Timing the Adaptive Learning Process with Events Ontology

Martin Homola[0000−0001−6384−9771], Ján Kľuka[0000−0002−3406−3574], Zuzana Kubincová[0000−0003−1909−6738], Patrícia Marmanová, and Milan Cifra

Comenius University in Bratislava,
Faculty of Mathematics, Physics, and Informatics
Mlynská dolina, 842 48 Bratislava, Slovakia
firstname.surname@uniba.sk

Abstract. A number of studies in personalized adaptive learning have focused on generating suitable learning paths based on user's model, considering the current level of knowledge of the user, preferred learning styles and a model of the subject domain. These factors are sufficient in many e-learning applications, where users consume the learning content at their own pace. In other applications, such as within organized curricula there are other factors to be considered too. At the university, we deliver courses featuring project work and examination which the students have to deliver based on a schedule of deadlines. This time axis, therefore, presents a significant factor in recommending the most suitable learning objects at the given time of the term. To tackle this issue we have designed a courseware platform where time is one of the key factors determining the learner's context. In this paper, we focus especially on modelling the time access using an ontology and we show some preliminary results that are implied by this approach.

Keywords: Courseware · Personalization · Leaner's context · Time · Ontology.

1 Introduction

The aim of personalized adaptive learning is to provide a tailored learning experience for each user according to their own needs. For example, recommender systems may be employed to help users to choose the most suitable and relevant learning materials that meet their individual learning needs [26, 20]. Many of the existing e-learning recommender systems rely on techniques such as collaborative filtering, content-based, demographic-based, utility-based, and knowledge-based techniques which basically develop a rating system expressing the recommendation relation between users and the learning content [26].

However, a more current approach in personalized adaptive learning notes the importance to consider the learner's specific demands and requirements more explicitly [17]. Different learners have different background knowledge, learning history, competence level, preferred learning styles, etc. [7] which are not taken into consideration in conventional recommender systems like collaborative filtering and content-based recommenders [26]. This drawback may be overcome by integrating knowledge structures representing complex models of all aspects that are relevant to the learning process.

In recent years, ontologies gained importance for representation of such models in e-learning systems.

Ontologies serve as formal definitions of concepts and their relations in some domain of interest of human expertise [11, 24, 25]. They are crafted with the purpose of making the conceptualization unambiguous and explicit. This goal is best achieved by writing ontologies in a formal and unambiguous language that supports reasoning. Currently, the most popular languages include RDFS [5] and OWL [10] (cf. [2, 1]). Ontologies can then be used within information systems to provide schemas for data, to assure data interoperability, enable data reuse, etc. [25].

In personalized adaptive learning, ontologies are employed especially to describe (a) the abstract domain of learning objects, their different types and properties, (b) particular domains of topics of interests which are the subject of learning, (c) pedagogical properties of learning processes, (d) user models, including user's existing knowledge, skills, and preferred learning styles.

User's experience may then be tailored based on the available data from these domains, especially by recommending the most suitable learning objects or even generating personalized learning paths, consisting of sequences of learning objects.

While such knowledge structures may be sufficient in many application scenarios of e-learning, especially in self-directed learning, there are other factors relevant in learning processes. *Context*, i.e., "the circumstances in which the learning process occurs" becomes relevant as well [3]. While many earlier works stemming from adaptive hypermedia mostly explore context in the form of platform and device adaptation issues, a broader and more general sense of context is becoming increasingly relevant in this area [6].

In this work, we focus on the user's context within an organized education setting. There are possibly different contextual issues to be considered. We narrow our aim towards the temporal context imposed by the courses being scheduled in form of a series of events such as lectures and lab sessions occurring during a fixed time period framed by school years or terms. While lectures and lab sessions may often be scheduled equally for all course attendants and hence may not require personalization, other events such as assignments and examination sessions may be, e.g., subscription-based and hence different students may have different schedules. In addition, even if some events, such as lectures, are fixed, it is useful to include and consider them in the overall personalization framework uniformly together with the other events and other personalization aspects.

We start by a brief review of related works in Section 2. Then in Section 3 we focus on the organized education setting and discuss how learning content adaptation in such a setting must consider also the past or upcoming events, e.g., by considering learning materials associated with current lectures or labs, or recommended w.r.t. upcoming coursework deadlines and examination dates. In Section 4, we present our proposed LMS platform that is based on multiple ontologies, including a time-frame ontology, which we focus on in detail in Section 5. In Section 6 we discuss how users of our platform would benefit from this approach: as our very first result, we show how tracking the course of events during a course run using the time-frame ontology enables us to present the learning content in concise timeline automatically generated from the time-related metadata associated with the remaining content. We then conclude and discuss future possibilities in Section 7.

