Rule-Based Reasoning for Building Learner Model in Programming Tutoring System

Boban Vesin[1], Mirjana Ivanović[2],
Aleksandra Klašnja-Milićević[1], and 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. Semantic Web provides huge potential and opportunities for developing the next generation of e-learning systems. Although ontologies have a set of basic implicit reasoning mechanisms derived from the description logic, they need rules to make further inferences and to express relations that cannot be represented by ontological reasoning. We implemented an adaptive and intelligent web-based PRogramming TUtoring System – Protus. One of the most important features of Protus is the adaptation of the presentation and navigation of a course material based on particular learner knowledge. This system aims at automatically guiding the learner's activities and recommend relevant actions during the learning process. This paper describes the functionality, structure and implementation of a learner model used in Protus as well as syntax of SWRL rules implemented for on-the-fly update of learner model ontology.

Keywords: Semantic Web, rules, SWRL, tutoring system.

1 Introduction

Semantic Web seems to be a promising technological foundation for the next generation of e-learning systems [1]. The main objective of the Semantic Web [2] is to describe Web resources in a way that allows machines to understand and process them. Ontology, generally defined as a representation of a shared conceptualization of a particular domain, is a major component of the Semantic Web. The initial work on implementing ontologies as the backbone of e-learning systems is presented in [3]. Since that time, many authors have proposed the usage of ontologies in different aspects of e-learning [4]. Ontologies allow specifying formally and explicitly the concepts that appear in a concrete domain, their properties and relationships [5]. Furthermore, they are especially useful in educational environments, as they enable people and/or software agents to share a common understanding of the knowledge structure.

Although ontologies have a set of basic implicit reasoning mechanisms derived from the description logic which they are typically based on, they need rules to make further inferences and to express relations that cannot be represented by ontological

H. Leung et al. (Eds.): ICWL 2011, LNCS 7048, pp. 154–163, 2011.

reasoning [6]. Thus, 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. Rules also add expressiveness to the representation formalism, reasoning on the instances, and they can be orthogonal to the description logic [7]. Moreover, rules can be defined to complement and extend ontologies, in the form of expressing constraints, reacting to events/changes, discovering new knowledge, transforming data, etc.

In order to provide individual adaptation in the tutoring systems it is necessary to store the information about the learners (goals, preferences, knowledge, etc.) to be used for adaptation purposes. This information constitutes the learner model [8]. To achieve the goal of personalized adaptive learning, prior knowledge helps to distinguish what learners already know and what they do not know [9]. The learner model represents the state of knowledge of the learner in the concerned subject and helps in deciding the correct teaching strategy to be used for the learner.

In the previous works realized an adaptive and intelligent web-based PRogramming TUtoring System – Protus that applies recommendation and adaptive hypermedia [10]. This system is realized as a general tutoring system for different programming languages, but it completely tested for an introductory Java programming course. The implemented system aims at automatically guiding the learner's activities and recommend relevant links and actions to him/her during the learning process.

The learner knowledge base in Protus is represented by an overlay model in which the current state of a learner's knowledge level is described as a subset of the overall architecture. The learner model includes learner's personal information, background, goals, and learning style as well as his/her competence levels for each concept node and each unit in the content tree, and an overall subject competence level.

The main objective of the paper is to propose a new features of Protus that will allow rule-based reasoning. The learner modeling 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, updates to learner model are generated and data about learner's navigation and progress through course is collected. This data can be further used to implement the concept of adapted content and adapted navigation.

In this work we will present the Protus learner model, its architecture, ontology and SWRL (Semantic Web Rule Language) [11] rules for updating the learner model.

The rest of the paper is organized as follows. In the second section appropriate related work is discussed. Section 3 describes the used technologies. Details about learner modeling in Protus are presented in Section 4. Section 5 presents implemented SWRL rules. Section 6 brings conclusions and indicates directions of further research.

2 Related Work

Recently, a lot of research efforts have been focused on applying Semantic Web technologies to different aspects of e-learning [4]. A wide range of educational software that implements ontology-based components has been developed, but the most of these systems use ontologies only for representation of concepts, knowledge or learners data

[5, 12] while details about rule-based reasoning are omitted. A model of a web-based personalized intelligent tutoring system with a learner model is presented in [13]. That model makes use of learners' knowledge levels, psychological characteristics and learning styles in order to construct and update learner details [14].

