Journal of Web Engineering, Vol. 7, No. 1 (2008) 025–041

REASONING ON THE SEMANTIC WEB FOR ADAPTIVE HYPERMEDIA

LYDIA SILVA-MUÑOZ[a] KARINA MEDINA MARCOS MARSICANO MARIO BONJOUR

Instituto de Computación, Universidad de la República
Montevideo, Uruguay
silva@cs.toronto.edu {medinakari, marsicano, mariobon}@adinet.com.uy

JOSÉ PALAZZO M. de OLIVEIRA

Instituto de Informática - Universidade Federal do Rio Grande do Sul
Rio Grande do Sul, Brasil
palazzo@inf.ufrgs.br

Received November 13, 2006
Revised February 4, 2008

So far, ontologies have been widely used to convey knowledge across the Semantic Web. Complementing web ontologies with Horn-like rules to assert relations among ontology individuals and properties is part of the ongoing implementation of the Semantic Web. Intelligent Web Adaptive Hypermedia Systems *(AHS)* are the next generation for adaptive hypermedia on the web. We present a web-based intelligent AHS for e-learning that configures on the fly complex learning objects tailored to the user profile. This automatic configuration is entirely accomplished by reasoning over a hybrid *Knowledge Base (KB)* composed of ontologies, and Horn-like rules defined on top of ontologies concepts. Interoperability on the semantic level is achieved by using an *application profile* of standard vocabularies, standard languages for the representation of ontologies and rules, and a standard interface for reasoning functionality.

Keywords: Semantic Web, Adaptive Systems, Reasoning, Ontologies, Standard Metadata, OWL, SWRL
Communicated by: G.J. Houben & K. Turowski

1 Introduction

Adaptive Hypermedia Systems (AHS) build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user in order to adapt to the needs of that user [1]. The Semantic Web consists of adding machine understandable pieces of knowledge related to web content so that automatic agents can behave as if they were able to understand the semantics of such content. Ontologies for the Semantic Web formally define structural vocabularies to be used in semantic markup, are situated on the Web, and are encoded in web standard languages such as the *Ontology Web Language (OWL)* [2]. *Description Logics (DL)* [3] is the formalism that underlies OWL. DL based languages are well suited to represent the structural properties of a domain that support reasoning involving entire classes. However, they are not sufficiently expressive to represent relations among ontology individuals. Therefore, instance reasoning is not possible on them.

[a]The current filiation of the author is *Department of Computer Science - University of Toronto*

On the other hand, Horn-like clauses can capture relationships between composite properties and make the manipulation of individuals possible, addressing the reasoning about instances that DL lacks [4]. The combination of the two paradigms, namely, ontologies and rules, is necessary for practical purposes. Some early DL systems that included rules on top of ontology concepts are KRYPTON [5], BACK [6], Classic [7] and CARIN [8]. The Semantic Web Rule Language (SWRL) [9] is the language recently proposed to represent Horn-like rules on top of ontologies for the Semantic Web. The existing reasoners [10] that address this very expressive combination, which is encodable in SWRL, are not feasible for practical purposes, because such a combination is undecidable [4][9]. However, decidable results can be obtained by separating reasoning in the structural component from reasoning in the rule component of a KB[11][12]. This approach encourages the combined use of pre-existing rule-based reasoners with up-to-date DL reasoners [13] [14] to address hybrid inference. Following this approach, our architecture deals with structural inference by the use of the RACER reasoner [13], and with rule-based inference, by the use of the Jess rule engine [15]. With respect to interoperability and scalability, we used the standard interface DIG Description Logics application program interface (DIG) [16] to ask and tell our DL KB. This enables us to use any up-to-date DL reasoner compliant with the standard DIG. AHS for e-learning have a Student Model that contains the style and preferences of each student and a model of the educative content. According to [20], an AHS for e-learning has an Information Space consisting of a *Hyperspace* of web pages with educative content and also a *Knowledge Space* telling the system which topics of the Domain being taught are addressed by each Hyperspace element. We have added some elements to the Knowledge Space. These extra elements convey structure and other metadata under the form of a set of axioms used for the on the fly computation of tailored learning objects. Since our system was envisioned to interoperate on the Semantic Web, i.e., to be used by Semantic Web agents, we make the knowledge about our e-learning data available in standard for the web terms. These terms are part of a terminology defined as an application profile that extends the terms defined in standard pre-existing vocabularies for the e-learning domain [17][18] [19]. Web agents can understand our OWL terminology, or can interpret each of its terms according to the semantics of their root standard terms. Since we have our terminology represented in OWL, which is a language envisioned to work under the Open World Assumption (OWA), the interchange of knowledge on the Web under the premise that the involved sources have non comprehensive information is possible. This is a necessary approach for the Semantic Web environment. On the other hand, we also take advantage of the Closed World Assumption (CWA) of the Datalog paradigm underlying the rule-based component of our KB to make the inference that only concerns our system. This approach is consistent with the vision given in [28]. This paper shows the decisions taken and the difficulties we had to overcome when addressing hybrid reasoning to achieve adaptation to user's profiles during the design and implementation stages of an Adaptive Hypermedia System for the Semantic Web. The remainder of this work is structured as follows. Section 2 introduces related work and contextualizes our contribution. Section 3 gives a brief account of existing approaches for addressing reasoning on the Semantic Web. Section 4 shows the built hybrid knowledge base, section 5 gives details about the hybrid inference that we carried out, and section 6 is a short conclusion.

