Temporary Belief Sets Management in Adaptive Training Systems

Robert Andrei Buchmann, Anamaria Szekely, and Delia Pulcher

Babes Bolyai University,
Faculty of Economic Sciences and Business Administration,
Business Information Systems Dpt., str. Teodor Mihali 58-60,
400591 Cluj Napoca, Romania
robert.buchmann@econ.ubbcluj.ro,
{b_ana_sz,delialoman}@yahoo.com

Abstract. The paper proposes a semantic view on the notion of „learning object" and an application model based on RDF-based learning objects and learning processes. Direct feedback is individualized for test subjects and learning tasks, according to requirements defined for corporate training. The knowledge model allows contextualization and subjectivity, which, in turn, are used to dynamically generate temporary belief sets, compare them to the (theoretically) objective belief set underlying the learning content and adapt learning recommendations to each particular user. The semantic models also determine learning prerequisites and the screen flow adapted to each individual learner, thus influencing usability.

Keywords: Learning object, qualified knowledge, context, RDF, SPARQL.

1 Introduction

The goal of the paper is to propose a knowledge subjectivity management model for adaptive computer-based training systems, based on a notion of "temporary belief sets", which designate the intermediate states of a trainee's knowledge, mapped on learning content modeled with the Resource Description Framework [1]. The study is guided by individualization requirements for corporate learning adaptivity in business settings, as defined by [2].

Learning objects are abstract notions with imprecise definitions, boundaries and granularity. In the context of our approach, we consider that one of their most important features is reusability. Some authors also point to self-containment [3], but this attribute is slightly forced, as the knowledge expressed through a learning object collection, although it might be decomposable and modular, must reflect some dependencies with other concepts and learning objects. Furthermore, in natural language most term definitions are based on other terms, which in turn are based on other terms and so on, until one can reach circular references, or primitive concepts that are defined through themselves.

RDF is a knowledge representation framework underlying the paradigm of Semantic Web [4]. In RDF, concepts are built in the above mentioned manner: the

W. Abramowicz, L. Maciaszek, K. Węcel (Eds.): BIS 2011 Workshops, LNBIP 97, pp. 99–110, 2011.

notions of **class** or **property** are defined primitively and axiomatic: a class is a member of the class of all classes (*rdfs:Class rdf:type rdfs:Class*), a type is a type of property (*rdf:type rdf:type rdf:Property*). Most belief systems are built on such axiomatic, self-referential conventions, more or less formally and rigorously developed over the fundamental set of axioms. Learning content is built on the same principles, even in what are considered to be „exact" disciplines such as mathematics (which is itself based on relatively primitive notions such as the object, the relation, the number, the set etc.).

These are some reasons why, throughout our studies, we prefer to ignore the attribute of self-containment and insist on the attribute of reusability for learning objects. Also, employing RDF in knowledge representation forces us to emphasize connectivity in the detriment of self-containment.

The proposed model also has some subtle implications on the usability of computer-based training, due to its adaptivity and intelligent response to learning needs, based on the user input while he's taking evaluation tests. This impacts the learning efficiency and user's feeling that the system is "supportive" to his/her needs.

The next section states the problem and some of its background (including related works). Section 3 lists the low cost instruments needed for implementation. Section 4 describes implementation dilemmas and solutions regarding the subjective knowledge, ending with a general overview of the application in whose context this issue was raised, followed by a SWOT evaluation and final conclusions.

2 Problem Statement and Background

Most implementations of training systems and platforms (Moodle [5], Blackboard [6] etc.) are basically formatted computer-mediated communication systems aiming at eliminating costs and inconveniences regarding the management of a learning environment, usually aspects regarding time and space. They are less involved in automating the training processes themselves, as the intervention of the human tutor is still considered essential.

Although our proposed model is still based on human intervention, it provides a superior level of semantic automation by storing a formal representation of the knowledge states through which the trainee goes during his training sessions. This implies that the main issue to be modeled here is knowledge subjectivity and temporality which are the main attributes of a "temporary belief set".