2 Related Works

While most of the studies addressing the recommendation of learning resources and personalization in e-learning environments using ontologies make only use of domain ontology [21, 18] there were also ontology-based recommenders developed using more domain ontologies or other types of ontology [8, 4, 27].

Cobos et al. [8] developed an ontology-based hybrid recommender system (RSPP) which allows lecturers to define their best teaching strategies for use in the context of a specific class. To construct the RSPP, a reference ontology representing the pedagogical patterns and their interaction with the fundamentals of the educational process was defined.

Bahmani [4] presents a recommendation algorithm for personalization of course and curriculum content for individual students considering various kinds of context such as the academic background, interests, computing environment of the student, and also past recommendations made to students with similar profiles. Context modelling is based on a combination of a generic and a domain ontology.

The Protus tutoring system for Java programming [27] automatically guides the learner's activities and recommends relevant links and actions. It provides (a) content adaptation – recommending optimal resources and pathways based on the domain model and information about learning styles of the current learner; and (b) learner interface adaptation – adjusting the appearance and/or availability of learning resources on a course web page based on recommendations respective to different learners. Each system component is represented by its own ontology, thus domain ontology, task ontology, learner model ontology, teaching strategies ontology and interface ontology are incorporated.

Saleena and Srivatsa [22] propose an adaptive e-Learning system, which generates user-specific e-Learning content by comparing the concepts using similarity measures. A cross ontology measure is defined over a fuzzy domain ontology as the primary ontology and a domain ontology as the secondary ontology for the comparison process.

Yu et al. [28] make recommendations by exploiting the user context of a learner, knowledge about learning content, and knowledge about the learning domain. They consider two kinds of contexts: the learner's prior knowledge and her learning goal. They rely on three ontologies: learner ontology, learning content ontology, and domain ontology.

Schmidt and Winterhalter [23] present an integrative, ontology-based approach in which the ontology is divided into several sub-ontologies, such as: organizational ontology (roles, departments); process ontology (workflow representations); task ontology; knowledge area ontology. Each is organized in layers so that the upper layers can be shared with other entities and the lower layers can still be extended in a domain-specific way. This enables them to propose recommendations w.r.t. learner's context which is described (at a general level) in terms of organizational structure, the current task at hand and given workflow that is being followed, etc.

Jovanović et al. [16] introduced Learning Object Context – a collection of LO metadata capturing all the information that characterizes the specific situations (contexts) in which certain LO has been used. They also developed a corresponding ontology framework (LOCO) consisting of (a) learning object content structure ontology, (b) learning design ontology, and (c) learning object context ontology. The proposed framework was implemented in TANGRAM, a Web-based application for personalized learning in the area of Intelligent information systems.

The granularity of recommended learning objects varies, including courses and curricula [4], learning patterns [8], relevant learning links and actions [27], learning paths and content [15], various types of learning objects, etc.

3 The Organized Education Setting

Unlike many existing works reviewed above, we aim to apply adaptive learning in organized university education where other factors besides for those typically studied determine the learner's context.

The content of a typical university course is delivered in the form of learning sessions such as lectures or practicals that happen in precise points of time, typically within a regular sequence. However, the sessions may possibly be grouped either based on time intervals, such as weeks, or either based on related topics.

Besides learning sessions, there are other relevant events that the learners need to track and take part in, especially related to coursework and examination.

Modern university education often favours the *learning by doing strategy*. Thus learners are required to submit coursework and possibly even take part in formative peer review processes which generate a number of deadlines that they have to watch and prepare for.

Thus the learner's context is not determined only by their current level of knowledge and preferred learning styles, but also by current time instance in the course of events generated by the current development of the course.

As an example let us consider a Master's level web design course delivered at our university. In this course, the students work in teams on a term-long project assignment. The assignment is delivered in four consecutive rounds (specification, prototype, application, content) which are preceded by team formation phase. In each round, the learners first deliver an *initial submission*. Consecutively they *peer-review* these submissions, and based on the peer-feedback they deliver an *improved submission*, which is then graded by course instructors. Finally, learners rate the teamwork within their teams during the *team review* period. Together with midterm and final test, and the oral exam, learners have to track and deliver results respective to 20 consecutive deadlines. For more details on these assignment workflows please refer to our previous works [13, 14].