2 Related Work

Adaptive Hypermedia Systems for e-learning use ontologies to agree on the semantics of the terms that their community of users share. Taxonomies are used to semantically relate the terms used in the areas of knowledge to be taught in their courses, and formal ontologies are used to represent knowledge about their typical activities, participants and resources [29]. In order to interoperate across the Web on the semantic level, metadata standards and web ontologies are needed. That interoperation is intended to facilitate search, evaluation, acquisition, and wide use of learning objects for example, for learners, instructors or automated software processes. In this context, namespace schemas, such as the Dublin Core schema [30] provide metadata standards for the Web. A namespace schema is a set of definitions for metadata elements that stand on the Web as reference points to be used to create metadata descriptions about resources of a specific domain in a standardized way. Generally, a schema is designed for a registration authority, and maintained as a stable reference on the Web. Such a design is made following a minimalist approach, which implies the use of a minimum set of elements with simple structure in order to facilitate the adoption of the schema by the community. Then, each particular community develops an application profile that tailors the standard to the community requirements while retaining interoperability with the original schema. In particular, the IEEE Learning Object Metadata (LOM) [25] standard provides a taxonomy for the e-learning domain composed of those terms needed to describe instructional content, learning objectives, instructional software and software tools, persons, organizations and events referenced during technology supported learning. Since XML support has reached a mature level, a number of projects have implemented their application profiles using XML bindings for LOM metadata, i.e., they have a specific XML Schema designed to validate documents with LOM descriptions of learning objects. Examples of projects using LOM metadata in XML are ARIADNE (http://www.ariadne-eu.org), Can-Core (http://www.cancore.org), SCORM (http://www. adlnet.org) and Heal (http://www.healcentral.org). However, XML is not able to convey the semantics of data structures by itself like RDF does. This means that applications that interoperate based on XML bindings need to agree in advance about the semantics of the terms of their application profiles. Conversely, the fundamental advantage of having an application profile based on RDF lies in the fact that RDF provides a standard way to represent meaning, making the realization of the machine understanding concept possible, and enabling the reutilization of pre-existing vocabularies, like Dublin Core. Authors in [25] give detailed explanations of the differences between XML and RDF bindings for LOM standard metadata.

We have developed an application profile based on RDF on top of which all our definitions were done. However, our novel approach consists of not only sharing domain knowledge through a RDF based application profile, but also of sharing all the knowledge needed for solving the problem of dynamically configuring a personalized learning object. All the machinery that we use for inference purposes, i.e. our definitions, axioms and rules, is formally represented in a machine understandable way, which means that any web agent SWRL / OWL DL aware, can unambiguously understand, and act, based on them. Following is a brief description of the strengths and weakness of the standard languages currently available to address reasoning on the Web.

3 Reasoning Approaches for The Semantic Web

Description Logics, the formalism underlying the ontology language for the Semantic Web OWL, structures the domain into categories called *classes* or *concepts* whose individuals hold complex predicates. Relations among concepts are called *properties or roles*. In terms of a correspondence with First Order Logic (FOL), atomic concepts can be seen as unary predicates, properties as binary predicates, and individuals as constants. Figure 1 shows a part of our ontologies where the relations between each course, its topics, and the material supporting their explanation are depicted.

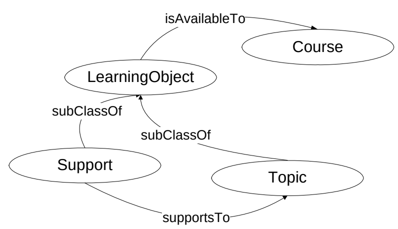

Fig. 1 Courses and Topics

We formally define `Support` as the class of individuals that are related to an element of class `Topic` by relation `supportsTo`:

$$Support \equiv \exists\ supportsTo.Topic$$

We provide an axiom to characterize class `Course` stating that its elements are those related by the inverse of relation `isAvailableTo` with individuals of classes `Support` or `Topic`.

$$Course \equiv \exists\ isAvailableTo^{-}.(Support \sqcup Topic)$$
$$TopicsForCourseX \equiv Topic \sqcap \exists\ isAvailableTo.\{courseX\}$$

We can query a DL KB to ask if a given individual satisfies a certain concept or ask for the entire set of individuals satisfying a concept, among others. A detailed account of DL can be found on [3]. DL reasoners work under the *Open World Assumption (OWA)*, this means that the knowledge in the KB is assumed to be incomplete, i.e. nothing can be inferred about what is not explicitly stated. For example, given a student, say `student1`, a course named `course1` composed of `topic1, topic2` and `topic3`, and having `hasKnowledgeOn(student1, topic1)` as the unique instance pertaining to relation `hasKnowledgeOn`, we are not allowed to conclude that `student1` has no knowledge of `topic2` and `topic3`. Such a conclusion is not possible under OWA because there is no explicit statement saying that the individual `hasKnowledgeOn(student1, topic1)` represent all the existing individuals of the relation `hasKnowledgeOn`. In absence of such a statement, the supposition is that they may exist, but our KB has not been told about them, i.e. it is assumed that our KB does not have a comprehensive account of the world that it represents. On the contrary, working under Closed World Assumption (CWA), particularly using negation as failure, which is the assumption of database systems, we should conclude from the same KB that `student1` only has knowledge of `topic1`, because we assume that our universe is closed to what our KB knows.

On the other hand, SWRL was designed to be the rule language for the Semantic Web. It is based on a combination of the OWL DL and OWL Lite sublanguages of OWL with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language [9]. SWRL rules are of the form of an implication between an antecedent (*body*) and a consequent (*head*). Rules with conjunctive consequents can be divided into multiple rules each with an atomic consequent of the form: $B1 \wedge B2 \wedge ..Bn \rightarrow H$, which resembles the Horn rules of FOL. Atoms in these rules can be of the form `C(x)`, `P(x,y)`, `sameAs(x,y)`, `differentFrom(x,y)`, or `builtIn(r,x,...)` where `C` is an OWL description, `P` is an OWL property, `r` is a built-in relation, and `x,y` are either variables, OWL individuals, or OWL data values. Informally, an atom `C(x)` holds if x is an instance of class description `C`, an atom `P(x,y)` holds if `x` is related to `y` by property `P`, and an atom `builtIn(r,x,...)` holds if the built-in relation `r` holds for the interpretations of the arguments. For the sake of decidability only variables that occur in the antecedent of a rule are allowed to occur in the consequent, i.e. only individuals that are explicit in the KB can be used to support conclusions. The SWRL built-ins approach is based on the reuse of existing built-ins of XQuery [21] and XPath [22], which are themselves based on XML Schema Part 2: Datatypes[23]. XML and RDF sintaxis are provided for the serialization of SWRL on the web.

Each reasoning approach targets a very different kind of inference. The strength of DL lies in its very expressive class constructors that enable the definition of classes by describing the properties that their individuals hold. Given the axiomatization of the domain in such a class hierarchy, a DL reasoner can classify individuals according to the general schema and is able to determine if a given individual falls into some categories unanticipated by the modeler.

