Reasoning-enabled Semantic E-Learning Approach

Martin Jovanović[1] and Dejan Todosijević[2]

Abstract – **This paper discusses a collaborative e-learning framework, based on Semantic Web concepts, which allows a certain degree of automated reasoning. The proposed framework is an extension to the existing DSi 2.0 framework developed at the Computer Science Department of the Faculty of Electronic Engineering Niš, Serbia. This framework has bi-directional communication with the learner, allowing them to alter the course material semantics. With each user interaction, implicit semantic data may occur. This data is inferred via the reasoner and added to the semantic layer for faster retrieval. Expected challenges have been discussed and the optimal solution, within given constraints, have been proposed.**

Keywords – **e-learning, semantic web, dsi, reasoning.**

I. INTRODUCTION

A. E-learning

The framework model discussed is being developed primarily as an e-learning tool. As such, it is aimed at satisfying several key aspects of e-learning, such as just-in-time approach that includes value chains [1], or delivery of individualized, dynamic real-time material [2], just when it's needed, and in quantities needed [3]. Though these requirements appear rather demanding, dynamic aggregation of learning units can be achieved by means learning objects [4], small units of instructional material that can be combined to form lessons and courses. In order to achieve that, each learning object must contain rich metadata, possible combinations of learning objects require specialized data structures and decision-making algorithms. This is why the tutoring systems that perform these operations are sometimes defined as the application of artificial intelligence on education. [5] However, artificial intelligence alone is not sufficient. It is only the Semantic Web that can provide all the expressivity and tools needed to put these concept into motion.

B. Semantic E-learning

The list of applications of Semantic Web tools and technologies in e-learning is comprehensive. [6] Envisioned as the "web of machines", with machine-understandable (rather than solely human-understandable) data - at least within an additional data layer - this approach is aimed at data integration (from various sources) as well as autonomous decision-making. [7] This approach accurately maps to the autonomous on-demand assembly of learning material from various learning object repositories, under well-defined conditions, with machines equipped with sufficient reasoning power. This, in turn, requires languages for expressing semantics, data structures (ontologies) and reasoning algorithms. The system described in this paper (DSi) benefits from one subset of these technologies in order to extend interactivity and accelerate the learning process - with a certain degree of autonomous reasoning ability.