Our past experience shows that tracking all the deadlines is challenging for the students. Some of them deliver poor results simply because they missed some deadlines or did not prepare properly ahead of the deadlines. Therefore we are determined to explore the options not only to be able to notify the learner before the approaching deadline but also to be able to recommend the most suitable learning materials to prepare for the coming deadlines.

4 Proposed LMS Design Overview

We are currently developing a novel LMS system that will pay increased attention to learner's context. It consists of several modules centred around the system's core. The overall design is depicted in Fig. 1. The *core* manages common data and provides common services and user interface (UI) components. These include components presenting the users with a unified view of some of the data managed by other modules. A module typically manages a specific kind of learning objects or tasks and provides UI components to display the data and control the management. Data of one module can be interlinked with data of other modules and a module can embed another one's UI components into its own to manage the linked data.

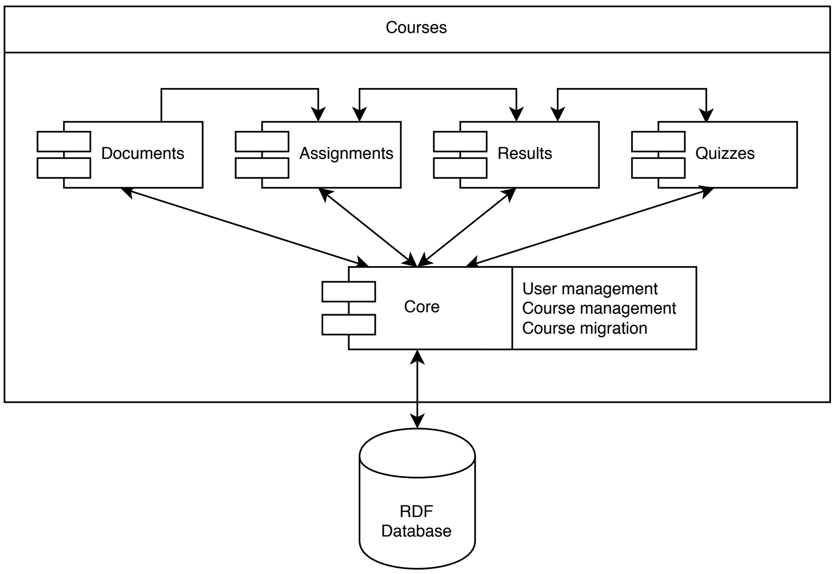

Fig. 1. System design overview

In particular, the *Assignments* module manages the kind of peer-reviewed assignments described above in Sect. 3. Instructors can specify an assignment's task and which deliverables of various kinds (text, programming code, media, links) the learners are expected to submit. They also choose which follow-up tasks (peer review, improved submission, team review) the learners will perform after their initial submission and set their parameters (e.g., time periods, review rubrics). The module then enables the learners to submit the deliverables and fill in the assigned reviews. The instructors can see all the artefacts thus produced, provide feedback, and evaluate.

The *Quizzes* module supports learning by formulating questions and administration of quizzes and tests. The instructors can task the students with formulating questions of various kinds (open, simple answer, single choice, etc.) covering given topics. Instructors can also create questions, and task the learners with taking a quiz, manually or automatically created from already existing questions.

The *Documents* module manages internal and external documents and multimedia, used chiefly as learning materials, but also as assignment specifications and deliverables submitted by learners. The *Results* module manages evaluation results, awarded to learners by the instructors for learning tasks carried out both within the LMS or externally. In the former case, the result is linked with the respective task.

5 The Time-Frame Ontology

The LMS aims at contextual support of the learners' work during learning activities and their access to relevant learning objects. There are also secondary goals such as easy reuse of data among the modules and the core. Storing the data in an RDF database, with a system of interrelated OWL ontologies as its schema, is especially useful to achieve these goals, which is also apparent from the related work. In addition to usual ontologies that model learning objects and processes, and domain ontologies that model the topics which represent the subjects of learner's studies, we also integrate an ontology that covers the *time frame* of the learning activities. As noted in Sect. 3, time is an important component of learner's context in organized university education.

