Searching for the knowledge in the Semantic Web

Nenad Stojanovic

AIFB Institute
University of Karlsruhe
D-76128 Karlsruhe
Germany
nst@aifb.uni-karlsruhe.de

Ljiljana Stojanovic

Research Center for Information Technologies
at the University of Karlsruhe
Haid-und-Neu-Str. 10-14
76131 Karlsruhe,
Germany
Ljiljana.Stojanovic@fzi.de

Abstract

The machine understandable semantic of information, achieved by using an RDF(S) structure and common-shared vocabularies (ontologies) is the big step in enabling the machine-agent interoperability on the Web. Machine agents can crawl annotated web pages, search for useful information from various sources, use the information to solve tasks at hand by using the internal reasoning mechanism and background knowledge. In order to enhance their inference capabilities, machine- (and also human-) agents need to update their knowledge, using relevant knowledge sources as much as possible. One of the possible scenarios is to search for relevant knowledge on the (Semantic) Web.

In this paper we discuss the prerequisites for design, and present an approach for representing rules in the machine understandable form, which is based on the current efforts in achieving the machine understandable semantic of information. Such representation of rules can serve as the backbone for a web-enabled knowledge management process. In the presented usage scenario we focus on the knowledge sharing phase in that process, i.e. on the searching for relevant knowledge (rules) on the Web.

Introduction

Current Web is the great success in terms of amount of information and number of human users. It starts to influence all aspects of our daily life and business. But the questions like: are the users satisfied with the automation of the services on the Web, or can the right information in the Web repository be found in the real-time for a non-expert user, lead to discussion about too human-oriented Web development. The information coded in HTML is machine readable, which means that the information can be presented to a user, but the meaning of the information is understandable only for human beings. In order to enable communication between machine agents, Web should support a machine-understandable semantic of information. That is the vision of the second generation of the Web, so called Semantic Web (Berners-Lee, Handler and Lassila 2001). Current W3C proposals enable the exchange of information in the form of object-attribute-value, using predefined vocabulary, which expresses well-defined meaning of used terms and relations between them (ontologies). So far, the language for expressing such ontological statements on the Web (DAML 2001), (Fensel et al. 2001) has been proposed in order to support the development of the tools, which will enable communication between machine agents on the semantic basis.

In order to behave intelligently, those agents need a reasoning mechanism and background knowledge about domain, which are used for processing information (facts) exchanged between agents. But it is too simple to assume that only such factual information is exchanged between agents. Agents operate in the highly changeable environment, and they need a mechanism to update their background knowledge and to adapt reasoning strategies to new requirements automatically (by interacting with the environment, contacting other agents etc). The sources of "new" knowledge could be the knowledge from knowledge repositories (rule bases) published on the web, the knowledge stored in other machine agents, as well as on the web published by individual experts (e.g. knowledge annotated on personal web pages). Therefore, the manipulation with the machine-understandable semantic of knowledge should be the next challenge for the Semantic Web research. It will enable not only "upgrading" the background knowledge of machine agents, but also support decision-making process of human agents.

In this paper, we present a format for exchanging knowledge on the Semantic Web, based on the current W3C proposals for sharing information (W3C 2000), (W3C 1999), (Brickley and Guha 2000). Since the knowledge can be packed in various representation formats (rules, decision trees, cases), we constrain our approach to the format for representing knowledge that is mostly used in commercial applications – rules. However, the whole usage scenario could be reused for other knowledge representation formalisms.

The paper is organised as follows: in the second section we analyse the current W3C efforts for representing metadata, and propose an RDF format for encoding a rule language; in the third section we discuss the characteristics of the proposed format; in the fourth, we present an application scenario. We conclude with some remarks about our future work.

Data/Information/Knowledge on the Web

In this section we briefly discuss current W3C metadata representation efforts (W3C 2000), (W3C 1999), (Brickley and Guha 2000) from the point of view of the knowledge transformation cycle (Sowa 2000), in order to define requirements for a machine-understandable format for representing knowledge on the Web.

