# Integration of Recommendations into Java Tutoring system

Aleksandra Klašnja-Milićević[1], Boban Vesin[1], Mirjana Ivanović[2], Zoran Budimac[2]
[1]High Business school, Novi Sad
[2]Faculty of Science, Department of Mathematics and Informatics, Novi Sad
aklasnja@yahoo.com, bobanvesin@yahoo.com, {zjb,mira}@im.ns.ac.yu

## ABSTRACT

A challenging topic in web-based education is to provide personalization support for learners when e-Learning takes place in an open and dynamic environment. The personalization can be recognized as a functionality, which customizes access to learning services and learning resources, in the context of the delivery of a learning content, based on learner profiles. Recommender systems can be used to support e-Learning personalization by searching for useful and appropriate learning objects that will be presented to every particular learner. This paper presents an idea for integration of recommender system into existing web-based Java tutoring system in order to introduce intelligence in the system and to make it adaptive to individual learner's needs and interactions.

Key words: recommender systems, e-Learning, web-based adaptive educational systems, Semantic web

## 1. Introduction

E-Learning systems provide useful tools for computer-supported learning, such as forums, chat rooms, discussion groups and e-mail. However, most of them display content and educational material in the same way to all learners, allowing them to choose their own learning pathway through the course, which is not necessarily the most effective one in terms of their previous knowledge or needs. One possible solution to this problem is to use adaptive and intelligent web-based educational systems. These systems can use different recommendation techniques in order to suggest online learning activities or optimal browsing pathways to learners, based on their preferences, knowledge and the browsing history of other learners with similar characteristics. Their main objective is to adapt and personalize learning to the needs of each learner.

The task of delivering personalized content is often framed in terms of a recommendation task in which the system recommends items to an active user [1]. Recommender systems help users find and evaluate items of interest. Such systems have become powerful tools in many domains from electronic commerce to digital libraries and knowledge management [2]. Some recommender systems have also been applied to e-Learning systems for recommending lessons (learning objects or concepts) that learners should study next [3] or for providing course recommendation about courses offered that contribute to the learner's progress towards particular goals [4].

Recommender systems can use data mining techniques for making recommendations using knowledge learnt from the action and attributes of users [2]. The objective of data mining is to discover new, interesting and useful knowledge using a variety of techniques such as prediction, classification, clustering, association rule mining and sequential pattern discovery. Currently, there is an increasing interest in data mining and educational systems, making educational data mining a new and growing research community [5]. The data mining approach to personalization uses all the available information about learners on the web site (in the web course) in order to create user/learner models and to use these models for adaptation of content.

Another approach for course personalization is the use of *adaptive hypermedia* methods and techniques that are used in *Adaptive educational hypermedia systems* [6]. Adaptive educational hypermedia systems can adaptively sort, annotate, or partly hide the links on web pages to make it easier to choose or to recommend to the learners where they should go from a certain point based on his/her goals, preferences and level of knowledge. Learners can be informed about importance and relevance of certain links.

We propose an architecture framework for new web-based learning system that will provide adaptive courses using hypermedia and recommender systems, based on existing web-based Java tutoring system called Mag. To personalize the learning process for each individual leaner, our system needs to use strategies of both recommender systems and adaptive hypermedia.

The rest of this paper is organized as follows. Section 2 presents related work from the area of research. Section 3 explains the general proposed architecture and system design for combining recommender systems and adaptive hypermedia. Finally, in section 4, conclusions are drawn and future work considered.

## 2. Related Work

Two important ways of increasing the quality of service in e-Learning systems are to make them intelligent and adaptive [6]. These goals are likely to be achieved by different approaches. In some systems developers apply different forms of learner models to recommend the content and the links of hypermedia course pages to the learner. These systems are named *Adaptive educational hypermedia systems* [6]. Different approach use aggregation of the recommended materials from other e-Learning web sites and prediction of more suitable material to learners.