The paper discusses several alternatives in expressing subjective knowledge, falseness and evaluated knowledge in a computing environment based on the Resource Description Framework and its inference vocabularies (OWL [7], custom production rules [8]). The instrumentation is provided by the coupled knowledge management system provided by OpenRDF Sesame and OWLIM, managed through a REST interface and extending a Python web application acting as a traditional e-learning application. The proposed model requires a knowledge engineer to define the mapping between the learning content and the learning objects, which are defined and delimited as RDF resources.

As for related works, first of all we mention a paper discussing the multiple interpretations assigned to the notion of learning object [9], an ambiguity that allows us to take the position mentioned above. Also, significant efforts have been made for the modeling of learning objects: The Learning Technology Standards Committee defined the Learning Object Metadata [10], while more ontology-oriented approaches can be found in [11]. A solution instrumentally closer to our approach (in the sense that uses Sesame as a storage platform) is [12] – it works with SeRQL as a query language and does not employ OWLIM for inferences. Learning object formalizations using OWL and semantic technologies have been also proposed by [13] (for the domain knowledge of first-aid), which in turn is inspired by the recomposition methodology proposed by [14].

Our work emphasizes subjectivity of knowledge rather than learning object structures and taxonomies, insisting on the idea that semantic repositories backing up adaptive learning systems must be flexible enough to support unstable knowledge.

3 Instrumentation

The tools required for implementation can be obtained freely:

- OpenRDF Sesame –RDF management system developed by Aduna Software, with weak inferential capabilities (RDF Schema and the Direct Type vocabulary) [15];
- OWLIM – rule-management system developed by Ontotext AD [16] as a an extension to Sesame, thus bringing inference capabilities and production rules to Sesame;
- Python is the language of choice for prototyping, which connects to the Sesame+OWLIM platform through its REST interface and Python's HTTP capabilities provided by the urllib package. SPARQL [17] queries can be run over the REST interface, with results returned as a JSON mapping of the the standard SPARQL Result Format [18].

4 The Proposed Model

An essential requirement for learning processes is the subjectivity of expressed knowledge and viewpoints – we have to express at least two sides – the trainer's and the trainee's, and on each of these sides there might be a large diversity of viewpoints. By stating this, we don't promote the philosophical debate that knowledge is volatile and non-absolute, but rather the fact that the trainee's knowledge set shifts through **multiple subjective states**, from the initial, realistic and „unlearnt" state, to a final, ideal and theoretical state, after the trainee had acquired all the knowledge provided by the training programme. Shifts between these states can be triggered during a test-driven learning recommendation system:

An evaluation of the initial state might reveal that the „unlearnt" responder chooses answers mostly randomly. This random set represents the first temporary belief set. When confronted with the right and wrong answers, some of the trainee's knowledge shifts towards the desired one, creating another temporary belief set. This confrontation shouldn't take the form of simply displaying the right answers, but rather the one of pushing forward (through reading recommendations) the learning objects containing the right answers. A new evaluation will reveal new problems, with new requirements for adjusting the belief set, and so on.

An essential problem with this is the representation of subjective knowledge – this includes all the temporary belief sets of all trainees, and a reference set that must be acquired through learning. This reference may also vary based on the tutor's interpretation, authorities, sources, without being necessarily wrong, but rather perfectible, unstable and temporary. Thus, the representation of training content must be aligned to the principle of AAA (anyone could say anything about anything), which is well served by the Semantic Web paradigm [19].

We have to emphasize that under the umbrella term of **subjectivity**, we fit both subjective viewpoints and a lack of viewpoints (for a beginner trainee). That's why the initial state of the trainee's belief set, although generated by a general lack of knowledge, is still represented as a subjective view rather than a lack of view.

Representing knowledge subjectivity opens capabilities for multiple analysis scenarios. The most important ones for our studies are filtering of right and wrong knowledge at any given time (according to a trainer's evaluation), delta analysis (differences between the temporary knowledge sets) and a general training strategy driven by evaluation (and concept dependencies) rather than content.