II. DRAG AND DROP SEMANTIC INTERFACE

A. DSi 1.0

The Drag and Drop Semantic Interface (DSi) was first presented in 2007 [8] and developed in version 1.0 in 2009. [9] It is web-based e-learning framework (or concept), rather than a stand-alone application. It consists of two layers: a textual learning material and a semantic layer. Semantic document (in RDF/XML format) holds relations between notions in the textual material (notions are mapped to single words in the text), expressed as RDF triples. An RDF triple example, asserting that *Grinder is a cofounder of NLP*. The foaf namespace has been used during development and is of no particular relevance.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:eg="http://example.org/foovocab#"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
 <foaf:Person rdf:nodeID="Grinder">
<foaf:name>Grinder</foaf:name>
<eg:was_a_cofounder_of rdf:nodeID="NLP"/>
</foaf:Person>
</rdf:RDF>
```

Fig. 1. One RDF/XML statement example

On page load, all the words from the text that exist in the RDF document (as statement subjects or objects) are assigned drag-and-drop capability. By virtue of dragging one word and dropping it onto another, the learner can query the semantic document for any relations that might exist between the two chosen words, as shown in Fig. 2. The first version was implemented completely on the client side.

[1]Martin Jovanović is with the University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: martin.jovanovic@elfak.ni.ac.rs
[2]Dejan Todosijević is with the University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš.

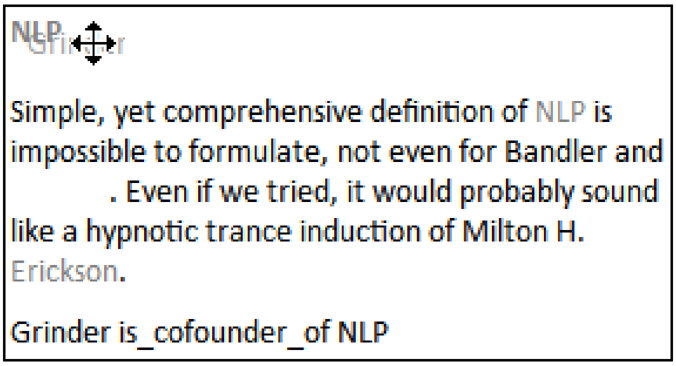

Fig. 2. DSi 1 typical user interaction

In the updated version (DSi 1.5) the web application was ported to PHP and the semantic layer was transferred to the server. [10] The DSi 1.5 demo application[1] is shown in Fig. 3. This was done primarily because the semantics is what adds value to the learning material and may be subject to corporate restrictions. In this version, the user interaction with the system (drag-and-drop action) invokes a server-side function that queries RDF and returns only relations for the chosen word pair. Exhaustive querying is possible though, but expensive due to the slow nature of GUI interaction. In this version, unsuccessful interactions (drags and drops of words that have no relation between one another) are prevented by additional marking of the related words: on the drag action, all words that are in relation with the one dragged are retrieved and marked (by color, underline or any appropriate CSS attribute). Still, the word can be dropped on any droppable, with or without a relation.

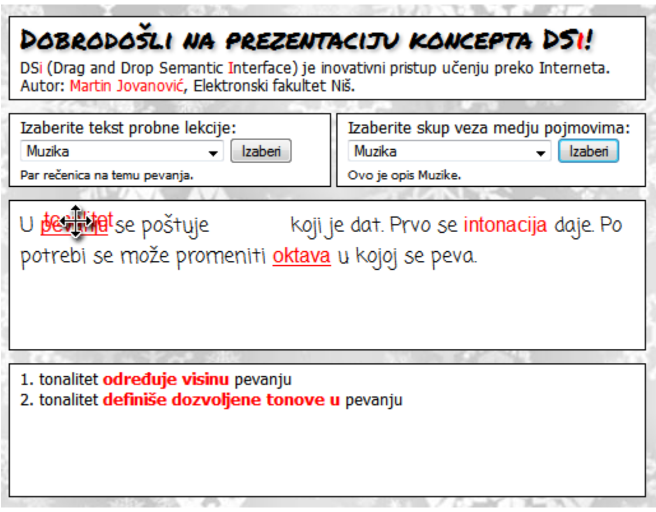

Fig. 3. DSi 1.5 demo application

Probably the main characteristic of both DSi 1.0 and 1.5 is that the communication with the learner is uni-directional: the learner can only retrieve information from the system. All the semantic data must be predefined by instructional designers, and expressed in the proper RDF/XML form. This can be done using any of the popular ontology editors, such as Protégé, [11] Semantic Works (discontinued), [12] TopBraid, [13] NeOn [14] etc. This approach requires the instructional designers (domain experts, but not necessarily from the IT domain) to have certain technical knowledge - which is rather limiting for wide use.

The other important characteristic of DSi is its lexical approach to semantics: both notions and relations are pure text. Notions are single words from the text, while relations are freeform human language expressions - human, but not machine-understandable. This adds to system simplicity, but subtracts from additional semantic options, such as reasoning.

