. 4. shows how the adaptation is carried out by the modified *Teaching Strategy Ontology*. The decisions are drawn based on the information contained in the *Condition* class (that is generated by

the information about learning style and performance of the learner) as well as list of the most popular tags. Class *Tag* contains information about tags entered by learners. Therefore, personalization that is based on most popular tags generates list of recommended tags for that resource.

Personalization presents the choice of the most popular tag for specific resource that will be presented to the learner.

5 USER INTERFACE FOR TAGGING

Collaborative tagging activities caused the appearance of tag-based profiling approaches, which assume that users expose their preferences for certain contents through tag assignments. Thus, the tags could be interesting and useful information to enhance recommender system's algorithms. The innovation with respect to the e-learning system lies in their ability to support learners in their own learning path by recommending tags and learning items, and also their ability to promote the learning performance of individual learners.

Learners could benefit from writing tags in several important ways. Tagging is proven to be a meta-cognitive strategy that involves learners in active learning and engages them with more effectively in the learning process. Tags could help learners to remember better by highlighting the most significant part of a text, could encourage learners to think when they add more ideas to what they are reading, and could help learners to clarify and make sense of the learning content while they try to reshape the information. Learners' tags could create an important trail for other learners to follow by recording their thoughts about specific learning material and could give more comprehensible recommendation about the learning process. The viewing of tags used on a webpage can give a learner some idea of its importance and its content.

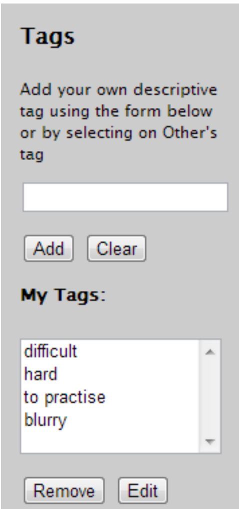

Fig. 5. Options for adding tags in Protus interface

Fig. 6. Options for reviewing tags in Protus interface

The information provided by tags makes available insight on learner's comprehension and activity, which is useful for both learners and teachers. Tagging, by its very nature, is a reflective practice which can give learners an opportunity to summarize new ideas, while receiving peer support through viewing other learners' tags/tag suggestions. Tagging provides possible solutions for learners' engagement in a number of different annotation activities - add comments, corrections, links, or shared discussion.

Modified ontology architecture of Protus provides possibilities for performing tag based recommendation to learners during learning sessions. Learner'sinterface, presented in our earlier work [15], is improved with elements for adding tags (Fig 5), overview of recommended tags (Fig 6.) and presenting other learner's tags for current resource (Fig 6.).

To create a tag in Protus the learner simply enters arbitrary keywords in the appropriate text field for an active resource. The system allows participants to enter as many tags as they wish, separated by commas. This makes it possible to specify several words for one tag separated by spaces, rather than restricting the participant to enter tag as a single word. This is in contrast to many popular tagging systems that only allow single word tags.

Whenever the learner returns to that particular resource, the list of tags he/she has previously made will re-appear, as shown in Fig5. Information provided by the individual learner is located under *My Tags* in the interface and they are ordered from the most to least frequently used tag. By clicking on each individual tag a list of three options pertaining to the tag are presented: *Search for this tag*, which links the learner to the search interface; *View learning objects you've tagged with this tag*, which shows all learning objects which have been described with the given tag; and *Edit this tag*, which give learner option to modify/delete this tag.By expanding the *Others' Tags* section, the list of active learners in community and the number of contributions are shown. Selection of one learner's tag section (see Fig. 6.) displays the most popular tags added by this learner in descending order of number of times used.

By expanding the *Recommended Tags* section (see Fig.6.), the most popular tags are shown in descending order of number of times used according to the overall use of tags, independent ofindividual learnerwhospecifiestags. This gives the learner an idea, at a much higher level, the overall view of all the content. The learner can get a sense of what are the most important terms and/or ideas at a course level. By clicking on one of the tags the learner can select to add the tag to the learning object in context, or to search by the given tag. As a result of searching, lesson and learning object for which the tag is placed will be displayed. For each tag, the number of occurrences is printed in brackets.