Nevertheless, reasoning over the combination of these kinds of rules defined on top of ontologies concepts is undecidable [4][9] and separating the reasoning in the structural component from the reasoning in the rule component of a hybrid KB is necessary in order to achieve decidable results [11][12]. In order not to be committed to a particular DL reasoner feature, we explored the Description Logic Interface (DIG), [16] which is a standard application program interface (API) for reasoning services on the web. DIG provides minimal functionality that is expected to grow until providing management of transactions, among others. The DIG specification consists of a XML Schema situated at http://dl-web.man.ac.uk/dig/2003/02/dig.xsd to encode DL language elements along with functionality that enables applications to ask and tell new statements to the Knowledge Bases that the reasoner is using. An application that uses DIG for reasoning services does not need to know which specific DL reasoner is providing those services. As part of our work, we have compared the main functionality provided by the NRQL query language implemented by the reasoner RACER [13] with the functionality provided by the interface DIG. It turned out that NRQL allows the management of OWL files while DIG does not. NRQL makes the use of Closed World Assumption possible, whereas this is impossible using DIG. NRQL also has a simple syntax that DIG lacks and provides functionality for deleting instances that DIG does not. Even though the NRQL query language is more expressive than the DIG interface, DIG provides a convenient standard style

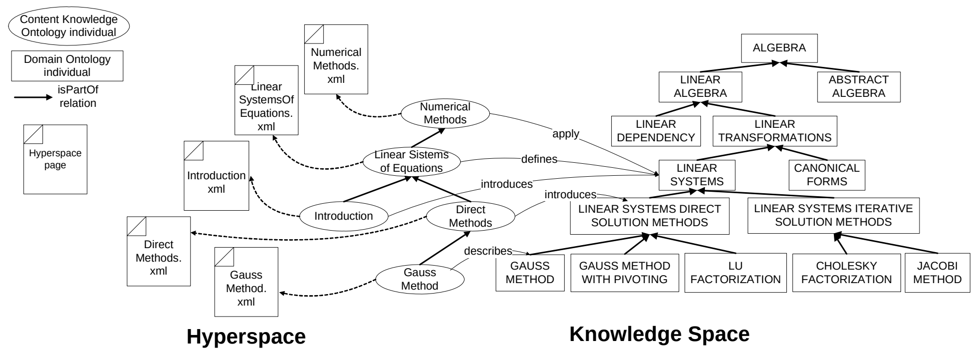

Fig. 2 Information Space

for a Semantic Web application and is sufficient for our practical purposes. In the end, we structured our knowledge base as a theory composed of a set of Description Logics axioms and Horn-like rules implemented using OWL DL and SWRL, respectively. Our KB takes advantage of both reasoning approaches, but the inference that each one contributes is carried out separately for efficiency purposes. The communication between the two parts of our KB is accomplished by using the API DIG [16].

4 The Hybrid Knowledge Base

As the Semantic Web vision requires, the terms that we have used to convey the meaning of our data are refinements of standard vocabularies that are already anchored on the Web. We have developed an *application profile* [17] of the Learning Object Metadata (LOM) standard [25] that conveys the meaning that our particular learning environment requires. We have used OWL DL, which is the most expressive OWL that is also decidable [2] for the application profile representation, and a RDF binding [24] as the implemented version of the conceptual standard. Standard terms, with a well known global semantics, are the basis for our application Profile, and all of our ontologies have their terminology based on them. Even RDF web agents that are not OWL aware can understand our terms according to the semantics of their roots on RDF terminologies. For the sake of clarity, and because it is not relevant for this paper, we have omitted the namespaces qualification accompanying the name of the elements of our KB, however, [17] has a detailed account of them. The DL segment of our KB is composed of the set of OWL statements that our ontologies contain. The rules segment of our KB resides on the SWRL sentences that the same ontologies files contain, they are Horn-like rules that assert conditions over individuals, concepts and relations defined in OWL axioms. We have addressed two different conceptual areas about which the system has to have knowledge about. One is the Knowledge Space of the system related to the educative content, and the other concerns the student profile to whom the educative content will be adapted [20].

4.1 Ontologies Structuring the Knowledge Space

The *Domain* and *Content Knowledge* ontologies form the Knowledge Space of the hybrid system. The Domain ontology contains a taxonomy of concepts that structures the domain being

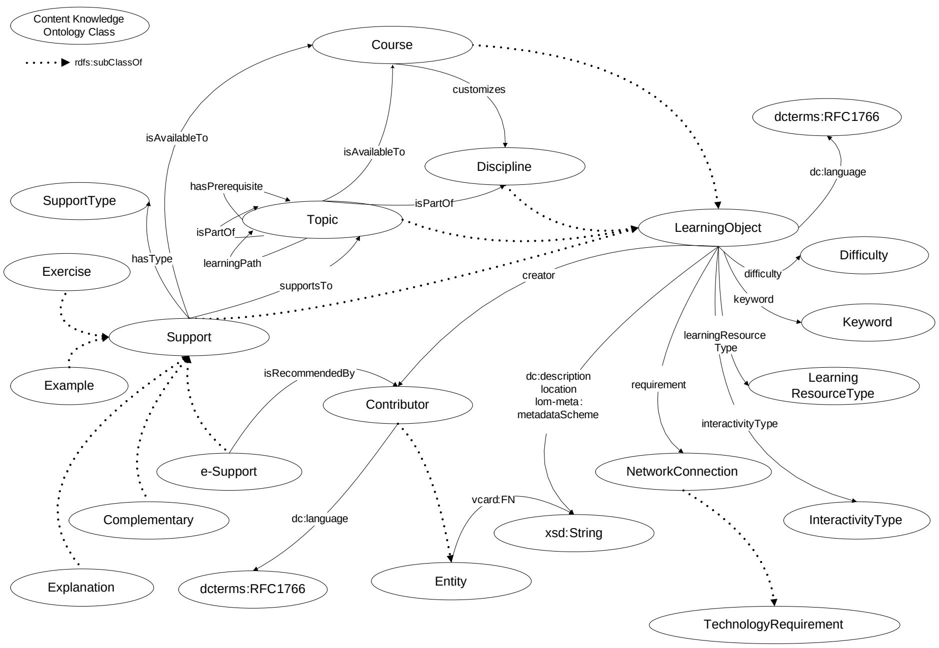

Fig. 3 Content Knowledge Ontology

taught. Most of its instances are related by mereological relations. The Content Knowledge ontology gives the necessary knowledge to correctly assemble the small pieces of educative content. It contains mereological relations, but also, among others, those that indicate a convenient order of presentation and prerequisite conditions. The knowledge contained in the Domain ontology is mostly used at the authoring or recommendation of educative content time. It enables a contributor to classify a piece of educative content as addressing the teaching of a certain topic of the domain taxonomy. On the other hand, the Content Knowledge ontology is intended to be used in the task of automatically computing complex learning objects according to the student's profile. Figure 2 shows the relations among a small set of individuals of the named ontologies. The *Hyperspace* is given by the XML pages containing the proper educative content.

