Rule-based Reasoning for Altering Pattern Navigation in Programming Tutoring System

Boban Vesin, Mirjana Ivanović, Aleksandra Klašnja-Milićević and Zoran Budimac

***Abstract* - Semantic Web technologies seem to be a promising technological foundation for the next generation of e-learning systems. Although ontologies have a set of basic implicit reasoning mechanisms derived from the description logic which they are typically based on (such as classification, relations, instance checking, etc.), they need rules to make further inferences and to express relations that cannot be represented by ontological reasoning. In our previous work, we presented tutoring system named Protus (PRogramming TUtoring System) that is used for learning Java programming basics. One of the most important features of Protus is the adaptation of the presentation and navigation system of a course based on the level of particular learner knowledge. It uses principles of adaptive hypermedia and content recommendation for course personalization. There can be different sequence of resources that depends on navigational sequence determined for particular learner. In this paper we present set of proposed SWRL rules for altering navigation sequences.**

I. INTRODUCTION

Semantic Web technologies seem to be a promising technological foundation for the next generation of e-learning systems. Ontologies allow specifying formally and explicitly the concepts that appear in a concrete domain, their properties and their relationships [1]. Furthermore, they are useful in many environments: and especially in educational environments, as they enable people and/or software agents to share a common understanding of the knowledge structure. Moreover, ontologies permit to reuse knowledge. It is not necessary to develop ontology from scratch if another ontology is available for use in the modeling of the current domain.

Ontologies are typically based on the description logic which has a set of basic implicit reasoning mechanisms (such as classification, relations, instance checking, etc.). Nevertheless, they need additional rules to make further inferences and to express relations that cannot be represented by ontological reasoning [2]. Ontologies require a rule system to derive/use further information that cannot be captured by them, and rule systems require ontologies in order to have a shared definition of the concepts and relations mentioned in the rules. In our current research, we are going to exploit SWRL [3], a Semantic Web Rule Language combining OWL and RuleML. We use the SWRL as a reasoning and inference mechanism to obtain the essential production rules, as well as analyze the domain knowledge and interaction data. SWRL allows interoperability with major rules systems (commercial or noncommercial) such as JESS [2]. Rules defined in SWRL can describe the business logic. Then rule-based reasoning can be achieved to generate adaptation in e-learning systems.

The major goal of learning systems is to support a given pedagogical strategy [4]. In order to achieve specific pedagogical goals, ontologies can be associated with reasoning mechanisms and rules to enforce a given adaptation strategy in learning system. Often this strategy consists of selecting or computing a specific navigation sequence among the learning resources. Thus, formal semantics are required here to enable such computation. Adaptation presents a key feature for modeling educational environments capable to present learning objects in the adequate order based on the specific learning profile of each individual learner.

In our previous work, we presented tutoring system named Protus (PRogramming TUtoring System) that is used for learning Java programming basic concepts [5], [6]. One of most important features of Protus is the adaptation of the teaching material and navigation in a course based on the level of knowledge of a particular learner. Adaptation module of Protus uses principles of adaptive hypermedia and content recommendation for course personalization. Ontology architecture for tutoring system supported by several ontologies was proposed in our previous works as a way of addressing the problems of maintenance and reuse of system's components.

Original architecture of Protus was extended by the use of ontology, where the representation of each component is made by a specific ontology [7]. This way, it is possible to promote a clear separation of concerns of the tutoring system's components:

- to make explicit the communication among the components,
- to specify the support to build the components and
- to emphasize the gains of the use of Semantic Web in the development of tutoring systems.

That will also allow better interoperability and reusability of Protus components in the future.

This ontology-based approach allows us to implement efficient adaptation in Protus. The main objective of the

B. Vesin and A. Klašnja-Milićević are with the Higher School of Professional Business Studies, Novi Sad, Serbia (e-mail: {bobanvesin,aklasnja}@ yahoo.com).

M. Ivanović and Z. Budimac are with the Faculty of Science, Department of Mathematics and Informatics, Novi Sad, Serbia (phone: +381 21 458 888, e-mail: {mira.zjb}@ *dmi.uns.ac.rs*).

paper is to propose new features of Protus that will allow rule-based reasoning. The learner demand is derived from the knowledge contained in the ontology. Various conditions are captured in the body of SWRL rules. As a result of the firing of rules, recommendations in the form of adapted navigation sequence are generated. This can be used to implement the concept of adapted content and adapted navigation.