XML is the language for transmitting semi-structured information, and the structure of an XML file can be validated according to a corresponding DTD or XMLSchema (Erdmann and Studer 2000). The basic model is directed, labelled graph (DLG). However, when information is represented as a DLG, the nodes don't actually contain any information: all content is in the connections (Berners-Lee 2001b). In order to define semantic of a connection explicitly, one has to define the domain and range of that connection explicitly. Although XML provides means for range specification, there is no possibility to define a domain of the element (connection) explicitly. The domain is defined implicitly as the element in which the definition appears. Therefore, this simple data-value (object-value) semantic in XML representation mechanism can be used only for representing data (facts), but not for defining information explicitly.

RDF is based on the simple Object-Attribute-Value knowledge representation mechanism, which is the basic structure in semantic networks and frame-based systems. O-A-V is a factual information, for example Jim-work_at-ProjectY. In RDF, the domain and range of a connection (property/attribute) can be defined explicitly, with the constraint that multiple range statements are not allowed. Such statements can be "simulated" – by having a class as the range of a property and by allowing the use of all its sub-classes as values of the property.

RDF, in combination with RDF Schema, offers modelling primitives that can be extended according to the needs at hand. Basic class hierarchies and relations between classes and objects are expressible in RDF Schema. In general, RDF(S) suffers from a lack of formal semantics for its modelling primitives, making the interpretation of how to use them properly an error-prone process.

The most reused definition of knowledge in the knowledge management literature is that knowledge is information in the context of other information. Using the reification mechanism, it is possible to make a statement about a statement in RDF, and in that way to form the context of information. However, in order to define a valid, uniform context of information, some extension of RDF Schema should be proposed (e.g. the means for expressing implication).

Since the semantic of RDF(S) is not powerful enough to express knowledge (rules) in RDF, we decide to encode the datastructure of a rule language in RDF. RDF Schema extension can serve as a repository for terms used in defining the structure of knowledge (e.g. rule body, rule head). The content of knowledge is expressed by using the vocabulary from domain ontology (e.g. person, project).

We constrain our knowledge representation to a set of Horn clauses, and we here discuss briefly some of our design rationales for rule language described in Figure 1 (in the form of RDF Schema)

-----------------RDF Schema for Rule Language -------------

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
<!ENTITY rule 'http://ontoserver.aifb.uni-karlsruhe.de/rule#'>]>

  <rdf:RDF xmlns:rdf="&rdf;"
   xmlns:rdfs="&rdfs;"
   xmlns:rule="&rule;"
   xmlns="&rule;" >

  <rdfs:Class rdf:ID="Rule"/>

  <rdf:Property rdf:about="Body">
   <rdfs:domain rdf:resource="&rule;Rule"/>
   <rdfs:range rdf:resource="&rule;Formula"/>
  </rdf:Property>

  <rdf:Property rdf:about="Head">
   <rdfs:domain rdf:resource="&rule;Rule"/>
   <rdfs:range rdf:resource="&rule;Formula"/>
  </rdf:Property>

  <rdfs:Class rdf:ID="Formula"/>
  <rdfs:Class rdf:ID="NilFormula">
     <rdfs:subClassOf
     rdf:resource="&rule;Formula"/>
  </rdfs:Class>

  <rdf:Property rdf:about="HeadFormula">
   <rdfs:domain rdf:resource="&rule;Formula"/>
   <rdfs:range rdf:resource="&rule;Fact"/>
  </rdf:Property>

  <rdf:Property rdf:about="TailFormula">
   <rdfs:domain rdf:resource="&rule;Formula"/>
   <rdfs:range rdf:resource="&rule;Formula"/>
  </rdf:Property>