Figure 3 partially shows the structure of classes of the Content Knowledge ontology. Instances of class *Topic* represent some concept or idea being part of a *Discipline* of study in the context of a given *Course*. The explanation of the topic comes with examples, exercises and complementary material represented in classes subsumed by class *Support*, such as *Example, Exercise* and *Complementary*. A topic may have sub-topics giving more specific and detailed explanations related by the *isPartOf* relation, i.e., they are considered part of the explanation of the prime topic. The *isPartOf* relation was declared transitive. The order in which the subtopics of a topic must be presented according to learning purposes is given by the *learningPath* relation, which was declared functional. Given a course, only certain topics of the discipline that the course customizes are available for it, and only certain support material of the available topics are in turn available for the course.

4.2 Ontologies Modeling the Student Profile

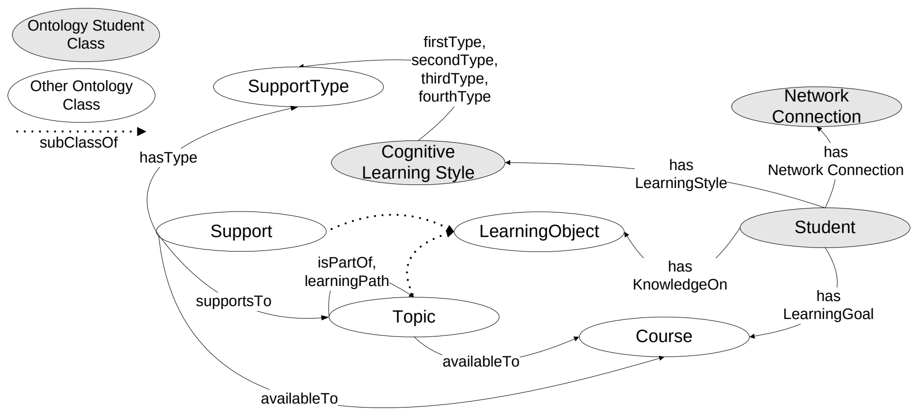

Fig. 4 Student Ontology

The Student ontology, depicted in figure 4, provides the necessary knowledge about the student profile used to address the task of computing "on the fly" those *Learning Objects* that pertain to the *Course* the student has as her learning goal, adapted to the capacity of her current *Network Connection*, and presented according to her *Cognitive Style of Learning*. The Cognitive Style of Learning (CLS) [26] is an individual aspect that describes the way in which a person habitually approaches or responds to the learning task. The entity `CognitiveLearningStyle` is intended to contain the four CLS identified in [27], i.e. (i) Analogue-Analytical; (ii) Concrete-Generic; (iii) Deductive-Evaluative and (iv) Relational-Synthetic. According to her CLS, a student may prefer to visit the examples about a topic, try the challenge of doing some exercises related to the topic, and then go on to the page containing the explanation of the topic, while another student with a different CLS may prefer to read the explanation of the topic first, then read the examples, and finally try the exercises. The order of precedence in which the different types of support material should be presented is given by relations `firstType`, `secondType, thirdType` and `fourType` shown in figure 4.

4.3 Rules for Reasoning on Instances and Properties

Rules have been used to tell our KB certain conditions that hold among the property values of related individuals. Description Logics has no tools to represent such a type of connection. For example, we use the following rule in order to tell our KB that it should infer that the language in which a learning object was written is the same as the language of its author, under the assumption that the most common situation is a creator authoring learning objects in her own native language.

$$learningObject(?x) \ \wedge \ creator(?x, ?y) \wedge language(?y, ?t) \rightarrow language(?x, ?t)$$

Considering the terminology defined by ontology *Content Knowledge*, figure 5 graphically shows through an example, that relations `isPartOf` and `learningPath` are expressive enough and human manageable to represent the semantics of the hierarchic structure of topics shown in figure 3, i.e., they are the clearest way to add or remove topics from a discipline structure,

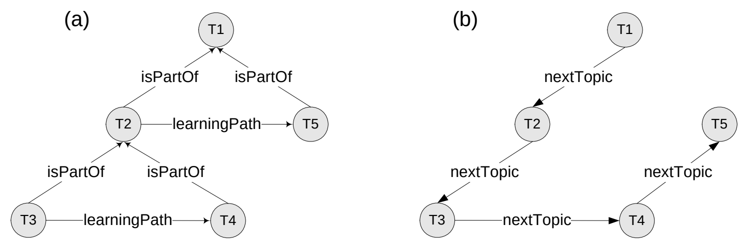

Fig. 5 Redundant properties for the presentation order of topics

they are also structurally well manageable for a DL reasoner. However, it is not possible for a DL reasoner to address the combination of instances of different relations. For example, in order to infer which the next topic of a given one is, based on the semantics of the two relations `isPartOf` and `learningPath`, there are three possible situations to be considered. Namely, topic `x` has topic `y` as its next topic on the presentation order, i.e., `nextTopic(x,y)` if (I) Topic `y` is the first child of topic `x` from left to right; (II) learningPath `(x,y)` holds and topic `x` has no children, or (III) isPartOf `(x,w)` holds and learningPath `(w,y)` holds, topic `x` has no children nor sibling at its right. Expressed in FOL:

$$(I) \qquad (\forall x,y,z,t)\ isPartOf(y,x)\ \wedge\ \neg learningPath(z,y)\ \wedge\ \neg(isPartOf(y,t)\ \wedge\ isPartOf(t,x))\ \rightarrow\ nextTopic(x,y)$$

$$(II) \qquad (\forall x,y,z)\ learningPath(x,y)\ \wedge\ \neg isPartOf(z,x)\ \rightarrow\ nextTopic(x,y)$$

$$(III) \qquad (\forall x,y,z,t,w)\ isPartOf(x,w)\ \wedge\ learningPath(w,y)\ \wedge\ \neg isPartOf(z,x)\ \wedge\ \neg learningPath(x,t)\ \rightarrow\ nextTopic(x,y)$$