The rest of this paper is organized as follows: Section 2 describes the related literatures review. We present the Protus architecture and principles of the implemented adaptation in Section 3 and 4, respectively. This is followed by a description of the proposed rules in Section 5. Finally, we draw some conclusions in Section 6.

II. Related Work

Various approaches to adapting navigation and sequencing have been presented in numerous projects. The vast majority of existing tutoring systems can adapt navigation to only one kind of teaching operation. For example, a number of tutoring systems including [8], [9] can only manipulate the order of questions or problems during testing. This approach was usually called task sequencing. A number of authors made a step further and their systems can do sequencing of educational material [10], [11]. The most advanced systems are able to sequence several kinds of teaching operations such as presentation of the teaching material, examples, and assessments [12], [13].

A wide range of educational software that implements ontology-based components has been developed in recent period, but the most of these systems use ontologies only for representation of concepts, knowledge or learners data [1],[14]-[16].

Several studies show how semantic rules can be used in combination with ontologies for providing adaptation process in e-learning systems [17]-[20]. In the structure of previously mentioned systems, the use of ontology focuses mainly on learning objects and their related aspects. Besides, that does not facilitate the definition and communication between the other components of the system's architecture. Architecture of Protus supported by several ontologies was described in [7]. We further enhance that work by not only adapting the content modeling but we will show a way how to link semantics and content with implementing rule-based reasoning for adaptation process in Protus.

III. Protus Architecture

Protus is a tutoring system designed to support learning processes in different courses and domains but with intention to be used for learning programming languages. General ideas, of ontology-based architecture for Protus have been presented in [7]. Protus architecture is highly modular with five central components: *the application module, the adaptation module, the learner model, session monitor* and *domain module*. The general architecture of Protus is specially constructed to allow several types of adaptation (Figure 1).

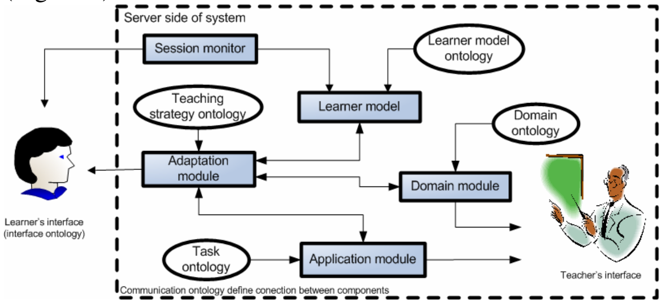

Fig. 1. Overall architecture of Protus

Domain module of Protus is assumed to be hierarchically structured into *Concepts* that are supported by the different *Resources* (Figure 2). There are several types of the *Resources* in Protus: tutorials, task, examples, tests, etc. A *hasPrerequisite* property is given by this hierarchical document structure and it is proposed for navigational purposes. It assigns the next document in sequence for each resource. It allows pointing out concepts that must be known before starting to study a concept, and the concepts for which it is a prerequisite. The concept will not be covered unless the prerequisite condition is satisfied. There can be different sequence of resources that depends on navigational sequence determined for particular learner.

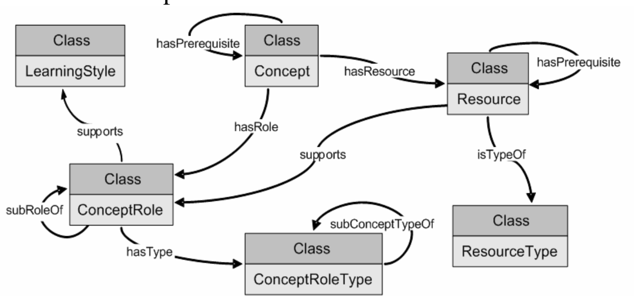

Fig. 2. An excerpt of domain ontology

IV. Resource Sequencing

According to [19], adaptation in tutorial systems can be done with regard to three levels:

- *Content Level Adaptation* by means of content hiding, additional explanations, different presentation methods, specific learning concepts filtering, adaptation to different student learning styles, etc.
- *Presentation Level Adaptation* by means of inserting/removing resources (learning objects), altering/sorting resources, adapting resources, etc.
- *Navigation Level Adaptation* by means of direct guidance, link ranking, link disabling/hiding, link annotation, link generation, etc. Often this strategy consists of selecting or computing specific navigation sequences among the resources.

Issues of content and presentation level adaptation implemented in Protus were subjects of our previous work

[5], [6], while focus of this paper will be navigational level adaptation.