The time-frame ontology is depicted in Fig. 2 in a modified VOWL notation [19]. The classes in this ontology (solid-line ovals) are subclasses of the *Event* class and they represent the time frame of various learning activities (represented by classes depicted by dotted-line ovals, as they are external to the time-frame ontology).

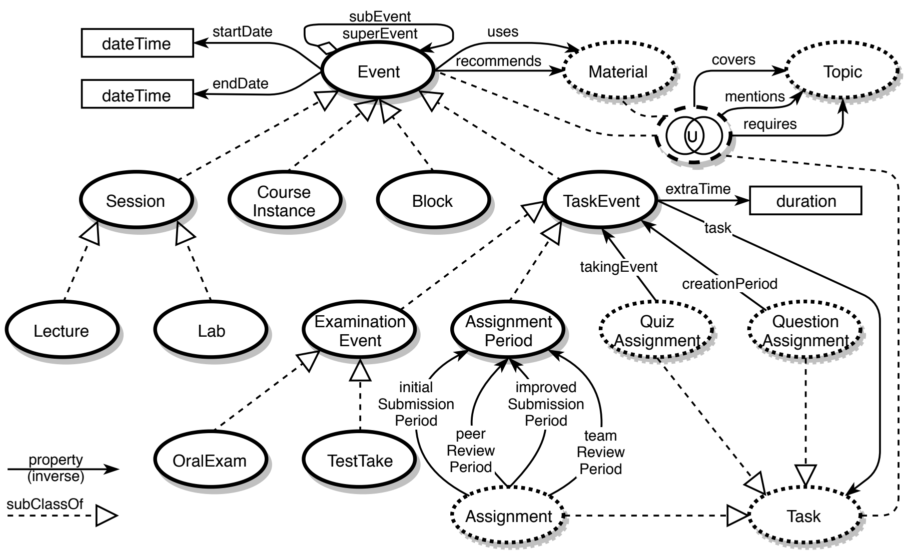

Fig. 2. Time-frame ontology

A *CourseInstance*, a particular run of the course in an academic year or semester, is itself an Event. The time frames of all activities within this instance are its *subEvents*.[1] Time-based groups of activities are represented as *Block*s. In the academic setting, one week is usually considered to be one block, but it is up to instructors to set up these blocks. A block may, e.g., span several consecutive weeks devoted to the study of one topic.

Blocks typically contain *Session*s – classroom activities involving face-to-face interaction of instructors and students. Sessions are usually regularly scheduled and specialized to lectures and labs, which are represented by the respective subclasses of Session.

Activities which require learners to complete some task either within a longer time frame or only once or a few times during a course instance are modelled as *TaskEvent*s. Examples of the latter kind are *ExaminationEvent*s – midterm and final tests or oral examinations. The former kind has three special cases related via the property *task* to a learning activity (*Task*) administered by the Assignments and Quizzes modules. We discuss these three kinds of events in the next two paragraphs.

The learners' work on an *Assignment* managed by the Assignments module is coordinated as a sequence of time periods during which initial submission, peer review, improved submission, and team review have to occur. These time periods are represented as instances of the *AssignmentPeriod* class, a subclass of TaskEvent, related to their Assignment by the respective properties. The *endDate*s of these periods represent the deadlines of the respective sub-activities.

The question formulation activity managed by the Quizzes module task is represented as a *QuizQuestionAssignment*. The value of its property *creationPeriod*, a TaskEvent instance, represents the time period within which it should be carried out. When instructors ask learners to take a quiz, a *QuizAssignment* is created, which has a *takingEvent*, another instance of TaskEvent, representing the time span within which the learners should take the quiz.

Tasks often have nominal deadlines which are advertised, but they are not strict because their strict enforcement by the LMS could adversely impact learners having minor technical difficulties. The *extraTime* property of TaskEvents thus allows the instructors to specify for how long the LMS should allow the learners to complete the task past its nominal endDate. We typically set extraTime to 45 minutes.

An Event, especially a Session, often *uses* some learning materials. A *Material* (also external to the time-frame ontology, thus depicted by a dotted-line oval) represents a hypertext document (e.g., lecture slides, labs exercises) or a multimedia file stored within the LMS or linked from external sources. For some Events, instructors can also *recommend* studying some materials beforehand (reading before a lecture, exercises before a test) or afterwards (reading with more details on a lecture's topic). Moreover, Events, Materials, and Tasks are related to *Topic*s from the topics-of-interest (i.e., domain) ontology in several ways. They can *cover* or just *mention* a topic, or they may *require* the

[1] Note the difference between subevent and subclass. The latter stands for a logical relation between two classes, one being more specific than the other (expressed by OWL subClassOf axiom). The former is an aggregative relation between two instances of events of different granularity a lecture happening during a course run (expressed by OWL object properties subEvent and superEvent).

knowledge of a topic in order to complete the task or to understand the content discussed during the event or within the material.