Due to the fact that property `isPartOf` was declared transitive, each topic is considered to be indistinctly part of all the topics included on the transitive clousure of relation `isPartOf`, i.e. topics `T1, T2` and `T3` are considered as holding `isPartOf(T3,T2)`, `isPartOf(T2,T1)`, but also holding `isPartOf(T3,T1)`. However, quering the KB did not allow us to differentiate between instances `isPartOf(y,x)` that were explicitly told to our KB and those that result from the transitive clousure of the relation `isPartOf` .

To deal with this issue, we have defined a property that collapses the two properties `isPartOf` and `learningPath` into only one, called `nextTopic`, whose individuals state what topic should be after a given one regardless of structure issues. The criteria used to define `nextTopic` in function of `isPartOf` and `learningPath` are those shown in the given conditions (I), (II) and (III). Instances of `nextTopic` relation are shown in part (b) of figure 5. In the end, what we considered the best way to approach the issue was to maintain the two expressive structural relations as the main way to express the structural terminology, but to solve the computation of the next topic when automatically configuring a new learning object, by using the new redundant relation. In order to maintain consistency, the instances of each alternative representation are computed in a controlled automatic basis. Following are the SWRL rules whose computation populates the extension of relation `nextTopic`, we have used the informal human readable notation given in [9].

$$(I) \qquad Topic(?x) \ \wedge \ Topic(?y) \ \wedge \ isPartOf(?y,?x) \ \wedge$$
$$\neg(Topic(?z) \ \wedge \ isPartOf(?z,?x) \ \wedge \ learningPath(?z,?y)) \ \wedge$$
$$\neg(Topic(?t) \ \wedge \ isPartOf(?t,?x) \ \wedge \ isPartOf(?y,?t)) \ \rightarrow \ nextTopic(?x,?y)$$

$$(II) \qquad Topic(?x) \ \wedge \ Topic(?y) \ \wedge \ learningPath(?y,?x) \ \wedge$$
$$\neg(Topic(?z) \ \wedge \ isPartOf(?z,?x)) \ \rightarrow \ nextTopic(?x,?y)$$

$$(III) \qquad Topic(?x) \ \wedge \ Topic(?y) \ \wedge \ (Topic(?z) \ \wedge \ learningPath(?z,?y) \ \wedge$$
$$isPartOf(?x,?z) \ \wedge \ \neg(Topic(?u) \ \wedge \ isPartOf(?u,?x)) \ \wedge$$
$$\neg(Topic(?t) \ \wedge \ learningPath(?x,?t)) \ \wedge$$
$$\neg(Topic(?v) \ \wedge \ Topic(?w) \ \wedge \ isPartOf(?x,?v) \ \wedge$$
$$isPartOf(?v,?z) \ \wedge \ learningPath(?v,?w)) \ \rightarrow \ nextTopic(?x,?y)$$

We have used the built-in `swrlb:booleanNot` provided in [9] to be able to represent negation in SWRL. Each time that the structure of topics is modified, the extension of relation `nextTopic` will automatically be computed by the inference of the rule engine. For the sake of clarity we have omitted the considerations related to prerequisite conditions that are actually used for the computation of the next topic, but whose use is similar to what was presented here. Another important issue that required the use of rules to be solved concerns the necessary inference for the automatic computation of the best order for a set of supporting material given the Cognitive Learning Style of the student. Part (b) of Figure 6 shows instances of relation `nextSupport` obtained similarly to what was presented to obtain the extension of `nextTopic`.

To depict the entire process we will start by Part (a) of Figure 6 that shows, as an example, part of the structure of `course1`. Two topics are available for the course. Topic `topic1` has an explanation, two examples, two exercises and one piece of complementary material. We assume that all the showed support individuals are available for `course1`. Each individual of class `Support` is related to an individual of class `SupportType` indicating the four types of support the system has. The individual `relational-Synthetic` of class

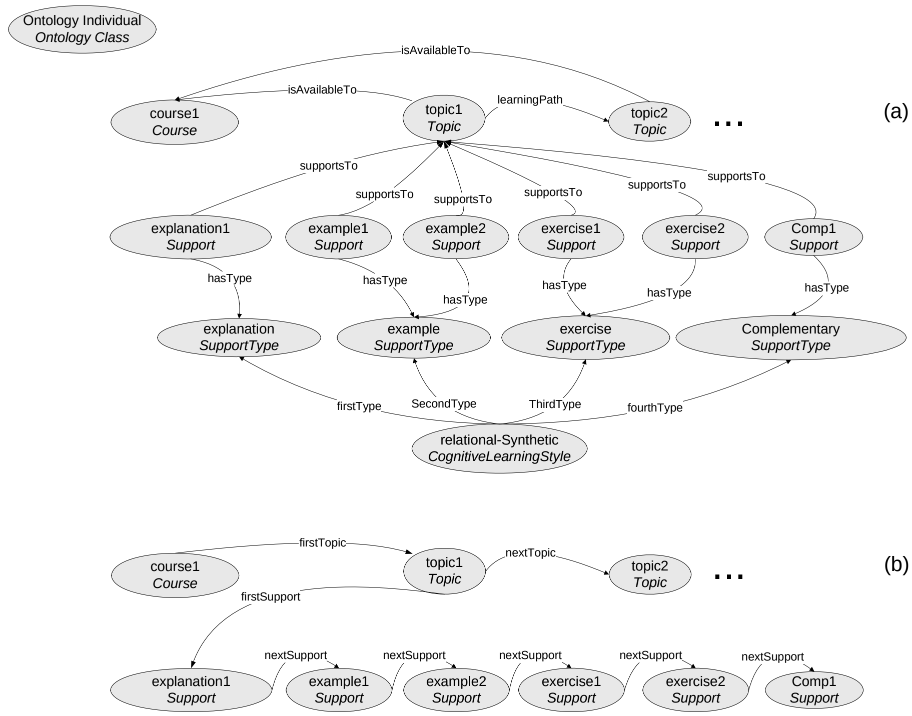

Fig. 6 Cognitive Learning Style determining the presentation order for the support material of `topic 1`

`CognitiveLearningStyle` is representing such a particular CLS. The order in which the support material should be presented to students that have `relational-Synthetic` CLS is given by the four relations named `firsType, secondType, thirdType` and `fourthType` whose range are in class `SupportType`. For example, `firsType(relational-Synthetic, explanation)` indicates that individuals with that CLS prefer to receive explanations first.