Main attention, in agent-based approach [15], is paid to the usage of ontologies for agent communication and formal description of learning components and processing strategies. Several aspects of usage of ontologies for selecting and utilization of instructional strategies can enhance power of thinking and problem solving.

Authors in [16] described an iterative methodology in order to develop and carry out the maintenance of web-based courses. They apply association rule mining technique (*if-then* recommendation rules) in order to discover interesting information through learners' activities.

The learner model of adaptive learning system based on Semantic Web has been presented in [17]. This model mainly considers three factors including learner study style, cognition level, interest and hobby and so on. The authors used Protégé to set up learner model ontology and used data mining technology to update learner model.

Other examples of an ontology-based learner models and description of the adaptation mechanisms are presented in [18-21]. These works show that the most relevant difficulty in the knowledge modeling in e-learning systems is related to the Semantic Web structures (such as ontologies). These ontologies can be exploited not only to organize learning objects and to state their inter-relationship but also to build personalized learning paths and to maintain up to date learner cognitive states.

The most of the systems concentrate more on advancing a learner's state of knowledge than on analyzing and improving the learner's cognitive state. Besides, that does not facilitate the definition and execution of rules that provide constant updating of learner model. Architecture for learner model ontology supported by several SWRL rules is described in this paper. Such approach brings acceptable solution for the personalization process. Further we present a way of linking semantics and content, and implementation of rule-based reasoning, within learner modeling in Protus. Appropriate SWRL rules are defined and executed in order to accomplish effective and scalable learner modeling in Protus.

3 Used Semantic Web Technology

With the rapid development of the semantic web the usage of ontologies is becoming common [22]. The Ontology Web Language (OWL) [23] standard is propelling this trend toward large scale application in different domains. However, the utility of the ontologies is limited by the processing mechanisms that are smoothly integrated with this form of representation. The SWRL is proposed as an important step in this direction, building on the experience of the previous work on Rule Markup Language - RuleML [11]. Eventually the availability of standardized rule language for the semantic web will make it possible to use both ontologies and rules as a basis for innovative applications that are connected to the semantic web.

SWRL is a language targeted to introduce inference rules in knowledge models represented in OWL [24]. SWRL [11] is probably the most popular formalism in Web community. The main advantage of SWRL is the simplicity, while extending the expressiveness of OWL. Its syntax and semantics is compatible with OWL, since they are both combined in the same logical language. Most of the existing rule-based applications for the Web have adopted SWRL approach in order to express rules [25].

Protégé is ontology editor and knowledge-base framework [26]. It provides graphical user interface for easy management of ontology. The Protégé OWL Plug-in provides a SWRL editor, which enables the formalization of SWRL rules in conjunction with OWL ontologies. Other ontology can be imported to achieve knowledge reuse.

Jess is a Java framework for editing and applying rules and it contains a scripting environment and a rule engine [27]. Recently, the evolution of rule technologies on the Web has led Jess to rebound its practical value in the community of Web developers (obtains integration with paradigms like Java servlets or applets).

4 Learner Modeling

Building of the learner model and tracking related cognitive processes are important aspects in providing personalization. The learner model is a representation of data about an individual learner that is essential for an adaptive system. The system uses that data from learner model in order to predict the learner's behavior, and thereby adapt to his/her individual needs. Learner model is a collection of static and dynamic data about the learner [28]. Static data include personal data, specific course objectives, etc. Dynamic data include scores, time spent on specific lesson, marks, etc. Also, learner model contains a data about this/her performance and learning history.