Romero et al. developed a personalized recommender system that uses web mining techniques for recommending a learner which (next) links to visit within an adaptable educational hypermedia system [7]. They presented a specific mining tool and a recommender engine that they have integrated in the AHA! system, in order to help the teacher to carry out the whole web mining process. They made several experiments with real data in order to show the suitability of using both clustering and sequential pattern mining algorithms together for discovering personalized recommendation links.

Another system implemented by Soonthornphisaj et al. [8] allows all learners to collaborate their expertise in order to predict the most suitable learning materials to each learner. This smart e-Learning system applies the collaborative filtering [1] approach that has an ability to predict the most suitable documents to the learner. All learners have the chance to introduce new material by uploading the documents to the server or pointing out the web link from the Internet and rate the currently available materials.

Most of the tutoring systems for learning programming languages found on the Web are more or less only well reformatted versions of lecture notes or textbooks. As a consequence these systems don’t have implemented interactivity and adaptivity. The functions that such systems can perform vary. Some of them are used for learner assessment like JavaBugs [9] and JITS [10,11], or some of them are adaptive web-based tutorials [7], [8]. One step further in implementation of adaptation was made by systems like JOSH-online [12] and CIMEL ITS [13],[14].

*JOSH* is an interpreter for the Java programming language [12] originally designed to ease teaching Java to beginners. Recently the interpreter was restructured into a server based interpreter applet and integrated into an online tutorial on Java programming called *JOSH-online.*

JavaBugs examines a complete Java program and identifies the most similar correct program to the learner’s solution

among a collection of correct solutions and builds trees of misconceptions using similarity measures and background knowledge [9]. They focused on the construction of a bug library for novice Java programmer errors which is a collection of commonly occurring errors and misconceptions.
Java Intelligent Tutoring System - JITS is a tutoring system designed for learning Java programming [10]. JITS implements JECA (*Java Error Correction Algorithm*), an algorithm for a compiler that error corrects by intelligently changing code, and identifies errors more clearly than other compilers. This practical compiler corrects errors by intelligently learning and changing source program code.
CIMEL ITS is an intelligent tutoring system that provides one-on-one tutoring to help beginners learn object-oriented analysis and design, using elements of UML before implementing any code [13]. It includes a three-layered *Learner Model* which supports adaptive tutoring by deducing the problem-specific knowledge state from learner solutions, the historical knowledge state of the learner and cognitive reasons about why the learner makes an error [14]. This *Learner Model* provides an accurate profile of a learner so that the ITS can support adaptive tutoring.
Systems like Logic-ITA [15] and Jeliot 3 [16] gave us good ideas and perspective which functionalities could be included in new web-based tutoring system. None of the above stated systems is complete web-based tutoring system with personalisation options. Some of them are not web-based and are only executed on stand alone machine (JavaBugs, JITS, CIMEL ITS, Jeliot 3) and some of them had just basic interactivity and adaptivity implemented (JOSH-online, JavaBugs, Logic-ITA). The new version of Mag system will integrate content and link adaptation in order to accomplish completely functional web-based tutoring system with personalization options.

## 3. Proposed Architecture and System Design

We propose an arhitecture for new web-based learning system that will provide adaptive courses using hypermedia and recommender systems. The architecture is an extension of existing web-based Java tutoring system called Mag [17]. To personalize the learning process for each individual leaner, our system needs to use strategies of both recommender systems and adaptive hypermedia.

### 3.1 Mag System

*Mag* is a tutoring system designed to help learners in learning programming languages in different courses [18]. It is an interactive system that allows learners to use teaching material prepared for programming languages within appropriate courses and to test acquired knowledge.
Mag is multifunctional educational system which fulfills three primary goals, identified by earlier exploration in this field [19]. The first goal is to provide intelligent tutoring system for learners in a platform independent manner. The second goal is to provide the teachers with useful reports identifying the strengths and weaknesses of learner's learning process. Finally, the third goal is to provide a rapid development tool for creating basic elements of tutoring system: tutorials and tests.
*Mag* supports learning by practicing and learning by samples. It combines traditional programming experience with distance education. System provides a learner with a more efficient and convenient way of taking a distance programming course. It provides three types of learning activity: tutoring, quiz-and-feedback, and on-line programming, to meet the needs of programming course.
In spite of fact that this system is designed and implemented as a general tutoring system for different programming languages, the first completely proposed and tested version was for introductory Java programming course.
Preliminary design of he *Mag* system was based on several basic system

requirements that every on-line learning system for a programming language should have [14]:

- separated user interfaces for learners and their mentors
- easy-to-access tutorials for learners
- various examples for every particular lesson (learning module)
- different tests for every particular lesson that can be adjusted to particular learner
- online programming, compiling and running of programs
- summaries and reports about learner's work
- functionalities for easy monitoring of learner's work
- functionalities for adding new lessons, examples, and tests possibilities for communication between learners and mentors.

Two main roles exist in the system, intended for two types of users:

- learners - they are taking the Java programming course and will be using the system in order to gain certain knowledge and
- mentors - their role is to be the lesson and learner database administrator, to track progress of learners learning and to help them with their assignments.

Therefore, two separated user interfaces are provided for both learner (learner) and instructor (learner's mentor). Instructor's interface helps in process of managing data about a learner and course material. Learner's interface is a series of web pages that provide two options: taking lessons and testing learner's knowledge. All data about learner and his progress in the course, as well as data about tutorials, tests and examples are stored in the system's server.

### 3.2 Adaptive learning and personalization of a content delivery

The ultimate goal for developing Mag system is increasing the learning opportunities and efficiency. Two important ways of increasing the quality of service of Mag are to make it intelligent and adaptive. Different techniques must be implemented to adapt content delivery to individual learners according to their learning characteristics, preferences, styles, and goals. In order to support adaptive learning and personalization of a content delivery, we must constantly measure the learner's knowledge and progress, build learner model and possibly redirect the course accordingly.

Three different levels of personalization (based on the levels of increasing abstraction and sophistication) must be included in the Mag system, which are suggested by Devedžić [6] and Martinez [20]:

- *Self-described personalization* – the learners will describe their preferences and common attributes with use of surveys or questionnaire, as well as identify their backgrounds and previous experiences. These create the initial learner models to start with in the instruction to follow.
- *Segmented personalization* - learners are grouped into smaller, identifiable and manageable clusters, based on their common attributes (e.g., class, age), preferences and surveys. Parts of the instruction are then tailored to the groups, and are applied in the same or similar way to all members of a segmented group.
- *Cognitive-based personalization* - adapting and delivering content and instruction to specific types of learners, defined according to information about their capabilities, and preferences. These may include, for example: a learner's preference for specific type of tests or tasks, or linear sequencing over grouping of hyperlinks, as well as recognition of the learner's reasoning capacity and capability for inductive reasoning. This type of personalization is more complex to implement than the previous types, as it requires collecting data, monitoring the learner's activity, comparing it to other learners' behavior, building a learner model and predicting and recommending what the learner would like to do or see next.

Our goal is to provide two general categories of personalization in system, based on previously mentioned levels of personalization:

- *Content adaptation* - presenting the content in different ways, according to the domain model and information from the user model. All learners and contents will be grouped into classes of similar objects in order to recommend optimum resources and pathways. The principle of clustering is maximizing the similarity inside an object group and minimizing the similarity between the object groups. Such clusters need to be defined in Mag system in order to provide learner with the most sutable learning material and to form the most sutable patheway.
- *Link adaptation* - the system modifies the appearance and/or availability of every link that appears on a course web page, in order to show the user whether the link leads to interesting new information, to new information the user is not ready for, or to a page that provides no new knowledge.

Tests in our system already consist of three types of questions [17]:

- *Multiple – choice of syntax*. This type of test is used to ask the learner to trace the correct sample code.
- *Multiple – choice of execution***.** This type of test is used to ask the learner to choose correct result after execution of offered code portion.
- *Code completion***.** Problem is presented in form of skeleton program with the specific area for entering appropriate code snippet according to program specification.