The actual order among elements of the same category is given by their value on property `difficulty`, which stands for the degree of difficulty that is expected the learning object will present when a student approaches it. We have defined rules for computing the extension of relation `nextSupport`, shown in Part (b) of figure 6, in an analogous way to what we did to comput the extension of `nextTopic` relation. We have also used rules to let the system know when a student has completed the study of a given topic. The following SWRL rule creates a new instance of relation `hasKnowledgeOn` relating the student to the topic once the student has knowledge about all the supporting material of the topic.

$$\neg(Student(?z)\ \wedge\ Topic(?x)\ \wedge\ Support(?y)\ \wedge\ supportsTo(?y,?x)\ \wedge$$

$$\neg hasKnowledgeOn(?z,?y))\ \rightarrow\ hasKnowledgeOn(?z,?x)$$

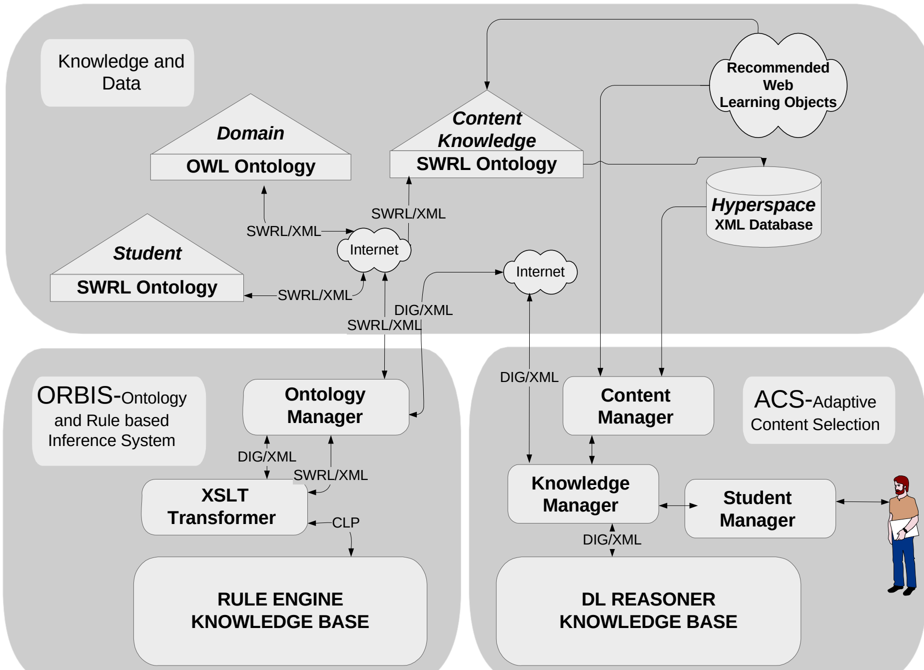

Fig. 7 System Architecture

5 Hybrid Inference for Adaptation

Figure 7 shows the architecture used to support hybrid inference for adaptation purposes. The **ACS- Adaptive Content Selection** module groups the system functionality related to adaptation. It contains three sub-modules, namely, the Student Manager, the Content Manager, and the Knowledge Manager. The **Student Manager** module continuously monitors the activities of the student on the system in order to update the Student Model. The **Content Manager** module makes the educative content required by the other modules available. The **Knowledge Manager** module uses the DL reasoner to query the Knowledge provided by both the Student Model and the Knowledge Space of the system. It acts as a broker among the other **ACS** sub modules given the foundations to decide both which educative content to provide to each student and what actualizations to the Student Model should be done. The Knowledge Manager module uses the standard for the Web **DIG/XML- Description Logics Interface** to communicate knowledge with other modules and in particular with the reasoner.

The **ORBIS- Ontology and Rule-based Inference System** module provides both functionality related to the ontologies storage and **rule based reasoning services**. It encapsulates the use of the **JESS rule engine** by means of a transformer based on **XSLT** technology. The transformer acts as a wrapper converting the standard for the Web **SWRL- Semantic Web Rule Language** rules that the Knowledge Manager has to tell to the rule engine KB, into the internal format CLP of JESS, and also transform the new ontology individuals inferred by the rule engine, into DIG/XML sentences that will be told to the KB of the DL reasoner. We have used the DL reasoner RACER[13] and the rule engine JESS[15]. After each cycle of the rule engine reasoning, the inferred instances are added to the DL KB, so that they are used for structural querying. In spite of the fact that OWL is based on the *open world assumption*, and the terminological part of our KB is treated under such an assumption, we take advantage of *closed world assumption*, in particular of *negation as failure* by encapsulating inference inside the rule engine.

Various students are allowed to work simultaneously on the system. We had to deal with some restrictions imposed by the use of the standard DL interface DIG, in particular, the interface does not allow for transactional management and for identifying different clients that are working at the same time in a certain KB. We decided to use one KB per student working on the system, since working with various KB at the same time is possible for the reasoner used. KB elements that need to be modified during the student activity are instances of the Student ontology that only concern the student doing the learning activity. The instances of knowledge about the educative content, that are commonly accessed for all the students, are not modified during the activity of taking courses.

Following is an example of the set of rules and DL queries executed over a temporal KB for the configuration of the best piece of educative content to be offered dynamically to a given student, say `student1`, who is actively working on the system. This configuration will be made according to her current knowledge of the material of the course she is taking, say `course1`, and her Cognitive Learning Style. Figure 8 shows most of the inferred concepts. The identifiers of the support material were not included for simplicity. Learning objects that the student already knows are shown in gray, and the rest of the course material in white.