Although learning activities are managed by different modules, their time frames are all derived from the Event class. This allows the LMS's core to integrate the data and to produce a unified presentation of the events, their related learning activities and objects, as discussed in Sect. 6. The core can also notify the learners on these events, suggest reviewing learning objects directly related to the event by the uses and recommends properties, or even derive related learning objects and previous or upcoming activities based on Topics related to the event or its activity.

The Event and CourseInstance classes, as well as the startDate, endDate, subEvent, and superEvent properties can be directly mapped to the respective types and properties in the Schema.org vocabulary [12]. Schema.org also specifies other properties (e.g., the *location* of an Event, the *instructor* of a CourseInstance), which may prove useful in the future development of the LMS. The start- and endDates can also be mapped to *Instant*s and Events can be associated with *Interval*s of the Time Ontology [9]. However, we do not plan to do so in the time-frame ontology. Events and time intervals are different entities, as time spans of multiple events can be equal to a one time interval. Moreover, the level of detail of modelling temporal entities provided by the Time Ontology is not required, simple data values suffice.

6 Timeline Interface Generated from the Ontology

Tight integration of the time ontology with the remaining data enables us to visualize the course timeline in an automatically generated timeline interface. A prototype of this interface is depicted in Fig. 3.

The timeline puts all consecutive events in order and visualizes events of different specific types in a specific manner. It is generated from events as follows: All Blocks from the given course are taken and sorted by their startDate. They are displayed as larger boxes wrapping up other events occurring during their timespan. Subsequently, all events of type Session within each block are selected, sorted by their startDate and displayed to the learner as a schedule for the block. Similarly, all TaskEvents that will be due in the given block based on their *deadline* (mostly the endDate, but startDate in the case of ExaminationEvents) are selected, sorted by the deadline, and displayed as a kind of to-do list for the given block. Different types of sessions and tasks are distinguished by different icons. Notice, e.g., the *Development process* block in Fig. 3.

All materials linked to every event falling within the block are displayed in the lower part of the block. It works the same way with every event. When the event is opened, the linked materials and its sub-events are displayed.

Vertical navigation features a concise list of all Blocks and a section with upcoming deadlines of TaskEvents. It enables the learner to keep track of the assignments that are due or an upcoming examination which might require some preparation.

Clicking on an event zooms in and the event is visualized, as shown in Fig. 4, showing the details and events that occur within (if applicable).