We need to add new categories of questions, tests, tasks and tutorials that will provide variety of testing and presenting possibilities to system. Also, learners must be categorized into clusters based on the preferable categories of content delivery. That categorization will be accomplished by different surveys (that the learner will be prompt to fill during the registration with the system and optionally after every lesson) and by monitoring the learner's actions, progress and overall performance. In this vein, we plan to track characteristics of the learner and collect a variety of useful information:

- information about the learner, including cognitive, affective and social characteristics,
- information about the learner's perspectives on the content itself, including the learner's feedback on the content, the learner's knowledge of the content (as determined, for example, by a test administered during the learner's interactions with system),
- information about the technical context of use, including characteristics of the learner's software and hardware environment,
- information about how the learner interacted with content, including observed metrics such as dwell time, number of learner keystrokes, patterns of access.

### 3.3 Proposed architecture of a hybrid system

The proposed architecture is concerned with producing previously mentioned types of personalization that will be implemented in the already existing learning environment and used by a large number of learners. Figure 1 shows a graphical representation of the proposed architecture. This architecture presents adapted architecture of first version of Mag [18] based on experiences from similar web-based learning systems [21],[22],[23] and an architecture for ontology-supported adaptive web-based education systems suggested by Devedžić [6] and De Bra et al [24].

This is essentially a centralized architecture. The core of the system includes the *adaptation model*, *learner model*, *application model* and *domain model*, all of them stored on a central server. *Domain model* presents storage for all essential concepts in the domain, tutorials and tests. Teachers (authors) can create *domain model* using appropriate authoring tool. The *adaptation module* uses different adaptation techniques, mentioned earlier to make the learning content presentation suit the learner's needs and current knowledge the best way possible.