This example is considered representative of the whole set of queries used. Other kinds of adaptation, like the one related to the type of network connection, are ommited for the sake of clarity. We have shown in section 4.3 how SWRL rules were used to set the extension of properties `nextTopic` and `nextSupport`. While the extension of `nextTopic` only concerns the structure of the educative content, the extension of `nextSupport` was inferred according to the Cognitive Learning Style of the student (see figure 6), i.e. the extension of property `nextSupport` has already been tailored to the preferences of the student .We have used the DL notation used in [3] for the purpose of clarity and the human readable syntax for SWRL rules given in [9]. The notation R^- stands for the inverse relation of relation R, i.e. $(\forall p, q), R(p, q) \rightarrow R^-(q, p)$. To begin with, a query is posed to compute the extension of concept `Known`, containing those learning objects available for `course1` that `student1` has already learnt. Instances of this concept are depicted in gray in figure 8.

$$Known \equiv \exists hasKnowledgeOn^-.student1 \sqcap \exists isAvailableTo.\{course1\}$$

Concept `TopicC1`, whose instances are shown on the top of figure 8, contains those topics that are available for `course1`, its extension is computed by the following query:

$$TopicC1 \equiv Topic \sqcap \exists isAvailableTo.\{course1\}$$

Concept `UnKnownTopic` contains those topics that are available for `course1` but that `student1` has not learnt yet, its instances are the topics shown in white on top of figure 8, and its extension is computed by the following SWRL rule:

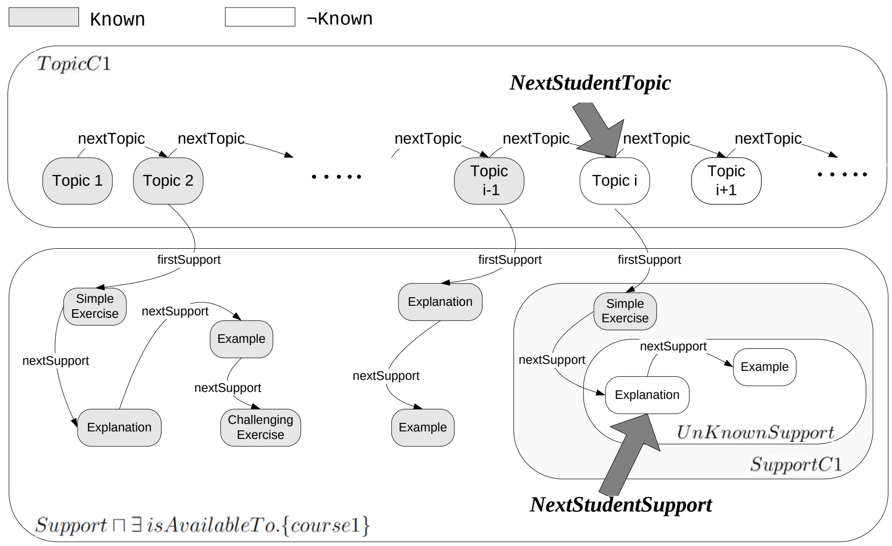

Fig. 8 Configuration of a Personalized Learning Object

$$TopicC1(?x) \wedge \neg Known(?x) \rightarrow UnKnownTopic(?x)$$

Concept `FirsTopic` contains the first topic of course `course1`, given by the following SWRL rule:

$$TopicC1(?x) \wedge \neg(TopicC1(?y) \wedge nextTopic(?y, ?x)) \rightarrow FirstTopic(?x)$$

`NextStudentTopic`, indicated in figure 8, describes the next topic to be presented to the student `student1`. It should be either the first topic of the course, if it is unknown, or an unknown topic preceded by a known one. Following is the DL query that sets its extension in the temporal KB:

$$NextStudentTopic \equiv FirstTopic \sqcap UnKnownTopic \sqcup$$
$$UnKnownTopic \sqcap \exists\, nextTopic^{-}.Known$$

`SupportC1`, indicated by a shaded box in figure 8, contains the support material available for `course1` that supports the topic of `NextStudentTopic`, given by the DL query:

$$SupportC1 \equiv\ Support\ \sqcap\ \exists\ isAvailableTo.\{course1\} \sqcap$$
$$\exists\ supportsTo.NextStudentTopic$$

`UnKnownSupport`, indicated by a white box in figure 8, contains the unknown support material for the next topic, its extension is computed by the following SWRL rule:

$$SupportC1(?x) \wedge \neg Known(?x) \rightarrow UnKnownSupport(?x)$$

`FirstSupport` is the first support according to relation `nextSupport` that is unknown to the student, its extension is computed by the following SWRL rule:

$$SupportC1(?x) \ \wedge \ \neg(SupportC1(?y) \ \wedge \ nextSupport(?y, ?x)) \rightarrow FirstSupport(?x)$$

Finally, `NextStudentSupport`, indicated in figure 8, contains the next learning object to be presented to the student, given by the DL query:

$$NextStudentSupport \equiv (FirstSupport \ \sqcap \ UnKnownSupport) \sqcup$$
$$(UnKnownSupport \ \sqcap \ \exists \ nextSupport^{-}.Known))$$

Once the concept `NextStudentSupport` is populated, the system can provide the student with the next educative content in its personalized learning. Following is the DIG serialization used for telling the temporal KB this last DL axiom stating the definition of `NextStudentSupport`:

```
<defconcept name="NextStudentSupport"/>
<equalc>
  <catom name="NextStudentSupport"/>
  <or>
    <and>
      <catom name="FirstSupport"/>
      <catom name="UnKnownSupport"/>
    </and>
    <and>
      <catom name="UnKnownSupport"/>
      <some>
     <inverse>
         <ratom name="http://www.inf.ufrgs.br/~tapejara/Ontology/
Generated/AWOntology.owl#nextSupport"/>
    </inverse>
       <catom name="Known"/>
      </some>
    </and>
  </or>
</equalc>
```

In spite of the fact that our work was carried out using the reasoner RACER [13], our system is able to obtain inference services from any up-to-date reasoner that implements the standard API DIG. As it can be seen, our approach involves the declarative representation of not only the knowledge concerning the application domain, but also the knowledge related to all the needed steps to solve the addressed problem.