B. DSi 2.0

The latest implemented version of the framework (DSi 2.0) takes a significantly different approach to interactivity - it is bi-directional. In this version, not only can the learner query the document semantics, but they also can add new relations - thus contributing to the semantic layer. This was achieved by splitting the framework functionality into two modes: read mode and edit mode (shown in Fig. 4). Edit mode is different in sense that all the words from the learning material are set as drag-droppable (by wrapping each word in a separate HTML span tag). From the user's perspective, not only marked - but all words can be dragged and dropped onto any. Highlighting is used to indicate words that the chosen (dragged) word is related, to avoid zero-result queries. On each drop, current relations are displayed but a textbox for adding a new relation is shown also. Learner is free to add as many relations as they want. The system only checks for duplicate relations (in lexical terms - as identical strings).

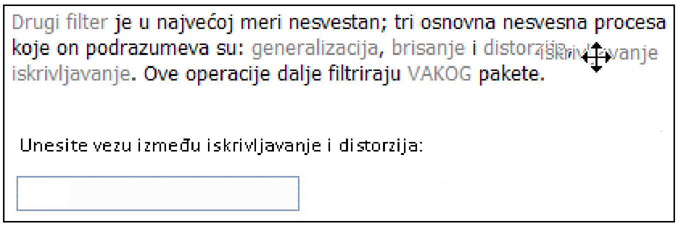

Fig. 4. DSi 2.0 edit mode interaction

DSi 2.0 aims at introducing the Web 2.0 approach to learning. Users can freely contribute to the semantic knowledge base, learning can gain collaborative dimension, and by introducing grading to students' relations - this system leads to a form of peer assessment. However, in terms of relations, it doesn't differ from its predecessor. Relations are freeform strings, expressing only human-understandable meaning - flat to machines. As mentioned before, this approach has been chosen for simplicity while the system is in early testing phases. Moreover, the framework was developed with students of various fields in mind - who cannot be expected to put much time and effort in defining precise

[1] Demo application URL: dsi.martin.rs (20.03.2013.)

relations, not to mention giving them in a strictly formal way. However, it significantly limits application of machine power on the level of semantics.

III. Reasoning in DSi 2.0

A. Expression and Reasoning

In order to infer any conclusion, the data upon which the reasoning is performed must be well-structured and expressed in a reasoner-understandable way. Current integration of Semantic Web (as of December 2012) offers 4 layers of expressiveness, in the form of 4 languages: RDF, [15] RDFS, [16] OWL [17] and OWL 2. [18] These languages stack expression wise in respective order.

RDF is the simplest among those and offer assertional knowledge representation, such as class membership and property membership. This language has been chosen for DSi for two main reasons: simplicity of implementation (and existence of a PHP-based framework to manipulate it), [19] and simplicity of user interactions with the system - remaining firmly in the domain of simple human expressions.

More complex expressions, such as taxonomy (hierarchy of classes) require RDF Schema (RDFS). This language also allows for property hierarchy, as well as defining domains and ranges, all of which gives more expressive power, yet limited in some aspects - probably most important of which being relation properties.

OWL inherits RDF in tools to assert fact and RDFS in class and property structuring, but expands them in several important directions. It can define new classes as unions, intersections or complements of the existing, closed classes,