In the next section we discuss several alternatives in representing subjectivity:

4.1 Representation of Contextual Knowledge

It is well known that in RDF the basic knowledge unit is the triple, which establishes a binary, directional relationship between a subject and an object. The subject-predicate-object structure, defined by the RDF standard, is fit to express most reality descriptions but it is based on the assumption of objective reality – every triple occuring in knowledge base is considered true and valid (until inconsistency detection).

Reification is a knowledge representation pattern that can be defined as "qualified knowledge" or, in other words, "contextual knowledge". This means that reification allows for the representation of subjective knowledge, by assigning a context to a knowledge unit.

Although weakly supported by RDF frameworks, there is a subset of the N3 syntax that allows for expressing reification (which should translate in a 4-arity relationship, also defined by the standard RDF vocabulary but avoided by many implementations) [20].

Several scenarios where contextual assertions would prove useful are:

- when the context is spatio-temporal and limits the validity of the statement:

```
{:John :hasAgeOf 20} :inYear 2000 .
{:Humans :SkinColor :Yellow} :InRegion :China.
```

- when the context is the subjective source of the statement:

```
:Mary :ThinksThat  {:John :hasAgeOf 20}.
:John :SaidThat
               {:Mary :ThinksThat
                       {:John :hasAgeOf 20}}.
```

- when the context is an evaluation (adverbial or quantified):

```
{:John :Plays :Footbal}     :Evaluation       :Well.
{:England :Weather :Rainy} :withProbabilityOf "70%".
```

In RDF, relationships with arity greater than 2 can be expressed by using an intermediary anonymous graph node [21]. Reification is a 4-arity relationship between the 3 components of a statement and the reifying context:

A sensitive problem in this respect is the truth value of statements. We consider it to be an evaluation, so it can be expressed as a reification:

```
{:Sun    :Orbits  :Earth}  :Evaluates  :False.
```

In our proposed model, falseness is essential in the temporary belief sets. Knowledge that evaluates to false will trigger more tutoring (recommendations) on the learning objects involved in the false knowledge. Here, the learning objects are the reified triple components (:Sun, :Earth, :Orbits, with this prioritization). Each learning object has several learning sources (web pages) assigned to it with annotation properties:

```
:Sun   rdfs:isDefinedBy <http://mycompany/lessons/Sun>.
:Sun   rdfs:seeAlso <http://someencyclopedia.com/Sun>.
```

Due to the **open world principle** which is fundamental to the Semantic Web paradigm, every RDF assertion is considered true, while the triples that are absent are NOT considered false, but rather missing (undefined, yet). Our proposed model requires a way of explicitly assigning falseness. Other contexts are also needed to effectively model subjectivity, such as the holder of the temporary belief set. With respect to a single triple, this would require at least a 5-arity relationship.

Explicit knowledge falseness can be accomplished through several artifices, of which the last two are relevant to our proposal:

1. The basic reification pattern, by defining a 4-arity relationship between the assertion elements and the truth value (of false, in this case):

```
_:MyStatement :EvaluatesTo :False.
_:MyStatement rdf:subject :Linda;
              rdf:predicate :PlayedIn;
              rdf:object :Avatar.
```

2. Relying on inference rules provided by restrictive RDF vocabularies of high expressivity, such as OWL:

```
:AvatarActors rdf:type owl:Restriction;
              owl:onProperty :PlayedIn;
              owl:hasValue :Avatar.
:NonAvatarActors owl:complementOf :AvatarActors.
:LindaHamilton rdf:type :NonAvatarActors.
```

(if the restriction occurs in the same knowledge base as :LindaHamilton :PlayedIn :Avatar, it would trigger an inconsistency, which is the basic mechanism of expressing negation in OWL).