  <rdfs:Class rdf:ID="Fact"/>
  <rdfs:Class rdf:ID="NilFact">
   <rdfs:subClassOf rdf:resource="&rule;Fact"/>
  </rdfs:Class>
```

```
<rdf:Property rdf:about="Not">
 <rdfs:domain rdf:resource="&rule;Fact"/>
 <rdfs:range rdf:resource="&rule;NilFact"/>
</rdf:Property>

<rdf:Property rdf:about="Pred">
 <rdfs:domain rdf:resource="&rule;Fact"/>
   <rdfs:range rdf:resource="&rule;Predicate"/>
</rdf:Property>
<rdf:Property rdf:about="Subj">
 <rdfs:domain rdf:resource="&rule;Fact"/>
 <rdfs:range rdf:resource="&rule;Subject"/>
</rdf:Property>
<rdf:Property rdf:about="Args">
 <rdfs:domain rdf:resource="&rule;Fact"/>
 <rdfs:range rdf:resource="&rule;Cons"/>
</rdf:Property>

<rdfs:Class rdf:ID="Predicate">
 <rdfs:subClassOf rdf:resource="&rdfs;Class"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Subject">
 <rdfs:subClassOf rdf:resource="&rdfs;Class"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Cons"/>
<rdfs:Class rdf:ID="NilCons">
 <rdfs:subClassOf rdf:resource="&rule;Cons"/>
</rdfs:Class>

<rdf:Property rdf:about="HeadCons">
 <rdfs:domain rdf:resource="&rule;HeadCons"/>
 <rdfs:range rdf:resource="&rule;Term"/>
</rdf:Property>

<rdf:Property rdf:about="TailCons">
 <rdfs:domain rdf:resource="&rule;TailCons"/>
 <rdfs:range rdf:resource="&rule;Cons"/>
</rdf:Property>

<rdfs:Class rdf:ID="Term"/>
<rdfs:Class rdf:ID="Variable">
 <rdfs:subClassOf rdf:resource="&rule;Term"/>
</rdfs:Class>
<rdfs:Class rdf:about="&rdfs;Literal">
 <rdfs:subClassOf rdf:resource="&rule;Term"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Resource">
 <rdfs:subClassOf rdf:resource="&rule;Term"/>
</rdfs:Class>

<rdf:Property rdf:about="Var">
 <rdfs:domain rdf:resource="&rule;Variable"/>
 <rdfs:range rdf:resource="&rule;Literal"/>
</rdf:Property>

<rdf:Class rdf:ID="Model">
 <rdfs:subClassOf rdf:resource/>
</rdf:Class>
<rdf:Property rdf:ID="Context">
 <rdfs:domain rdf:resource="&rule:Rule"/>
 <rdfs:range rdf:resource="&rule:Model"/>
</rdf:Property>