Designers and instructors, while developing the on-line structure of the course and course material, have a navigation pattern in mind and assume all on-line learners would follow a consistent path. That path was determined in the design and materialized by some hyperlinks [21]. Learners, however, could follow different paths generating a variety of sequences of learning activities. Often this sequence is not the optimal sequence, and probably not the sequence intended by the designer.

Resource sequencing is a well-established technology in the field of intelligent tutoring systems [22]. The idea of resource sequencing is to generate a personalized course for each learner by dynamically selecting the most optimal teaching actions, presentation, examples, task or problems at any given moment. By optimal teaching action it is considered an operation that in the context of other available operations brings the learner closest to the ultimate learning goal. Most often the goal is to learn and acquire some knowledge up to a specific level in an optimal amount of time. However, it is easy to imagine other learning goals, such as minimizing student error rates in problem solving.

As stated in [23] the instructional contents' adaptation to the learner's individual characteristics implies that the system must:

- recognize the cognitive patterns of each learner and its pedagogical implications
- know the domain knowledge being proposed
- know the organization of domain material, and
- be able to dynamically generate the best suited selection and sequence for each learner in a particular course stage.

In contrast to traditional classroom-based learning, the learning behavior in web-based environments is more determined by the learner's own decisions how to organize learning process [24]. Learners could follow different paths based on their preferences and generate a variety of learning activities. All these variations in series of learning activities are noted down by the Protus system. In order to investigate learning activities in detail, sequential pattern mining algorithm of AprioriAll [25] is adopted to extract behavioral (interaction) patterns from the log file. These patterns will be used to analyze how learners evolve from the beginning of learning of particular unit, until they successfully finish it, or less successfully, give up. Learners with different learning habits and/or knowledge make different sets of frequent sequence. Hence, behavioral patterns were discovered for each learner by AprioriAll algorithm.

When the learner completes the sequence of learning materials, the Protus system evaluates the learner's acquired knowledge (Figure 3). The learners' ratings can be interpreted according to the percentage of correct answers. Two learners are said to be similar to each other if they are evaluated by the system with the same ratings for a similar navigational sequence. Recommendation process can be carried out according to these learning sequences based on the collaborative filtering approach that is described in our earlier work [6].

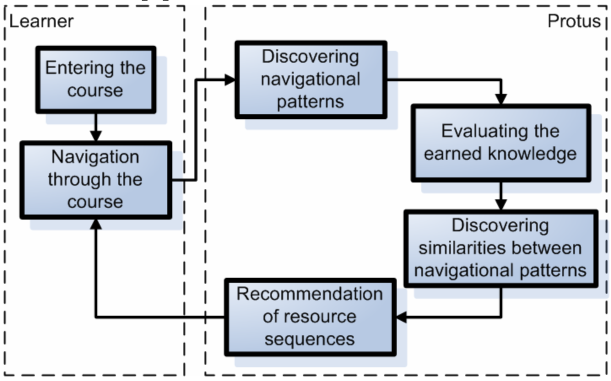

Fig. 3. Learning and recommendation process in Protus

There can be different a sequence of resources that depends on navigational sequence determined for particular learner. In next section we will present the set of proposed SWRL rules for altering navigation sequences. The benefit of representing teaching strategies as SWRL rules is that the strategies' computations would be explicitly represented in the ontology, and could be viewed and edited, as well as reasoned about by other applications.

V. Implemented Rules

SWRL rules are one of the most popular forms of knowledge representation, due to its simplicity, comprehensibility and expressive power [11]. There are different types of rules depending on the knowledge they store. They are referred to as: decision rules, association rules, classification rules, prediction rules, causal rules, optimization rules, etc. The types of rules we will use are the mostly decision and classification rules. Decision rules are used for choice of presented learning objects (resources) while classification rules helps in categorization of those resources.

In some cases, the interface elements for sequential navigation (in our case the buttons *Next* and *Previous*) are hidden, giving the learner the possibility to freely jump through the courseware. Furthermore, there will be added links to related or more complex content to help situate the learnt subject and contribute in creating clear overall view on the subject being thought.

Adaptation module of Protus used for building automatic recommendations will be composed of three groups of rules [6]:

- learner-system interaction rules
- off-line rules
- recommendation rules

The details of the whole process of recommendation and examples of implemented rules are presented in the rest of the paper.