3. The solution that we opted for defines negated variants for the predicates:

```
:LindaHamilton :PlayedInNot :Avatar .
```

This solution is problematic since it is:

- *convention-based* (we convene that all properties ending in "Not" are negations of the same properties without the "Not" particle). Of course, from a natural language perspective, it would be more intuitive to place the negation in front (NotPlayedIn, or DidntPlayIn) but keeping in mind that prefixed names are mapped to full URIs (<http://myorganzation.com/myconcepts#NotPlayedIn>) placing the negation particle at the end makes it easier to identify and extract using regular expressions; another variant would be to place a delimited token expressing the truth value:

```
:LindaHamilton :PlayedIn-True  :Terminator.
:LindaHamilton :PlayedIn-False :Avatar.
```

- based on a set of rules expressing *the mutual exclusiveness* of the true and false versions of the predicate. This is only supported by OWL 2.0 through its proposal of disjoint properties. An OWL 1 pattern for describing an aproximation of this exclusiveness is:

```
:AvatarActors rdf:type owl:Restriction;
              owl:onProperty :PlayedIn-True;
              owl:hasValue :Avatar.
:NonAvatarActors rdf:type owl:Restriction;
                 owl:onProperty :PlayedIn-False;
                 owl:hasValue :Avatar.
:AvatarActors owl:disjointWith :NonAvatarActors.
:LindaHamilton rdf:type :NonAvatarActors.
```

(this triple set states that LindaHamilton belongs to a class that does not have common elements with the class of entities who have the PlayedIn-True relationship with Avatar).

An alternative to this pattern of mutually exclusiveness may be provided by production rule systems such as OWLIM, which allows for the customization of rule sets. Custom rules are expressed in a textual format such as:

```
<:PlayedIn-True> <:MutuallyExclusive> <:PlayedIn-False>
```

(this would be a custom axiom)

```
Relationship1 <:MutuallyExclusive>      Relationship2

X   Relationship1     Y

X   Relationship2     Y

Relationship1 <:TriggeredInconsistency> Relationship2
```

(this rule defines a conveniently named inconsistency when a predicate and its negation hold between the same entities).

4. A second viable solution, much more efficient, but limited with regard to inferences, is implemented in the very foundation of the Sesame platform. It proposes the extension of the basic triple structure with a fourth concept – the context.

```
:False :LindaHamilton :PlayedIn :Avatar.
:True :LindaHamilton :PlayedIn :Terminator.
```

The limitations of these are:

- It breaks the RDF model, by employing the unit of quadruple instead of the triple, so it won't be interoperable with strict RDF systems. On the other hand, it is compliant with the RDF query language – SPARQL – which supports the so-called „named graphs". In this respect, :False and :True become graph identifiers: all true assertions can be merged in the context named :True, all false assertions can be merged in the context named :False;
- The context is not supported by the inference engines (of both Sesame and its OWLIM extension). More precisely, one could assign context to every asserted triple at upload time, but all inferred triple lose their context. This means that it's not possible to filter inferred knowledge in the same intuitive and efficient way as to filter asserted knowledge.

The context usefulness is not limited to truth values. It can be used as a replacement for any kind of reification:

```
:Mary :LindaHamilton :PlayedIn :Avatar .
```

This states that the statement of :LindaHamilton :PlayedIn :Avatar belongs somehow to the context Mary (this can be interpreted freely, in this case, as a subjective opinion of Mary).

Furthermore, as the context itself is an RDF resource, it can be subjected to its own description, under a supercontext:

```
:False rdfs:comment "this is the context of all false
beliefs".
:Mary rdfs:comment "this is the set of Mary's beliefs".
```

As reifications can be nested, supercontexts can be applied to contexts:

```
:Ann :Mary rdfs:comment "this is the set of Mary's
beliefs".
```

This states that the comment on :Mary is itself a subjective opinion of Ann (Ann thinks that Mary thinks that....). Consequently, we can combine owners of subjective knowledge with truth values or other types of contexts (spatio-temporal, adverbial, probabilistic etc.):

```
:TempBeliefSet1 :LindaHamilton :PlayedIn :Avatar.
:TempBeliefSet1 :Sun :Orbits :Earth.
:TempBeliefSet2 :LindaHamilton :PlayedIn :Terminator.
:TempBeliefSet2 :Earth :Orbits :Sun.
```