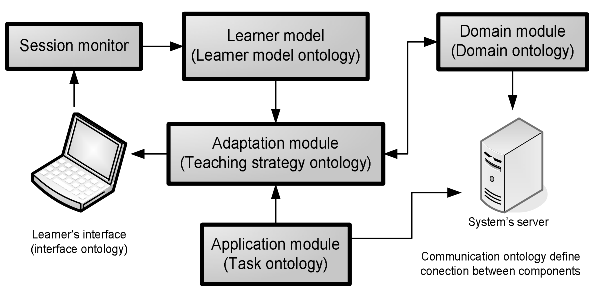

Fig. 1. Global architecture of Protus

Learner model in Protus presents crucial element of global Semantic Web infrastructure (figure 1). Educational ontologies for different purposes was included in the system, such as for presenting a domain (*domain ontology*), building learner model (*learner model ontology*), presenting of activities in the system (*task ontology*), specifying pedagogical actions and behaviors (*teaching strategy ontology*), defining the semantics of message content languages (*communication ontology*) and specifying behaviors and techniques at the learner interface level (*interface ontology*).

Data from learner model in Protus is classified along three layers [29]:

- *Objective information*: personal data, previous knowledge, preferences, etc. The learner edits this data during his/her registration on the system.
- *Learner's performance:* data about level of knowledge of the subject domain, his/her misconceptions, progress and the overall performance for particular learner.
- *Learning history*: information about lessons and tests learner has already studied, his/her interaction with system, the assessments he/she underwent, etc.

In order to accomplish successful categorization of learners we tracked characteristics of the learner and collected a variety of useful information:

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

The learner model is initialized by a simple but carefully designed questionnaire which is presented to the learner in first session [10]. The initial overall competence level is decided by checking the learner grades of prerequisite courses and previous experience data, if available. The learning styles are assessed by tracking learning behavior. The learners are allowed to set and modify their learning preferences and goals [29]. The competence level of each concept is dynamically updated at each interaction, which is then used to update the competence levels of the related leaf units in the content tree. The competence levels of non-leaf units are determined.

Protus system gradually re-builds the learner model during the session, in order to keep track of the learner's actions and his/her progress, to detect and correct his/ her errors and possibly to redirect the session accordingly. At the end of the session, all of learners' preferences are recorded. The learner model is then used along with other information and knowledge to initialize the next session for the same learner [29].

4.1 Learner Model Ontology

The learner model ontology was built using Protégé and presents a means for storing personal preferences and data about the learner's mastery of domain concepts [29]. The information is regularly updated according to the learner's interactions with the content and is used by the adaptation module of Protus to draw conclusions and decisions. Figure 2. illustrates this ontology. This ontology offers the opportunity to map all information about the learner, from confidential data (password), to a knowledge evolution history. The top class of learner model ontology is *User* class, which has metadata such as Identifier, Name, Last name, Gender, Address, etc. Subclasses of the *User* class are *Teacher* and *Learner* (represent details about teacher and learner). The class *Learner* consists of: *Performance*, *PersonalInfo*, and *LearningStyle* components. These three classes are related to association through *hasPerformance*, *hasInfo*, and *hasLearningStyle* properties. Class *LearningStyle* represents the preferred learning style for particular learner, according to Felder-Silverman Learning Style Model (sequential/global, active/reflective, visual/verbal and sensing/intuitive) [30].