Learner-system interaction rules pre-process data to build learner models. *Interaction* presents any kind of action performed by learner on specific learning object (page visiting, submitting test and tasks results, search and browse options, requesting additional examples, etc.). The data about learners' activities (like sequential patterns, pages that learner had visited, test results and grades earned) are collected with this rules. When the learner completes the sequence of learning materials, the system evaluates the learner's knowledge for each lesson [6]. Following rule updates learner model.

```
Learner(?x) ∧ Interaction(?y) ∧
hasInteraction(?x,?y) ∧ Concept(?c) ∧
conceptUsed(?y,?c) ∧ Performance(?p) ∧
hasResult(?y,?p) ∧ hasGrade(?p,?m) ∧
swrlb:greaterThan(?m, 1) → isLearned(?c,
true) ∧ hasPerformance(?x,?p)
```

With the previous rule, Protus is using recorded results of learner's interaction, earned grade and data about used concepts to memorize learner's performance in the session. Variables *x, y, c, m* and *p* present *Learner, Interaction, Concept, Grade* and *Performance,* respectively. Class *Performance* contains results of learner's interaction with specific concept (learning object): time, date, possible grade range, earned grade, etc. Meaning of the rule is: if in any time of the execution of Protus, there exists learner which interacts with specific concept, and during that interaction he/she took the test and earned specific grade, than system should memorize that learner's performance and mark that concept as learned. Rule is executed only if the learner earned a positive grade. The learners' grades have been interpreted according to the percentage of correct answers [6], as follows:

- 5 (excellent) - (80–100%)
- 4 (good) - (70–79%)
- 3 (average) - (60–69%)
- 2 (passing) - (50–59%)
- 1 (marginal) - (0–49%)

This five-point grading scale is based on our secondary school grading system. Consequently, learners have a better sense of having mastered the material using this system of evaluation. The system can be easily transformed and adapted to other standards of grading.

In the same time, system must update database of learners' interactions that is later used for grading of sequential patterns. Following rule is used for that matter:

```
Learner(?x) ∧ Concept(?c) ∧ Resource(?r) ∧
hasResource(?c,?r) ∧ isLearned(?c, true) ∧
hasPerformance(?x,?p) ∧ BehaviourPattern(?b)
∧ NavigationSequence(?n) ∧ isTypeOf(?b,?n) →
swrl:add(?n,?r,?p)
```

Previous rule adds visited resource to the navigation sequence in the current session. Built in function swrl:add(?x,?y,?z) adds resource y and details about performance z to navigation sequence x. Variables *x, r, c, b, p* and *n* presents *Learner, Resource, Concept, Behavior Pattern, Performance* and *Navigation Sequence,* respectively. Meaning of the rule is: if in any time of the execution of Protus, exists learner who successfully learned specific concept which is supported with specific resource, than system should add that resource and details about performance to the successful navigational pattern.

Off-line rules use learner models on-the-fly to recognize learners' goals and content profiles. Learners are grouped into clusters. These clusters are used to identify coherent choices in frequent sequences of learning activities [6]. Then, a recommendation list can be created according to the ratings of these frequent sequences provided by the Protus system. First, Protus distinguishes cluster that learner belongs to with one of the appropriate rules. Example of rule that is triggered if learner belongs to cluster *cl1* is (analog rules are used for other clusters):

```
Learner(?x) ∧ Performance(?p) ∧
hasPerformance(?x,?p) ∧ Condition(?c) ∧
generates(?p,?c) ∧ BehaviourPattern(?b) ∧
include(?p,?b) ∧ isTypeOf(?b,?n) ∧
NavigationSequence(?n) ∧ consistsOf(?n,a1) ∧
consistsOf(?n,b2) ∧ consistsOf(?n,c1) ∧
consistsOf(?n,d1) ∧ consistsOf(?n,e4)∧
swrlb:greatherThen(grade, required) →
belong(?x,?cl1)
```

This rule determines that learner belongs to the appropriate pattern, based on navigation sequence *n* that contains learning resources {a1, b2, c1, d1, e4}. This sequence is a list that contains: taken tutorial, visited examples, tasks and tests taken, etc. In this case, determined pattern is *cl1*.