These represent two qualified temporary sets of beliefs. They can be further described through metabeliefs which might be the trainer's evaluations:

```
:Trainer1 :TempBeliefSet1 :Source :Trainee1;
                          :Evaluates :False.
:Trainer2 :TempBeliefSet2 :Source :Trainee2;
                          :Evaluates :True.
```

(the two belief sets could come from the same source, thus separating the correct knowledge from the wrong knowledge of a given trainee).

Obviously, there's no limit to the number of levels on which we can develop the knowledge qualification, in a much more efficient manner than using previously described techniques. And, we can use the named graph feature when querying the knowledge base, using the contexts as graph names:

```
SELECT ?X ?Y ?Z
WHERE
{
GRAPH ?G {?X ?Y ?Z}
GRAPH :Trainer1 {?G :Evaluates :False}
}
```

This query will extract all the knowledge that was evaluated as false according to Trainer1. The evaluations could be further filtered based on the trainee, subject or other criteria, limited only by the SPARQL capabilities. By using contexts as graph names, the query engine provides a native way of filtering knowledge based on subjectivity and correctness.

The temporary belief sets can also be qualified by temporality (timestamps), for any given trainee. This expresses how the belief set shifts from one state to another, thus recording the progress towards the ideal state. The state shifting is recorded during an on-line evaluation, while the trainee takes an on-line multiple choice test. Each question is backed by learning objects consisting in RDF assertions that express the knowledge behind the question (usually the subject and predicate maps on the question, while the object maps on the correct answer but this is not mandatory, the tutor or a knowledge engineer working with the tutor may assign more complex RDF triple sets, depending on the granularity desired for the learning objects with respect to the question meaning). The RDF assertions are qualified by its source (the tutor who defined the question) and the truth value assigned by the source. Thus, every question is represented in the system's knowledge repository.

When the trainee selects an answer, the corresponding assertions are also qualified by his identity. After such an evaluation, the system produces the trainee's temporary belief set, split in false and correct subgraphs according to the tutor's qualifications. From the false subset, the learning objects are linked to their defining pages/lessons, which are recommended for more in-depth reading.

As previously mentioned, we need to employ both the techniques 3 and 4 as long as OWLIM doesn't adopt the underlying Sesame's contextualization model at production rule level. Inferences (or even rules) limited to a given context would be of great help in delimiting subjective knowledge in an intuitive way (and, more important, in detecting inconsistencies of subjective knowledge).

4.2 Knowledge Dependencies

Usually the concepts of a learning content are linked through various degrees of dependencies: in order to understand concept x, you have to understand concept y, and so on. Learning object dependencies are easy to express in RDF, using a taxonomy of prerequisites for each learning object. This allows the system to evaluate the trainee gradually – by displaying questions as they are enabled by the fact that the trainee acquires the prerequisite concepts. From the trainee's point of view, this is perceived as a usability feature, as it affects the "feel" of a learning process – reading/learning is guided by wrong answers in the temporary belief set and by concept dependencies.

An initial dependency taxonomy (further refined by the knowledge engineer) is generated using a CONSTRUCT query which creates a (Subject-DependsOn-Object) triple for every learning object (if the Object is a resource):

```
:Earth rdf:type :Planet .
:Planet rdfs:subClassOf :CelestialBody .

CONSTRUCT {?x :DependsOn ?y}
WHERE {GRAPH :Trainer1
             {?x ?predicate ?y. FILTER isURI(?y)}
```