</rdf:RDF>
```

Figure 1: RDF Schema for Rule Language

Encoding Rule Language in RDF

A rule contains a number of antecedent facts and a number of consequent facts that can be inferred to be true when all of the antecedent facts are determined to be true. We consider derivation rules whose action happens to only add or "assert" a conclusion when certain conditions (premises) are fulfilled. A rule format contains variables and should cope with negated statements. The format contains also additional properties related to context information, but also properties related to uncertainty, and inference strategies can be added easily.

The design rationales for the proposed RDF rule language are: (i) format, which can enable efficient searching for arbitrary rule patterns and (ii) format, which is compatible to ongoing rule markup initiatives (Boley, Tabet and Wagner 2001), (Sintek and Decker 2001). Moreover, we tried to reuse as much as possible of the native RDF model to encode complex characteristics of our rule language. In the following, we discuss some of those characteristics.

Naming of values: The problem often encountered when looking at converting information to an RDF-based format is that the original does not use URIs for identifying values. In order to take advantage of RDF, the naming format used should be based upon or mapped to a URI or equivalent form (Berners-Lee 2001a).

Facts of arbitrary arity: RDF can present only monadic or dyadic predicates directly. In order to represent n-ary predicates consistently with dyadic ones, we decided not to use rdf:Seq container type. We adopt the rule format so that n-ary predicate has the form of a normal RDF statement, whose subject is the first argument of the predicate on which the fact is based, whose property is the predicate and whose object is a list of any remaining arguments. It introduces a number of special resource identifiers, associated here with the namespace prefix **rule**: (it points to URI <'http://ontoserver.aifb.uni-karlsruhe.de/rule#'>)

An example of representing the list of arguments in a n-ary predicate is presented below.

rule:Cons is an auxiliary data type (i.e. an rdfs:Class resource) that describes a list constructor. It constructs a list from a rule:Pred element and rule:Arg sublist.

rule:HeadCons is a property that designates the head of a list represented by rule:Term

rule:TailCons is a property that designates the tail (i.e. all of the list elements other than head) of a list represented by a rule:Cons value.

rule:NilCons is a distinguished value with type rule:Cons that represents an empty list. A list is terminated by this value as the rule:TailCons property of a rule:Cons value.

Representation of rules: The basic structure of a rule is a set of antecedent facts and a set of consequent facts. A rule also provides the scope for any variables that appear in the facts (see below). It is clear that representation of rule:Head and rule:Body is based on the same list-recursive mechanism as in the case of facts of arbitrary arity (rule:HeadCons and rule:TailCons).

Representation of variables: In order to have a useful expressive power, a rule language has to incorporate a concept of variable. A variable is an identifier that represents a value, but which may stand for different values when used in different contexts. When a variable appears in a fact that is part of a rule, the scope of the variable is the containing rule. That means that all occurrences of a given variable which appears within the rule are required to bind to the same value in any single invocation of the rule. Variables are represented by using class rule:Variable.

Negation Handling: This is a controversial issue in the semantic web context: non-monotonic or monotonic web, negation by failure or explicit negation, a *weak* negation expressing *non-truth* (in the sense of "I don't like snow") and a *strong* negation expressing explicit *falsity* (in the sense of "I dislike snow") (Boley, Tabet and Wagner 2001). The strong negation is an "open world" negation, since in an open world such as the Web, the non-truth (or failure) of a statement does not imply its falsity (the inference from "nothing in this knowledge base matches A" to "A is false" is the non-monotonic step that should be avoided, or at any rate somehow controlled and isolated, in order to make the semantic web useable). But there is also the opinion that the scope of rules need not be the entire Web, but it may be restricted to a subset (domain), or particular DB, where then negation-as-failure is as viable for such Semantic Web application as it is in a relational (or deductive) database application. We define only weak negation using property rule:NOT .

Model: In many applications it is important to distinguish between different kinds of RDF data, i.e. different sources of RDF data. The most simple example is that one source is trustworthy and the other one not. This requirement need be presented in a rule language. One of the solutions is to express the difference between different sources by model identifiers, as in our approach. We represent models explicitly - they are “first class citizens”. The first step is to be able to distinguish between different sources - that mechanism is the model. Extended properties, for example the notation of belief or trust, could be attached to the model, representing the level of trust assigned to that data source.

Rule attributes: If some extra information has to be added to a rule, then it is easy to attach that information in the form of a new RDF property for a rule. The mostly used properties in the business rule community are: certainty factor, for handling uncertainty and priority, for handling conflict-resolution.