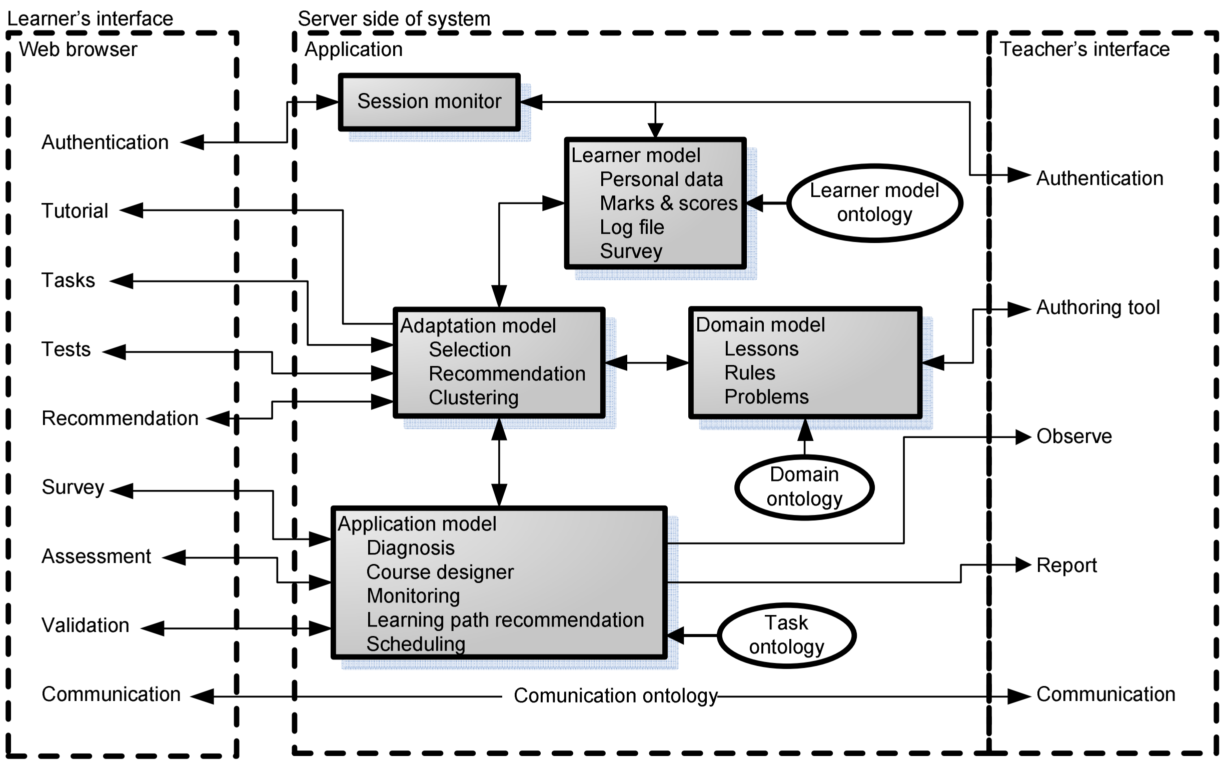

**Figure 1.** Proposed system architecture

Each *learner model* is a collection of both static (personal data, specific course objectives, etc.) and dynamic (marks, scores, time spent on specific lesson, etc.) data about the learner, as well as a representation of the learner's performance and learning history. The system will use that information in order to predict the learner's behavior, and thereby adapt to his/her individual needs.

Within *session monitor* component, the system gradually builds the *learner model* during the session, in order to keep track of the learner's actions and his/her progress, detect and correct his/her errors and possibly redirect the session accordingly. In the end of the session, the *learner model* is updated. It is then used along with other information and knowledge to initialize the next session with the same learner.

The *application model* executes the adaptation. To be exact, the *adaptation model* follows the instructional directions specified by the *application model*. These two components are separated in order to make adding new content clusters and adaptation functionalities easier.

Ontology engineering is a key aspect for the success of our web-based educational systems. Educational ontologies for different purposes must be included, such as presenting a domain (*domain ontologies*), building learner model (*learner model ontologies*) or presenting of activities in the system (*task ontologies*) [22]. A repository of ontologies must be built to achieve easier knowledge sharing and reuse, more effective learner modeling and easier extension of a system. Plan is to build ontologies based on the e-Learning standards defined in SCORM (*Sharable Content Object Reference Model*) [25].

## 4. Conclusion and Future Work

E-Learning can use different recommendation techniques in order to suggest the most appropriate online learning activities to learners, based on their preferences, knowledge and the browsing history of other learners with similar characteristics. The ultimate goal is improving the teaching and learning process by providing the learners with personalized course.

In this paper we presented the idea for integration of recommender system into existing web-based Java tutoring system in order to introduce intelligence in the system and to make it adaptive to individual learner's needs and interactions. The proposed architecture will contain elements of two different categories of e-Learning systems. First category is one that applies different forms of learner models to adapt the content and the links of hypermedia course pages to the learner, called *Adaptive educational hypermedia systems (AEHSs).* Recommender systems for classifying learners and contents in order to recommend optimum resources and pathways are the second one.

The existing system can be easily rearranged according to proposed architecture and it can be expanded with additional learning objects. Appropriate clustering of both learning material and learners must be executed in order to implement recommendation of material to specific types of learners. In this vein, we plan to track characteristics of the learner and collect a variety of useful information about the learner's perspectives on the content itself.

## References

[1] Mobasher, B., Data *Mining for Personalization. In The Adaptive Web: Methods and Strategies of Web Personalization*, Springer-Verlag, Berlin Heidelberg, 2007, pp. 1-46

[2] Schafer, J.B., *The application of data-mining to recommender systems,* Encyclopedia of data warehousing and mining, Hershey, PA. Idea Group, 2005, pp. 44-48

[3] Ksristofic, A., *Recommender System for Adaptive Hypermedia Applications,* In Proceeding of Informatics and Information Technology Student Research Conference, Bratislava, 2005, pp. 229-234

[4] Farzan, R., Brusilovsky, P., *Social Navigation Support in a Course Recommendation System,* In proceedings of 4th International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Dublin, 2006, pp. 91-100

[5] Romero, C., Ventura, S., *Educational Data Mining: a Survey from 1995 to 2005,* Expert Systems with Applications. Elsevier 1:33, 2007, pp. 135-146