OWL restriction classes are used for grouping categories of enabled/enabling questions and answers and their concepts. A sample of the triplestore is expressed as follows (prefixes avoided for brevity; partially inspired by an example from [19, chapter 10]):

```
:PossibleAnswer a owl:ObjectProperty;
    rdfs:domain :Question;
    rdfs:range :Answer.
:AnswerText a owl:DatatypeProperty;
   rdfs:domain :Answer;
   rdfs:range xsd:string.
:QuestionText a owl:DatatypeProperty;
    rdfs:domain :Question;
    rdfs:range xsd:string.
:Enables a owl:ObjectProperty;
    rdfs:domain :Answer;
    rdfs:range :Question.
:SelectedAnswer a owl:ObjectProperty;
    rdfs:subClassOf _:x.
_:x a owl:Restriction;
    owl:onProperty :Enables;
    owl:allValuesFrom :EnabledQuestion.
:AnsweredQuestion a owl:Restriction;
    owl:onProperty :SelectedAnswer;
    owl:someValuesFrom :Answer.
```

```
:DependsOn a owl:ObjectProperty;
   rdfs:domain :LearningConcept;
   rdfs:range :LearningConcept.
:EvaluatedBy a owl:ObjectProperty;
   rdfs:domain :LearningConcept;
   rdfs:range :Question.
:AcquiredConcept a owl:Restriction;
   owl:onProperty :DependsOn;
   owl:allValuesFrom :CorrectAnsweredConcept.
```

The store is managed and queried in order to extract contextual semantics based on the current learning objects approached by the student. The semantics control the question flow (by making sure the content is not delivered unless the student acquires all prerequisite concepts).

4.3 General Architecture

All queries are executed over Sesame's REST interface, via HTTP requests triggered by Python's urllib module. Data is returned as JSON, easily parsable in Python dictionaries for further processing. Due to the limitations of the open world notion of inequality, aggregate computations (such as counting the wrong/correct triples) are not supported by the (current) implementation of SPARQL [22], so they are executed with Python, over the query result dictionaries.

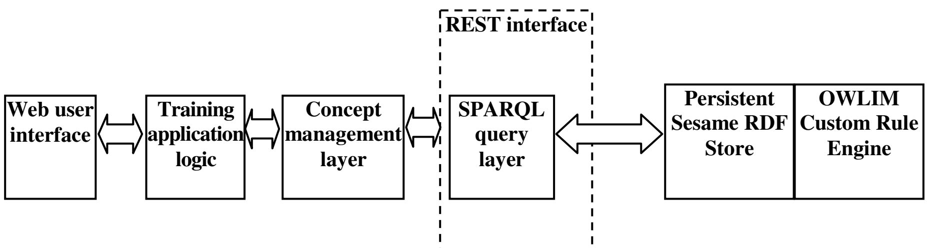

Fig. 1. The general architecture

In this respect, the proposed architecture involves a concept management layer, working as a wrapper for SPARQL queries, applying:

- pre-processing to the data extracted from the interface (building the belief sets to be uploaded on Sesame based on the trainee evaluation or the knowledge engineer interface)
- post-processing to the SPARQL results (generating recommendation links based on incorrect or unstable beliefs, displaying inferred inconsistencies); further development is needed to integrate on this level a delta analysis module for detecting differences between the temporary belief sets of the same trainee.

5 SWOT Evaluation and Conclusions

Strengths

The paper presents a low-cost methodology for representing temporary, subjective knowledge which can be tracked during the learning process, in the context of a computer-based training system.

The proposed solution adds value to learning systems by monitoring closely the evolution of a knowledge portfolio of a trainee and by optimizing the training process with suggestions of content directly related to the trainee's detected insufficiencies.

Weaknesses

In order to respect the low cost requirement, the model employs the free version of OWLIM, SwiftOWLIM, which does not support inconsistency detection. This is compensated by the possibility of customizing rulesets using an intuitive syntax for Horst rules. The lack of OWLIM's support for contextualization of inferences complicates the management of inferences over subjective knowledge.

Opportunities

The emergence of Semantic Web, although slow, will greatly affect the field of e-learning, which must prepare for this and adopt as soon as possible the semantization of content.