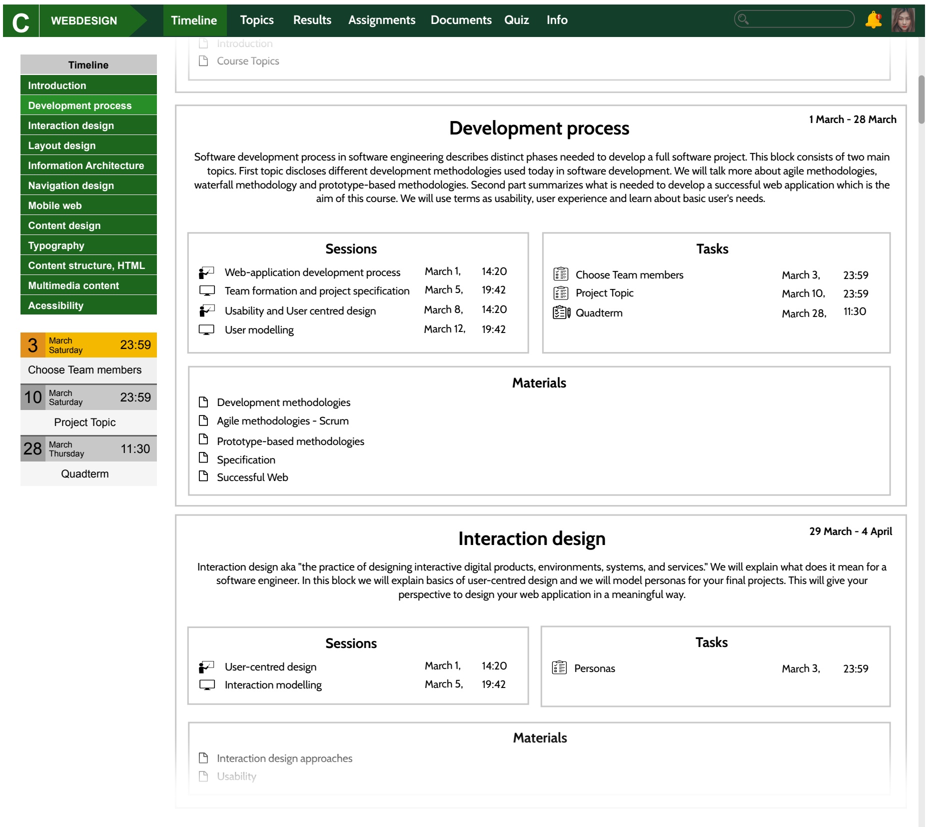

Fig. 3. Prototype of the timeline interface

Similar timelines can be created in, e.g., Moodle, and other courseware systems, however, the advantage of our approach is that the timeline is automatically generated from the available meta-information concerning the time context.

7 Conclusions and Future Work

We have argued that when applied in organized education, adaptive learning systems must also consider the context of the organized learning activity which is above all the time context given by a series of events constituting a typical course run. To that end, we have designed a time-frame ontology and described its role within a proposed LMS platform that we are currently developing. We have shown how interlinking the learning content with time metadata immediately enables to produce an automatically generated timeline, that helps users to orientate in the large pile of learning materials and deadlines associated with the course run *at any given time*.

Of course, using the ontology to keep track of all (even fine-grained) events associated with a course run in a uniformed way would bring many other benefits to our sys-

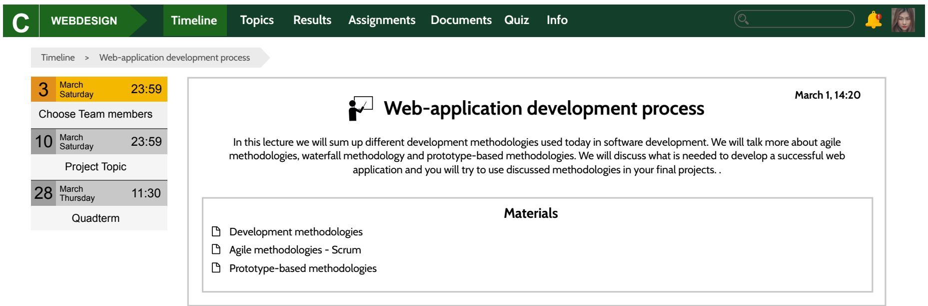

Fig. 4. Prototype of the zoomed-in lecture event interface

tem and to its users. For example, as we explained above applying collaborative learning strategies including team assignments and peer-review of submitted coursework generates a number of deadlines. The previous version of our system used hardcoded notifications to call the students' attention to these deadlines. Modelling the assignments (together with all other events) in the ontology using task-events and their sub-events (such as e.g. initial submission, peer-review period, final submission, team-review period) and even dependencies such as peer-review period being a prerequisite of final submission enables us e.g. to base the notifications on the type of event (a lecture or lab session may be notified one day ahead, the midterm test may be notified three days ahead, and some other events may not be notified at all).

In addition, combining the time-related data based on the time-frame ontology with the other semantic information may be used to provide even more fine-grained and more narrowly directed suggestions regarding the recommended next learning actions. For example, the examination events will also be linked with concepts from domain ontologies representing the topics that the examination will cover. Together with topics linked to materials and other learning objects and together with data about the learners current level of knowledge and preferred learning styles this may be used to produce the recommended learning actions to take before the examination takes place. Exploring this direction is part of our ongoing work.

Acknowledgments. This work was supported from the Slovak national VEGA project no. 1/0797/18 and from APVV grant no. SK-TW-2017-0006.