Variables *x, p, c, b* and *n* present *Learner, Performance, Condition, Behavior Pattern,* and *Navigation Sequence,* respectively. *Condition* class collects data about learner's performance and his/her learning style and generates appropriate type of performed personalization that will be implemented. Generated personalization, in fact, presents specific navigational pattern recommended to learner. Meaning of the rule is: if in any time of the execution of Protus, exists learner which interacts with system under specific condition and during that interaction he/she successfully completed navigational sequence (predefined for specific behavior pattern), than that sequence can be treated as appropriate for that learner and he/she should be put in adequate cluster. Pattern discovering is only executed if learner successfully completes navigation sequence, that is to say, if learner has earned sufficient grade.

Recommendation rules produce a list or recommended learning objects. From the filtered list of learning content and based on the discovered sequences, the list of recommended actions and recourses is sent to alter learner–

system interaction within a new session. The recommendation module is design to create a recommendation list according to the ratings of these frequent sequences, provided by the Protus system [6]. For example, if the learner is determined to belong to cluster cl1, it means that he/she is attended sequence of resources: {a1, b2, c1, d1, e4}. Based on that initial sequence, Protus rated highest extended set of resources: {a1, b2, c1, d1, e4, f2} and, therefore, resource f2 is recommended to him/her with next rule:

```
Learner(?x) ∧ Performance(?p) ∧
hasPerformance(?x,?p) ∧ Condition(?c) ∧
generates(?p,?c) ∧ BehaviourPattern(?b) ∧
include(?p,?b) ∧ isTypeOf(?b,?n) ∧
NavigationSequence(?n) ∧ belong(?x,?cl1) →
isRecommended(f2,true)
```

Meaning of the previous rule is: if in any time of the execution of Protus, exists learner whom specific navigation sequence of resources has been recommended (with specific condition) than system should recommend to him/her next specific resource that belongs to that navigational sequence. Recommendation status of the resource f2 is set to true, therefore link to that resource is annotated or highlighted. Following of the recommended path does not influence the rules itself, but influence execution of rules in next sessions because ratings of frequent navigation sequences are calculated after every session. Ratings of frequent sequences are not calculated only by followed sequences itself but earned grades throughout session are also included in calculation. Therefore, every system-imposed path still counts towards placing the learner in a particular cluster.

VI. Conclusion

In the previous sections, both OWL and SWRL were used to demonstrate that the generic benefits promised by the Semantic Web can be useful to formally represent instructional models, and to let computers to automatically deduce new knowledge from existing facts and allow resource sequencing and adaptation of navigation patterns.

While the ontology represents the different adaptivity dimensions in terms of learner, domain, task, environment and system model, in this paper we tried to demonstrate that the logic underlying the adaptation can also be explicitly captured on the basis of a rule-based model. Our aim was to increase interoperability, reusability and extensibility by the use of SWRL rules.

In Protus, behavioral patterns are discovered for each learner by AprioriAll algorithm. Next, a recommendation list is created according to the ratings of the frequent sequences, provided by the Protus system. The provided recommendation is expected to have a higher accuracy in matching learners' requirements to learning material and thus a higher level of acceptance by the learners.

Mining the frequent sequences has the potential to improve the quality of an intelligent tutoring system, as well as keep the recommendation up-to-date. We have presented a set of adaptation rules that completes personalized recommendation of the learning content according to the ratings of these frequent sequences, provided by the Protus system.

The automatic recommendation is based on the intended sequence of navigation in the course material, or based on navigation patterns of other successful learners. This technique is used for recommending shortcuts or jumps to some resources to help learners better navigate the course materials.

As a result of the execution of rules, recommendations in the form of content presentation are generated, which can be used to implement the concept of adapted navigation. Our goal was to offer a simple concept for adaptation and personalization of both the content and the navigation to the user needs and goals. For the future work, we consider presenting how the system adapts to various learners in the actual circumstances.

References

[1] J. M. Gascueña, A. Fernández-Caballero, and P. González, "Domain ontology for personalized e-learning in educational systems, "*The Sixth IEEE International Conference on Advanced Learning Technologies,* 2006, pp. 456-458.

[2] F. Carmagnola, F. Cena, C. Gena, and I. Torre, "A semantic framework for adaptive web-based systems," *In Proceedings of SWAP 2005, the 2nd Italian Semantic Web Workshop, Trento, Italy*, 2005, p.14

[3] SWRL: A semantic Web rule language combining OWL and RuleML. Retrieved 19.12.08 from <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>

[4] S. Dehors, and C. Faron-Zucker, "QBLS: A semantic Web based learning system," *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications,* 2006, pp. 795-802.