Threats

E-learning has its typical threats regarding student motivation, technology adoption, arbitration and security. A more specific threat is related to the slow adoption of Semantic Web technologies in Web development in general, as legacy systems are considered sufficient for the current requirements of eliminating the spatial and temporal costs of learning.

The paper presented some issues regarding the representation of subjective knowledge and its implications on learning systems and learning objects within the context of a computer-based training system architecture based on semantic storage.

Future efforts will be invested in more advanced processing of the temporary belief sets aimed mainly at pattern detection and the relationship between the evolution of belief sets and the concept dependencies.

Acknowledgment. This paper represents results of the research project 2443/2009 funded by the UEFISCSU-CNCSIS Program IDEI, managed by Lect. Robert Buchmann Ph.D.

References

1. The Resource Description Framework, `http://www.w3.org/RDF/`
2. Höver, K.M., Steiner, C.M.: Adaptive Learning Environments: A Requirements Analysis in Business Settings. International Journal of Advanced Corporate Learning 2(3), 27–33 (2009)

3. Beck, R.: What are Learning Objects?, http://www4.uwm.edu/cie/learning_objects.cfm?gid=56
4. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 34–43 (2001)
5. Moodle – the Official Website, http://moodle.org/
6. Blackboard – the Official Website, http://www.blackboard.com/
7. The Web Ontology Language, http://www.w3.org/2004/OWL/
8. Bishop, B.: Primer Introduction to OWLIM, http://www.sirmaitc.com/confluence/display/OWLIM/Primer+Introduction+to+OWLIM
9. Gonzalez-Videgaray, M., Hernandez-Zamora, G., del Rio-Martinez, J.H.: Learning Objects in Theory and Practice: A Vision from Mexican University Teachers. Journal Computers & Education 53(4), 1330–1338 (2009)
10. The IEEE Learning Object Metadata Standard, http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
11. Bennacer, N., Bourda, Y., Doan, B.: Formalizing for Querying Learning Objects Using OWL. In: Looi, C.T., Sutinen, E., Sampson, D.G., Aedo, I., Uden, L., Kähkönen, E. (eds.) Proceedings of the IEEE International Conference on Advanced Learning Technologies, ICALT 2004, pp. 321–325. IEEE Computer Society, Washington (2004)
12. Bouzeghoub, A., Defude, B., Ammour, S., Duitama, J.F., Lecocq, C.: A RDF Description Model for Manipulating Learning Objects. In: Looi, C.T., Sutinen, E., Sampson, D.G., Aedo, I., Uden, L., Kähkönen, E. (eds.) Proceedings of the IEEE International Conference on Advanced Learning Technologies, ICALT 2004, pp. 81–85. IEEE Computer Society, Washington (2004)
13. Ahmed, S., Parsons, D., Ryu, H.: Supporting Adaptive Learning Interactions with Ontologies. In: Ryu, H., Champion, E. (eds.) Proceedings of the 11th International Conference of the NZ Chapter of the ACM Special Interest Group on Human-Computer Interaction, pp. 17–24. ACM Press, New York (2010)
14. Jovanovic, J., Gasevic, D., Devedzic, V.: TANGRAM for Personalized Learning Using the Semantic Web Technologies. Journal of Emerging Technologies in Web Intelligence 1(1), 6–21 (2009)
15. Open RDF Official Website, http://www.openrdf.org/
16. OWLIM Official Website, http://www.ontotext.com/owlim/
17. SPARQL Official Website, http://www.w3.org/TR/rdf-sparql-query/
18. The Serialization of SPARQL Query Results in JSON, http://www.w3.org/TR/rdf-sparql-json-res/
19. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist. Morgan Kaufmann, San Francisco (2008)
20. N3 Resources, http://www.w3.org/DesignIssues/N3Resources
21. Defining N-ary Relations on the Semantic Web, http://www.w3.org/TR/swbp-n-aryRelations/
22. SPARQL Issues, http://www.w3.org/2001/sw/DataAccess/issues#countAggregate