References

1. Allemang, D., Hendler, J.: Semantic web for the working ontologist: effective modeling in RDFS and OWL. Elsevier (2011)
2. Antoniou, G., Van Harmelen, F.: A Semantic Web Primer. MIT press (2004)
3. Aroyo, L., Dolog, P., Houben, G.J., Kravcik, M., Naeve, A., Nilsson, M., Wild, F.: Interoperability in personalized adaptive learning. Journal of Educational Technology & Society **9**(2), 4–18 (2006)

4. Bahmani, A., Sedigh, S., Hurson, A.: Ontology-based recommendation algorithms for personalized education. In: International Conference on Database and Expert Systems Applications. pp. 111–120. Springer (2012)
5. Brickley, D., Guha, R.V. (eds.): RDF Schema 1.1. Recommendation, W3C (25 February 2014)
6. Brusilovsky, P.: From adaptive hypermedia to the adaptive web. In: Mensch & Computer 2003, pp. 21–24. Springer (2003)
7. Buder, J., Schwind, C.: Learning with personalized recommender systems: A psychological view. Computers in Human Behavior **28**(1), 207–216 (2012)
8. Cobos, C., Rodriguez, O., Rivera, J., Betancourt, J., Mendoza, M., León, E., Herrera-Viedma, E.: A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes. Information Processing & Management **49**(3), 607–625 (2013)
9. Cox, S., Little, C. (eds.): Time Ontology in OWL. Recommendation, W3C (19 October 2017)
10. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: Owl 2: The next step for owl. Web Semantics: Science, Services and Agents on the World Wide Web **6**(4), 309–322 (2008)
11. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2), 199–220 (1993)
12. Guha, R.V., Brickley, D., Macbeth, S.: Schema.org: Evolution of structured data on the web. Commun. ACM **59**(2), 44–51 (Jan 2016)
13. Homola, M., Kubincová, Z., Čulík, J., Trungel, T.: Peer review support in a virtual learning environment. In: State-of-the-Art and Future Directions of Smart Learning, pp. 351–355. Springer (2016)
14. Homola, M., Kubincová, Z., Kľuka, J., Švolík, M., Darjanin, M., Zapalač, P.: Team workflow and peer review in a virtual learning environment. In: Information Technology Based Higher Education and Training, 2017 16th International Conference on. pp. 1–6. IEEE (2017)
15. Huang, C., Liu, L., Tang, Y., Lu, L.: Semantic web enabled personalized recommendation for learning paths and experiences. In: International Conference on Information and Management Engineering. pp. 258–267. Springer (2011)
16. Jovanović, J., Gašević, D., Knight, C., Richards, G.: Ontologies for effective use of context in e-learning settings. Journal of Educational Technology & Society **10**(3), 47–59 (2007)
17. Klašnja-Milićević, A., Vesin, B., Ivanović, M., Budimac, Z.: E-learning personalization based on hybrid recommendation strategy and learning style identification. Computers & Education **56**(3), 885–899 (2011)
18. Lau, R.Y., Song, D., Li, Y., Cheung, T.C., Hao, J.X.: Toward a fuzzy domain ontology extraction method for adaptive e-learning. IEEE transactions on knowledge and data engineering **21**(6), 800–813 (2008)
19. Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing ontologies with VOWL. Semantic Web **7**(4), 399–419 (2016)
20. Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., Koper, R.: Recommender systems in technology enhanced learning. In: Recommender systems handbook, pp. 387–415. Springer (2011)
21. Robin, C., Uma, G.: An ontology driven e-learning agent for software risk management. International Journal of Academic Research **3**(2) (2011)
22. Saleena, B., Srivatsa, S.K.: Using concept similarity in cross ontology for adaptive e-learning systems. Journal of King Saud University-Computer and Information Sciences **27**(1), 1–12 (2015)
23. Schmidt, A., Winterhalter, C.: User context aware delivery of e-learning material: Approach and architecture. Journal of Universal Computer Science **10**(1), 28–36 (2004)

24. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing (2000)
25. Staab, S., Studer, R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems, Springer (2009)
26. Tarus, J.K., Niu, Z., Mustafa, G.: Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. Artificial intelligence review **50**(1), 21–48 (2018)
27. Vesin, B., Ivanović, M., Klašnja-Milićević, A., Budimac, Z.: Protus 2.0: Ontology-based semantic recommendation in programming tutoring system. Expert Systems with Applications **39**(15), 12229–12246 (2012)
28. Yu, Z., Nakamura, Y., Jang, S., Kajita, S., Mase, K.: Ontology-based semantic recommendation for context-aware e-learning. In: International Conference on Ubiquitous Intelligence and Computing. pp. 898–907. Springer (2007)