[5] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, "Integration of recommendations and adaptive hypermedia into Java tutoring system," *Computer Science and Information Systems – ComSIS*, Volume 8, Number 1, 2011, pp. 211-224.

[6] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, Z. "E-learning personalization based on hybrid recommendation strategy and learning style identification," *Computers & Education 56,* 2011, pp. 885-899.

[7] B. Vesin, A. Klašnja-Milićević, M. Ivanović, and Z. Budimac, "Ontology-based architecture for providing recommendation in Java tutoring system," *The 6th International Conference on Information Technology ICIT 2011 Jordan*, 2011, to be published

[8] P. Brusilovsky, and J. Vassileva, "Course sequencing techniques for large-scale web-based education," *International Journal of Continuing Engineering Education and Lifelong Learning,* 13(1–2), 2003, pp. 75–94.

[9] A. Rios, E. Millarn, M. Trella, J. L. Perrez, and R. Conejo, "Internet based evaluation system," *In S. P. Lajoie and M. Vivet (Eds.), Artificial intelligence in education: open learning environments. Amsterdam: IOS Press*. 1999, pp. 387–394

[10] P. Brusilovsky, "ILEARN: an intelligent system for teaching and learning about UNIX," *In Proceedings of SUUG international open systems conference. Moscow: ICSTI,* 1994, pp. 35–41.

[11] C. Romero, S. Ventura, C. Hervas and P.Gonzales, „Rule mining with GBGP to improve web-based adaptive educational systems", *Data mining in e-learning, wit press*, 2006, pp. 173-190.

[12] M. Chen, M. H. Chang, C. Y. Liu, and W. C. Chiu, "Personalized learning path generation based on real-coded genetic algorithm for

web-based instruction," *In Proceedings of the 16th international conference of information management (ICIM'05*), Taiwan, 2005, p. 56.

[13] E. Mor, J. Minguillon, and J.M. Carbo, „Analysis of user navigational behaviour for e-learning personalisation," *Data mining in e-learning, wit press*, 2006, pp. 227-246.

[14] C. Hee Lee, J. Hyun Seu, and M.W. Evens, "Building an ontology for CIRCSIM-tutor," *13th Midwest AI and Cognitive Science Society Conference,* Chicago, 2002, pp. 161–168.

[15] R. Mizoguchi, Y. Hayashi, and J. Bourdeau, "Inside theory-aware and standards-compliant authoring system," *SWEL Workshop of Ontologies and Semantic Web Services for IES*, AIED, 2007, pp. 1-18.

[16] P. Dolog, and W. Nejdl, "Semantic Web technologies for the adaptive web," *The Adaptive Web, LNCS 4321, Springer-Verlag Berlin Heidelberg,* 2007, pp. 697–719.

[17] M.A. Sicilia, M.D. Lytras, S. Sances-Alonso, Garcia-Barriocanal E. and Zapata-Ros M., "Modeling instructional-design theories with ontologies: Using methods to check, generate and search learning designs," *Computers in Human Behavior,* 2010, doi:10.1016/j.chb.2010.07.040

[18] Y.L. Chi, "Ontology-based curriculum content sequencing system with semantic rules," *Expert Systems with Applications 36,* 2009, pp. 7838–7847.

[19] E. Popescu, C. Badica and P. Trigano, "Rules for learner modeling and adaptation povisioning in an educational hypermedia system," *Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2007, pp. 492-499.

[20] N. Henze, and W. Nejdl, "A logical characterization of adaptive educational hypermedia", *Hypermedia - Special issue: Adaptive hypermedia in the age of the adaptive web Volume 10 Issue 1,* 2004, pp. 77 – 113.

[21] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, "Integration of recommendations into Java tutoring system," *The 4th International Conference on Information Technology ICIT 2009 Jordan*, 2009, Paper no 154.

[22] M.J. Huang, H.S. Huang, and M. Y. Chen, "Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach," *Expert Systems with Applications 33*, 2007, pp. 551–564.

[23] M. A. M. Souto, et al., "Web-adaptive training system based on cognitive student style," *IFIP World Computer Congress, 17th Edition, Montréal, Canada*, 2002, pp. 281 – 288.

[24] P. A. Northrup, "Framework for designing interactivity into web-based instruction," *Educational Technology, 41(2),* 2001, pp. 31–39.

[25] W. Tong, and H. Pi-Lian, "Web log mining by an improved Aprioriall algorithm," *In: Proceeding of world academy of science, engineering, and technology,* 2005, pp. 97–100.