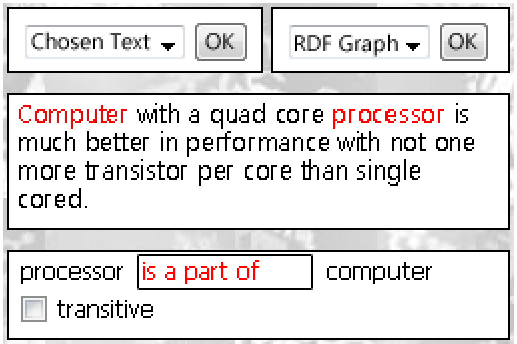

Fig. 5. User experience change

build new classes by means of quantifiers, cardinality restrictions or class, property or individual names. It also allows inference of the existence of a property by chaining existing ones. Properties can also be defined as inverse or disjoint with another properties. OWL was layered into 3 sublanguages: OWL Lite, OWL DL and OWL Full, with both expressiveness and computational time rising respectively.

The second iteration f OWL (known as OWL 2) inherits the OWL in sense that any valid OWL ontology is a valid OWL 2 ontology, and adds new features like disjoint union of classes, keys, property chains, data ranges, asymmetric, reflexive and disjoint properties and enhanced annotation capabilities. It is defined through three profiles: EL, QL and RL, any of which are trimmed versions of OWL 2 that trade some expressive power for reasoning efficiency.

B. The Reasonable Framework Architecture

All expressive power offered by these languages allows for precise description of various concept - and any relation in the learning material can be described in a well-formed manner, allowing full reasoning power. This, however, contrasts with principles of collaborative learning - the very aim of DSi 2.0 approach - in which the freedom of form in contributing is one of the key motivational factors, and where too much of superimposed form proves discouraging and demotivating. [20] The proposed framework architecture is balanced between these two opposed requirements: sufficient form and expressiveness to allow reasoning, and sufficient simplicity and freedom of human expression as the motivational aspect. Given the constrains, the chosen reasoning enabler is the transitivity. The main reason for this is the following: from the user's perspective, relational properties are straightforward: they can be defined in an immediate fashion, for one specific relation in question, without lateral references (such as sets/classes related entities might belong to); the user doesn't have to "look around" the context and find other links that the current relation they are adding (or entities it connects) might have with it - this would impose additional cognitive load and time, and thus demotivate. Even if we consider a property as simple as the inversion (one relation being inverse to another), this involves two relations, i.e. up to four entities (notions from the text), so the learner cannot stay on their contributed relation, but instead needs to refer to another one (or more). Relational property of choice for the initial "reasonable" version of DSi is *transitivity*; by defining a relation as such, reasoner can easily infer implicit relations, while the user interactions remain as simple as possible.

Neither RDF nor RDFS can express transitivity - the farther the RDFS goes is class and property taxonomy. In order to define transitivity, one must resort to the first version OWL.

C. User Experience and Semantic Changes

From the user perspective, system will undergo one small change: when prompted to enter a new relation, user is offered a checkbox for transitive relation property (yes or no), as shown in Fig. 5, and additional help line may be included. This way, action required consists of entering a new relation and checking (or not) that it is by nature transitive.

Corresponding semantic language change consists of a new OWL element: owl:ObjectProperty. This statement adds transitivity to the newly-added relation. In the example case, the statements are as following:

```
<rdf:Description rdf:ID="processor">
  <is_a_part_of rdf:resource="computer" />
</rdf:Description>
```

```
<owl:ObjectProperty rdf:ID="is_a_part_of">
  <
   rdf:type
   rdf:resource="&owl;TransitiveProperty"
  />
</owl:ObjectProperty>
```

The first statement asserts that the processor is a part of a computer. This is the relation the user has added to semantics. For simplicity case is ignored, and plural has been deliberately avoided in the example sentence.

Should the same or another user add the same relation between other two notions (e.g. transistor and processor), two statements would be present:

```
<rdf:Description rdf:ID="processor">
  <is_a_part_of rdf:resource="computer" />
</rdf:Description>

<rdf:Description rdf:ID="transistor">
  <is_a_part_of rdf:resource="processor" />
</rdf:Description>
```

Combined with the transitivity of this relation, it can be inferred that transistor is a part of computer.

D. Reasoners

W3C maintains a list of OWL compliant reasoners, most of which are free and open-source. [21] However all available reasoners are implemented in Java, mostly due to the availability of OWL API for this language. Since the DSi framework was developed in PHP, bridging between the application and the reasoner is possible either with a Java enabled server, or through evoking a reasoner on a remote server (e.g. through the DIG protocol). [22] The implementation with least technical challenges was chosen - the reasoner setup within the local (Apache) server, wrapped in the CGI. This approach doesn't require Java servlets, which are not necessary for it is only the reasoner I/O that is required for the semantic document update.

IV. Conclusion

In this paper an extension to the DSi 2.0 semantic-based collaborative e-leaning framework, that adds reasoning capability, has been discussed. The current DSi 2.0 framework, as well as the previous DSi 1.0/1.5 versions, on which the current proposed architecture is based, have been briefly described. In building the proposed architecture, two opposed requirements have been followed: the system should have enough expressiveness to enable reasoning (the more the better), but users - learners - must not be presented with too complex tasks when contributing to semantics, to avoid demotivation. The proposed solution is transitivity: for each contributed relation, learners are offered a checkbox to confirm that the new relation they add is by its nature transitive. This is a non-invasive approach that is expected to show the applicability of automated reasoning in systems based on the DSi approach. In order to achieve this, a bridge from PHP (DSi) to Java (reasoner) must be obtained. Further development may include adding other relation properties (reflexivity, symmetry) and/or equal/inverse properties, but with constant caution not to introduce too much complexity into the interaction with learners.

References

[1] Stojanovic Lj., Staab, S. i Studer, R. - Elearning Based on the Semantic Web, WebNet2001 - World Conference on the WWW and Internet, 2001, pp. 23-27.

[2] Drucker, P. - Need to Know: Integrating e-Learning with High Velocity Value Chains, A Delphi Group White Paper, http://www.delphigroup.com/whitepapers/pdf/20001213-e-learning-wp.pdf

[3] Horton, W. – Leading E-Learning: Here is how You Can Chart your Course, Champion Implementation, Ensure Success, American Society for Training and Development, 2001, ISBN 1562862987.

[4] Downes, S. – Learning Objects: Resources for Distance Education Worldwide, International Review of Research in Open and Distance Learning, ISSN 1492-3831, Vol 2, No. 1, 2001.

[5] Samuelis, V. – Notes on the Components for Intelligent Tutoring Systems, Acta Polytechnica Hungarica, Vol. 4, No. 2, 2007.

[6] Devedžić, V. – Education and the Semantic Web, International Journal of Artificial Intelligence in Education 14 (2004).

[7] Tim Berners-Lee – Plenary at WWW Geneva 94 – First International World Wide Web conference, CERN, Geneva, September 1994. Slides available online at the following address: http://www.w3.org/Talks/WWW94Tim/

[8] Jovanović, M. - "Arhitektura konceptualno-orijentisanog alata za podršku učenju", Infoteh¬-Jahorina, Proceedings, Jahorina, 28-30. 03. 2007, Vol. 6, Ref. E-IV-4, pp. 477-479, issued by Elektrotehnički fakultet Istočno Sarajevo, 2007, ISBN 99938-624-2-8.

[9] Jovanović, M., "Semantička nadgradnja nastavnog materijala u sistemima za elektronsko učenje", M.Sc. thesis, University of Nish, Faculty of Electronic Engineering, 2009.

[10] The DSi 1.5 framework has been accepted as a Technical Solution at the University of Niš, Faculty of Electronic Engineering, doc. number 07/10-004/13-001, 17.01.2013.

[11] http://protege.stanford.edu/

[12] http://www.w3.org/2001/sw/wiki/SemanticWorks

[13] http://www.topquadrant.com/products/TB_Composer.html

[14] http://neon-toolkit.org/wiki/Main_Page

[15] http://www.w3.org/RDF/

[16] http://www.w3.org/TR/rdf-schema/

[17] http://www.w3.org/2004/OWL/

[18] http://www.w3.org/TR/2012/REC-owl2-overview-20121211/

[19] In the current DSi implementation, the RAP - RDF API for PHP framework has been used. This framework can be retreived at: http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/

[20] P. Dillenbourg, Over-scripting CSCL: The risks of blending collaborative learning with instructional design, in Kirschner, Paul A. (Ed.), Three worlds of CSCL. Can we support CSCL?, Open Universiteit Nederland, 2002, pp. 61–91.

[21] http://www.w3.org/2007/OWL/wiki/Implementations

[22] http://protegewiki.stanford.edu/wiki/Using_Reasoners