[6] Devedžić V, *Education and the Semantic Web*, International Journal of Artificial Intelligence in Education 14, IOS Press, 2004, pp 39-65

[7] Romero C., Ventura S., Delgado J.A., De Bra P., *Personalized Links Recommendation Based on Data Mining in Adaptive Educational Hypermedia Systems,* Creating New Learning Experiences on a Global Scale (2007), pp. 292-306

[8] Soonthornphisaj N., Rojsattarat E., Yim-ngam S., *Smart E-Learning Using Recommender System*, ICIC Springer-Verlag Berlin Heidelberg 2006, pp 518-523

[9] Suarez M., Sison R. *Automatic Construction of a Bug Library for Objected-Oriented Novice Java Programmer Errors*, ITS 2008, Springer-Verlag Berlin Heideberg 2008, pp. 184-193

[10] Sykes E.R., Franek F., Presenting JECA: *A Java Error Correcting Algorithm for the Java Intelligent Tutoring System,* Advances in Computer Science and Technology - 2004, St. Thomas, US Virgin Islands 2004

[11] Sykes E.R., Franek F., *An Intelligent Tutoring System Prototype for Learning to Program Java*, The third IEEE International Conference on Advanced Learning Technologies (ICALT'03) Athens, Greece, 2003. pp 485-492

[12] Bieg C., Diehl S., *Education and tehnical design of a Web-based interactive tutorial on programming*

*in Java*, Science of Computer Programming, 2004, pp. 25-36

[13] Wei F., Moritz S.H., Parvez S.M., Blank G.D., *A Student Model for Object-Oriented Design and Programming*, Journal of Computing Sciences in Colleges, 2005, Pp. 260 - 273

[14] Glenn B., Shahida P., Wei F. Moritz S., *A Web-based ITS for OO Design*, 12th International Conference on Artificial Intelligence in Education, Amsterdam, 2004, pp. 59-64

[15] Merceron A., Yacef K., *A Web-Based Tutoring Tool with Mining Facilities to Improve Learning and Teaching*, Artificial Intelligence in Education, IOS Press, 2003. pp. 201-208

[16] Bednarik R., Moreno A., Myller N., *Program Visualization for Programming Education – Case of Jeliot 3*, Association for Computing Machinery New Zealand Bulletin, 2006, p.8

[17] Ivanović M., Pribela I., Vesin B., Budimac Z.**,** *Multifunctional Environment For E-Learning Purposes*, Novi Sad Journal of Mathematics, NSJOM Vol. 38, No. 2, 2008, pp. 153-170

[18] Vesin B. Ivanović M., Budimac Z., Pribela I., *Tutoring System for Distance Learning of Java Programming Language*. 10th Symposium on Programming Languages and Software Tools SPLST, Dobogókő, Hungary, 2007. pp. 310-320.

[19] Jones N., Macasek M., Walonoski J., Rasmussen K., Heffernan N., *Common Tutor Object Platform – an e-Learning Software Development Strategy*. Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, 2006. pp. 307-316.

[20] Martinez, M., *Designing learning objects to mass customize and personalize learning*, in: The Instructional Use of Learning Objects, D. A. Wiley, ed. 2000, Article 3.1, p. 29

[21] Chen C.M. *Ontology-based concept map for planning a personalized learning path*, British Journal of Educational Technology, Blackwell publishing, 2008 pp. 1-31

[22] Šimić G. *The Multi-courses Tutoring System Design*, ComSIS Vol. 1, No. 1, 2004 pp. 141-155

[23] Merino P.J.M., Kloos C.D., *An Architecture for Combining Semantic Web Tehniques with Intelligent Tutoring Systems*, ITS 2008, Springer-Verlag Berlin Heideberg 2008, pp. 540-550

[24] De Bra P., Aroyo L., Chepegin V., *The next big thing: adaptive Web-based systems*, Journal od Digital Information 5, Article No. 247, 2004, p.12

[25] SCORM - http://www.adlnet.gov/scorm/