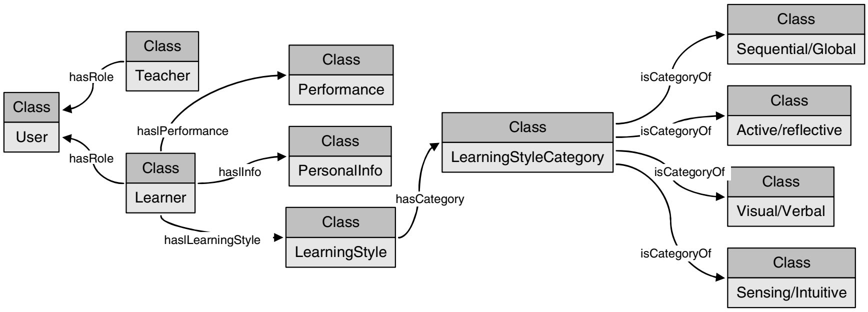

Fig. 2. Learner model ontology of Protus

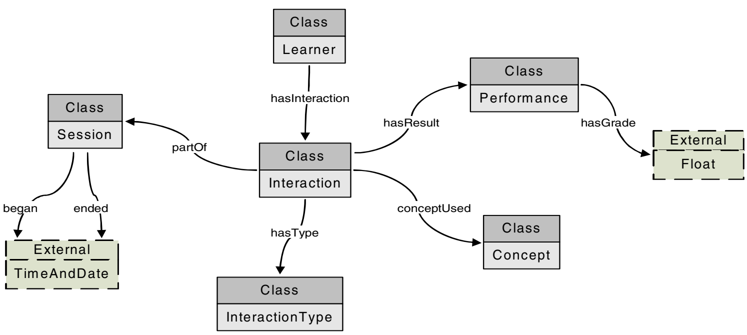

Fig. 3. Ontology for learner observation and modeling

At run time, learner interactions can be used to draw conclusions about his/her possible interests, goals, tasks and knowledge, in order to provide personalization. Ontology for learner observations should therefore provide a structure of information about possible learner interaction. Figure 3. depicts such ontology as a part of *Learner model ontology*. Learner performance is maintained according to a class *Interaction*. *Interaction* is based on actions taken by specific learner, during specific *Session*. *Interaction* implies a *Concept* learned from the experience, which is represented by *conceptUsed* property. Interaction has a certain value for *Performance*, which is in this context defined as a floating point number and restricted to the interval from 1 to 5. This ontology is responsible for updating the *Learner model ontology*.

5 Rule-Based Reasoning

Implemented ontologies have a set of basic implicit reasoning mechanisms derived from the description logic but they need rules to make further inferences and to create useful relations [31]. Thus, rule systems require taxonomies in order to have a shared definition of the concepts and relations, and taxonomies require a rule system to derive/use further information that cannot be captured by them [18].

The proposed rules consist of an antecedent (body) and a consequent (head), each of which consists of a set of atoms.

Learner Modeling Rules - This section describes some examples of inference rules. When a learner is logged in, a session is initiated based on learner specific learning style and sequence of lessons are recommended to him/her [10]. After selecting a lesson, from available the collection of Java tutorials, system chooses presentation method based on the preferred style. For the rest of the lesson, learners were free to switch between presentation methods by using the media experience bar [10]. When the learner completes the sequence of learning materials, the system evaluates the learner's knowledge degree for each lesson. Following rule updates learner model:

```
Learner(?x) ∧ Interaction(?y) ∧ hasInteraction(?x,?y) ∧
Concept(?c) ∧ conceptUsed(?y,?c) ∧ Performance(?p) ∧ ha-
sResult(?y,?p) ∧ asGrade(?p,?m) ∧ swrlb:greaterThan(?m,
1) → hasLearned(?x,?c) ∧ 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*. Concept presents a learning object which has been accessed by the learner in the current session. Meaning of the rule is: if in any time of the execution of Protus, 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. Previous rule is executed when learner earn positive grade. If learner shows insufficient knowledge, next rule is executed:

```
Learner(?x) ∧ Interaction(?y) ∧ hasInteraction(?x,?y) ∧
Concept(?c) ∧ conceptUsed(?y,?c) ∧ Performance(?p) ∧ ha-
sResult(?y,?p) ∧hasGrade(?p,?m) ∧ swrlb:equal(?m, 1) →
hasExecuted(?x,?c) ∧ hasPerformance(?x,?p)
```

Previous rule marks concept as executed but learned status is still left negative, meaning that new concept that supports same learning object will be used in next iteration.

If learner does not provide required level of performance results within session with presentation method used for certain learning style category, his/her initial learning style category will be modified with following rule:

```
Learner(?x) ∧ hasLearningStyle(?x,verbal) ∧ Interac-
tion(?i) ∧ hasInteraction(?x,?i) ∧ Concept(?c) ∧ concep-
tUsed(?i,?c) ∧ ConceptRole(?r) ∧ hasRole(?c,?r) ∧ sup-
ports(?r, verbal) ∧ Performance(?p) ∧ hasResult(?i,?p) ∧
hasGrade(?p, grade) ∧ swrlb:lessThan(grade, required) →
hasLearningStyle(?x,visual)
```

Variables *x, i, c, r* and *p* present *Learner, Interaction, Concept, Concept role* and *Performance,* respectively. Meaning of the rule is: if in any time of the execution of Protus, exists learner with *Verbal* learning style which interacts with system and during that interaction he/she had accessed appropriate concept but not earned sufficient grade (required grade level is kept in global value *required*), than, learning style of that learner should be changed. Protus provides different presentation methods for 18 Java programming lessons, depending of learning style of particular learner. If initial learning style for learner was visual, than next rule would be executed:

```
Learner(?x) ∧ hasLearningStyle(?x, visual) ∧ Interac-
tion(?i) ∧ hasInteraction(?x,?i) ∧ Concept(?c) ∧ concep-
tUsed(?i,?c) ∧ ConceptRole(?r) ∧ hasRole(?c,?r) ∧ sup-
ports(?r, visual) ∧ Performance(?p) ∧ hasResult(?i,?p) ∧
hasGrade(?p, grade) ∧ swrlb:lessThan(grade, required) →
hasLearningStyle(?x,verbal)
```

Similar rules will be executed for other categories of learning styles (intuitive/sensing, global/sequential and active/reflective). The above SWRL rules can be executed using the Jess rules engine. After firing the rule engines, the inferred knowledge can be written back to the OWL repository as used to update the knowledge base. Whereas ontologies were used to increase interoperability and reusability of domain information, rules were employed to represent the adaptation logic in a way that teachers can inspect, understand and modify the rationales behind adaptive functionalities.

In our opinion the main achievements of our work are threefold:

- explicit representation of the rules, encouraging their understandability, maintainability and reusability,
- component-based definition of adaptive tutoring system that uses first-order logic to perform personalization,
- appropriate implementation of the rules in an adaptive tutoring system.

These rules represent good bases for further extension and modification in detailed modeling process. Also, the rules can be modified for specific learner modeling requirements.

6 Conclusion

Although some systems take a learner's characteristics (e.g. knowledge levels, learning styles, etc.) and needs into account, choice of right learning material or presentation method to the specific learner is especially important in order to reach the desired teaching effects. Hence, various pedagogical tactics are introduced in tutoring systems to perform personalized teaching. Intelligent pedagogical agents produce personalization based on data from learner models, teaching material, teaching methods etc.

In this paper we proposed usage of ontologies and rule languages for building learner model in Java tutoring system. Proposed architecture for building learner model completely relies on Semantic Web standards. The form of several learner model ontologies and SWRL rules for inferring and updating learner model has been presented. This ontology-based approach allows implementing adaptation customized to different requirements. 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, updates, necessary for further adaptation to learner model had been performed.

Acknowledgements. This paper is part of the research project *Infrastructure for Technology Enhanced Learning in Serbia* supported by the Ministry of Education and Science of the Republic of Serbia [Project No. 47003].

References

1. Devedžić, V.: Semantic Web and Education. Springer Science, New York (2006)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 184(5), 34–43 (2001)
3. Mizoguchi, R., Bourdeau, J.: Using Ontological Engineering to Overcome AI-ED Problems. International Journal of Artificial Intelligence in Education 11(2), 107–121 (2001)
4. Jovanović, J., Rao, R., Gašević, D., Devedžić, V., Hatala, M.: Ontological Framework for Educational Feedback. In: SWEL Workshop of Ontologies and Semantic Web Services for IES, AIED, pp. 54–64 (2007)
5. Gascueña, J.M., Fernández-Caballero, A., González, P.: Domain Ontology for Personalized E-Learning in Educational Systems. In: The Sixth IEEE International Conference on Advanced Learning Technologies, pp. 456–458 (2006)
6. Carmagnola, F., Cena, F., Gena, C., Torre, I.: A semantic framework for adaptive web-based systems. In: Bouquet, P., Tummarello, G. (eds.) SWAP, vol. 166 of CEUR Workshop Proceedings. CEUR-WS.org (2005), `http://www.ceur-ws.org/Vol-166/17.pdf`
7. Henze, N., Dolog, P., Hejdl, W.: Reasoning and Ontologies for Personalized E-Learning in the Semantic Web. Educational Technology & Society 7(4), 82–97 (2004)
8. Kobsa, A., Koenemann, J., Pohl, W.: Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. The Knowledge Engineering Review 16(2), 111–155 (2001)