6 Conclusions

We have developed an Intelligent Adaptive Hypermedia System that entirelly works according to the foundations of the SemanticWeb. Its knowledge is encoded in the standard for the Web languages SWRL, OWL, RDF, XML and DIG. DL based languages enabled us to declaratively encode the knowledge about our domain, taking automatic account of inheritance and performing structural inference, while Horn-like rules defined on top of our ontology terms enabled us data-directed inference over instances. Our system takes advantage of an open world assumption as the default assumption for sharing our knowledge on the web, while closed world assumption is encapsulated in rule inference when needed. We are able to share the knowledge behind our inference on the Web by exposing our OWL axioms and our SWRL rules in a machine understandable representation. We avoided encapsulation of knowledge in ad-hoc programming code. All our knowledge is expressed in a declarative, shareable way, in languages that have well founded semantics and is used for inference in reasoners that work in a principled way. Our system is able to automatically configure complex learning objects tailored to student profiles by reasoning in a domain theory.

References

1. P. Brusilovsky, *Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction*, 6 (2-3), 87-129, 1996.
2. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider and L. Stein, *OWL Web Ontology Language Reference*. W3C Recommendation, 2004.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider editors, *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2002.
4. I. Horrocks, P. F. Patel-Schneider, S. Bechhofer, and D. Tsarkov, *Owl rules: A proposal and prototype implementation*. Conditionally accepted for publication. Journal of Web Semantics, 2005.
5. R. Brachman, V. Gilbert and H. Levesque, *An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of KRYPTON*. IJCAI 1985: 532-539.
6. B. Nebel, and K. vonLuck, *Hybrid Reasoning in BACK. In Methodologies for Intelligent Systems*, Volume 3, eds. Z. W. Ras and L. Saitta, 260-269. New York: North-Holland, 1988.
7. A. Borgida, R. Brachman, D. McGuinness, and L. Resnick, *CLASSIC: A structural data model for objects*. In Proc. Of the ACM SIGMOD Int. Conf. on Management of Data, pages 59-67, 1989.
8. A. Levy and M. Rousset, *Combining Horn rules and description logics in CARIN. Artificial Intelligence*, 104(1-2):165-209, 1998.
9. I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabets, B. Grosof, M. Dean, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML* W3C Member Submission 21 May 2004.
10. http://owl.man.ac.uk/hoolet/
11. G. Antoniou, C. Damasio, B. Grosof, I. Horrocks, M. Kifer, J. Maluszynski, and P. Patel-Schneider, *Combining rules and ontologies*. A survey. http://rewerse.net/deliverables/m12/i3-d3.pdf, 2005. REWERSE Deliverables, No. I3-D3, February 2005.
12. R. Rosati. *On the decidability and complexity of integrating ontologies and rules*. Journal of Web Semantics, 3, 2005.
13. V. Haarslev, R. Moller, RACER *User's Guide and Reference Manual*. Version 1.7.7. 2003.
14. I. Horrocks, The FaCT system. In Harrie de Swart, editor, Proc. Of the 2nd Int. Conf. on *Analytic Tableaux and Related Methods (TABLEAUX'98)*, volume 1397 of Lecture Notes in Artificial Intelligence, pages 307-312. Springer, 1998.
15. Jess Rule Engine: http://herzberg.ca.sandia.gov/jess/
16. S. Bechhofer, R. Moller, and P. Crowther, *The DIG Description Logic Interface*. In Proc. of International Workshop on Description Logics (DL2003), Rome, Italy, 2003.
17. L. Silva-Muñoz, J. Palazzo, *Adaptive Web-Based Courseware Development Using Metadata Standards and Ontologies*. The 16th International Conf. on Advanced Information Systems Engineering - CAISE , volume 3084 of Lecture Notes in Computer Science, pages 414-428. Springer, 2004.

18. E. Duval, W. Hodgins, S. Sutton, S. Weibel, *Metadata Principles and Practicalities*. D-Lib Magazine 8(4), 2002.
19. R. Heery, M. Patel, *Application profiles: mixing and matching metadata schemas*. September 2000, ARIADNE Issue 25.
20. P. Brusilovsky, *Developing Adaptive Educational Hypermedia Systems: ¿From Design Models to Authoring Tools*. in T. Murray, S. Blessing and S. Ainsworth (Eds.), In Press.
21. XQuery: http://www.w3.org/TR/xquery/
22. XPath: http://www.w3.org/TR/xpath/
23. P. V. Biron and A. Malhotra. XML schema part 2: Datatypes. W3C Recommendation,May 2001. http://www.w3.org/TR/xmlschema-2/.
24. M. Nilsson, IEEE Learning Object Metadata RDF binding. May 2001. http://kmr.nada.kth.se/el/ims/metadata.html.
25. Learning Technology Standards Committee of the IEEE. Draft Standard for Learning Object Metadata (LOM) IEEE 1484.12.1-2002.
26. R. Riding, S. Rayner, *Cognitive Style and Learning Strategies*. London: David Fulton Publishers, 2000.
27. M. Souto, M. Nicolao, R. Viccari, J. Palazzo, R. Verdin, K. Beschoren, M. Madeira, R. Zanella, *Web Adaptive Training System Based on Cognitive Student Style*. IFIP World Computer Congress, 17th Edition, August 25-30, 2002, Montreal, Canada.
28. P. Patel-Schneider I. and Horrocks, 2006. Position paper: *a comparison of two modelling paradigms in the Semantic Web*. In Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 3-12. DOI= http://doi.acm.org/10.1145/1135777.1135784.
29. L. Silva-Muñoz, J. Palazzo, *Applying Semantic Web Technologies to Improve Personalization and Achieve Interoperability between Educational Adaptive Hypermedia Systems*. International Workshop on Applications of Semantic Web Technologies for Adaptive Educational Hypermedia. - SW-EL'04, in conjunction with the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, August 23-26, 2004, Eindhoven, The Netherlands.
30. Dublin Core Metadata Element Set, Version 1.1: Reference Description. Available at http://purl.org/dc/documents/rec-dces-19990702.htm