During learner-system interaction, details about active session, learner and current visited resource are entered in newly created instance of *Interaction* class in *Ontology for Learner Observation* (Fig. 3.). When learner enters specific tag for current resource, system creates appropriate class instance in *Task Ontology* (Fig. 2.). Based on the information from *Task Ontology* and *Ontology for Learner Observation,* system generates tag repository that is used for further recommendation. In the current version, system only recommends the most frequent tags to learners.

6 CONCLUSION

In this paper, we proposed modified semantic web architecture for a tag-based recommender system implemented in intelligent web-based programming tutoring system that takes into account tags entered by the learner. The form of several modified ontologies has been introduced which correspond to the components of a tutorial system: Task Ontology, Learner Model Ontology and Teaching Strategy Ontology. For generating presentation structures, examples of learner interface have been introduced.

This ontology-based approach allows adaptation of programming tutoring system to different requirements of the learners. An important part of future work will be the implementation of different and more complex technics for tag-based recommendation within Protus 2.0 system,as well asdefinition of appropriate SWRL adaptation rules that will make that recommendation possible.

This work contributes to research on personalization of programming tutoring system. Although this paper shows an application in programming tutoring system, considered approach can be applied in a variety of other learning domains.

References

[1] Aroyo, L., & Mizoguchi, R. (2003). Process-aware authoring of web-based educational systems.*International Workshop on Semantic Web and Web-based Education SWWL*, Velden, Austria, 212-221.

[2] Bateman, S., Brooks, C., McCalla, G., & Brusilovsky, P. (2007). Applying collaborative tagging to e-learning. *Proceedings of ACM WWW 3(4), May 2007*.

[3] Dehors, S., & Faron-Zucker, C. (2006). QBLS: A semantic web based learning system. *Proceedings of World Conference on Educational Multimedia*, Hypermedia and Telecommunications, 795-802.

[4] Devedžić, V. (2006). *Semantic Web and education*. New York: Springer Science.

[5] Drachsler, H., Hummel, H., & Koper, R. (2008). Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. *International Journal of Learning Technology 3 (4)*, 404-423.

[6] Kardan, A. A., Abbaspour, S., & Hendijanifard, F. (2009). A Hybrid Recommender System for E-learning Environments Based on Concept Maps and Collaborative Tagging. *Proceedings of the 4th International Conference on Virtual Learning ICVL,* 300-307.

[7] Karen, H., Tso-Sutter, L., Marinho, L., & Schmidt-Thieme, L. (2008). Tag-aware recommender systems by fusion of collaborative filtering algorithms.*SAC '08 Proceedings of the ACM symposium on Applied computing*, 1995-1999.

[8] Klašnja-Milićević, A., Vesin, B., Ivanović, M., & Budimac, Z. (2011). Integration of recommendations and adaptive hypermedia into java tutoring system.*Computer Science and Information Systems, 8(1),*211-224.

[9] Klašnja-Milićević, A., Vesin, B., Ivanović, M., &Budimac, Z. (2011). E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style identification, *Computers & Education, 56(3)*, 885-899.

[10] Klašnja-Milićević A., Vesin B., Ivanović M., &Budimac Z. (2012). Personalisation of Programming Tutoring System Using Tag-based Recommender Systems, *12th IEEE International Conference on Advanced Learning Technologies*, Rome, 666-667.

[11] Manouselis, N. & Costopoulou, C. (2007). Experimental analysis of design choices in a multi-attribute utility collaborative filtering system. *International Journal of Pattern Recognition and Artificial Intelligence 21*, 311–331.

[12] Rafaeli, s., Dan-Gur, Y. & Barak, M. (2005). Social Recommender Systems: Recommendations in Support of E-Learning. *International Journal of Distance Education Technologies*, 3(2), 30-47.

[13] Resnick, P., & Varian, H. (1997). Recommender systems. *Communications of the ACM 40(3)*. 56-58.

[14] Tang, T. & McCalla, G. (2005). Smart recommendation for an evolving e-learning system: architecture and experiment. *International Journal on e-Learning*, 4(1), 105 – 129.

[15] Vesin B., Ivanović M., Klašnja-Milićević A., & Budimac Z. (2012). Protus 2.0: Ontology-Based Semantic Recommendation in Programming Tutoring System. *Experts Systems with Application, 39*, 12229–12246.

[16] Wetzker, R., Zimmermann, C., & Bauckhage,C. (2010). I tag, you tag: Personomy translation for advanced user models.*Proceedings of International Conference on Web search and data mining*, 71-80.