9. Wang, F., Chen, D.: A.: Knowledge Integration Framework for Adaptive Learning Systems Based on Semantic Web Languages. In: Eighth IEEE International Conference on Advanced Learning Technologies, pp. 64–68 (2008)
10. Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z.: E-Learning Personalization Based on Hybrid Recommendation Strategy and Learning Style identification. Computers & Education 56, 885–899 (2011)
11. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, `http://www.w3.org/Submission/2004/SUBM-SWRL-20040521`
12. Hee Lee, C., Hyun Seu, J., Evens, M.W.: Building an Ontology for CIRCSIM-Tutor. In: 13th Midwest AI and Cognitive Science Society Conference, Chicago, pp. 161–168 (2002)
13. Zhiping, L., Tianwei, X., Yu, S.: A Web-Based Personalized Intelligent Tutoring System. In: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, vol. 05, pp. 446–449 (2008)
14. Yu, S., Zhiping, L.: Intelligent Pedagogical Agents for Intelligent Tutoring Systems. Computer Science and Software Engineering (1), 516–519 (2008)

15. Jaesoo, K., Shinn, Y.H.: An Instructional Strategy Selection Model Based on Agent and Ontology for an Intelligent Tutoring System. In: IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, pp. 848–853 (2010)
16. Garcia, E., Romero, C., Ventura, S., de Castro, C.: An architecture for making recommendations to courseware authors through association rule mining and collaborative filtering. User Modeling and User-Adapted Interaction 19(1-2), 99–132 (2009)
17. Baishuang, Q., Wei, Z.: Student Model in Adaptive Learning System based on Semantic Web. In: First International Workshop on Education Technology and Computer Science, pp. 909–913 (2009)
18. Henze, N., Nejdl, W.: A Logical Characterization of Adaptive Educational Hypermedia. In: International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems, vol. 10(1), pp. 77–113 (2004)
19. Brut, M., Asandului, L., Grigoras, G.: A Rule-Based Approach for Developing a Competency-Oriented User Model for E-Learning Systems. Internet and Web Applications and Services, 555–560 (2009)
20. Liu, H., Tang, S., Ma, L.: A Rule-Based Approach for Student Modeling. In: Fifth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 04, pp. 526–530 (2008)
21. Zhiyong, L., Lei, L., Hui, K., Shaochun, Z., Bing, J.: An Ontology-Based Method of Adaptive Learning. In: Fifth International Joint Conference on INC, IMS and IDC, pp. 1586–1591 (2009)
22. Swartout, W., Tate, A.: Guest Editors' Introduction: Ontologies. IEEE Intelligent Systems 14(1), 18–19 (1999)
23. Tran, T., Cimiano, P., Ankolekar, A.: Rules for an Ontology-based Approach to Adaptation. In: First International Workshop on Semantic Media Adaptation and Personalization, pp. 49–54 (2006)
24. Sicilia, M.A., Lytras, M.D., Sances-Alonso, S., Garcia-Barriocanal, E., Zapata-Ros, M.: Modeling instructional-design theories with ontologies: Using methods to check, generate and search learning designs. Computers in Human Behavior 27(4), 1389–1398 (2011)
25. Papataxiarhis, V., Tsetsos, V., Karali, I., Stamatopoulos, P., Hadjiefthymiades, S.: Developing rule-based applications for the Web: Methodologies and Tools. Web Technologies: Concepts, Methodologies, Tools, and Applications 4, 456–477 (2010)
26. Protégé, `http://protege.stanford.edu/`
27. Jess (Java Expert System Shell), `http://jessrules.com/`
28. Ullrich, C.: Description of an Instructional Ontology and its Application in Web Services for education. In: Poster Proceedings of the 3rd International Semantic Web Conference, pp. 93–94 (2004)
29. Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z.: Integration of recommendations and adaptive hypermedia into Java tutoring system. Computer Science and Information Systems – ComSIS 8(1), 211–224 (2011)
30. Felder, R.M., Soloman, B.A.: Index of learning styles questionnaire (1996), `http://www.engr.ncsu.edu/learningstyles/ilsweb.html` (retrieved December 17, 2009)
31. Carmagnola, F., Cena, F., Gena, C., Torre, I.: MUSE: A Multidimensional Semantic Environment for Adaptive Hypermedia Systems. In: The Proceedings of ABIS 2005, pp. 14–19 (2005)