Comment: Our model has one main difference in comparison to the most prominent and accepted RuleML initiative: we don`t use RDF containers to express n-ary structures, while this structure is not properly tied into RDF model (Conen and Klapsing 2000) and the logical interpretation of containers gave rise to a number of discussions.

Application Scenario – Searching for Expert Rules

Representing rules in the common-agreed format which could be machine-processed on the semantic basis enables the Web-support for all phases in the knowledge management cycle: creating/importing knowledge, knowledge capturing, searching for knowledge, knowledge use (Staab et al. 2001). In this section we give a part of the whole scenario, related to searching for knowledge on the Web.

As the recent results in knowledge management research teach us, the intelligent assistance in knowledge-intensive problem solving has more value for the user than automated reasoning (e.g. using an expert system) (Abecker et al. 1998). It means that users in searching for results in ill-defined domains prefer to combine results of many lines of automated reasoning, or to consult many knowledge sources. This motivated us to pay more attention to searching capabilities in our rule approach, in order to find, as much as possible, rules that are relevant for the given problem (expert rules).

Our approach is based on treating rules as first-class citizens in the given specification of the problem domain (domain ontology), which enables reasoning on rule “instances” in the similar manner as in the case of ordinary facts. It means that we can treat searching as an inference process, benefiting from the possibility to combine some explicit information (rules) with others, in order to get some not explicitly stated rules. This searching scenario exploits all semantic means in order to find useful rules for particular task, similarly as in our previous scenarios for semantic-based searching for information on the Semantic Web (Stojanovic et al. 2001). Particularly we use the ontological background to form relevant queries and to find all relevant rules (it means that we use all ontological properties e.g. hierarchies, axioms to expand a set of retrieved relevant rules). The prerequisite is that such common-shared understanding already exists in the community, which means that rules, formatted in the presented RDF formalism and described using domain specific vocabulary (ontology), are published on the Web.

We benefit in several ways from using these ontological means in searching:

- it is possible to find expert rules that are not explicitly stated in the rule base, but which can be constructed by regular combination of existing rules,

- retrieved rules can be ranked semantically, according to the distance in the hierarchy between the term which is searched for and the term which is retrieved (this is very important for generating the selecting strategy in the case of the information overload),

- in case that no results are found, we could change our search, searching for terms upper in the hierarchy (over-generalisation, for example, a query that searches for 'Students can be replaced by query for 'all People').

Searching for rules can also be enriched with uncertainty or trust bias (for example, searching only for rules with certainty higher than 0,75).

Our idea is to expand the approach in using meta-knowledge (rules) about rules (for example rules about "conferences" are related to rules about "travelling", or rules with inverse conclusions can be related to each other), in order to make intelligent assistance to the user`s problem solving. This meta-knowledge should reflect the general experience (knowledge) obtained in the whole process of the knowledge-base development (the formal expression in the form of rules is the last step in this refinement of expert knowledge). Using such meta-knowledge, we develop resilient knowledge base that will be heavier in connections and hyperlinks and, therefore, resilient to change and reuse (Gil and Ratnakar 2001).

Conclusion

The current efforts in the Semantic Web Community are mainly focused on representing factual information in the World Wide Web, so that these information can be used by machines not just for display purposes, but also for automation, integration, and reuse across applications. However, communication between human/machine agents assumes the exchange of some more expressive statements, for example sharing clausal/conditional statements, presented in the form of rules. Such statements can also be basic elements in the full automation of web services.

In this paper, we presented a proposal for encoding a rule statement using W3C metadata representation standards. We discussed the representation requirements and the characteristics of the proposed rule language. Such representation of rules can serve as the backbone for a web-enabled knowledge management process. In the presented usage scenario, we focus on the knowledge sharing phase in that process, i.e. on the searching for relevant knowledge (rules) on the Web.

There are more benefits of the proposed approach: (i) machine-understandable representation of knowledge in the form of rules on the web, (e.g. possibility to publish rules on personal web pages), (ii) RDF compatible rule format, which can be used as the interchange format for sharing business rules between different tools/systems on the Web and (iii) very expressive description of the content of existing knowledge sources on the Web, including non-textual documents, e.g. images, is enabled. That will enhance the precision and recall in searching for relevant knowledge sources.

Our future work will be oriented towards building a rule management system, which will support efficient capturing, sharing and using rules on the Web, enabling in that way the development of a (semantic) web-based Knowledge Management system.

References

Abecker, A., Bernardi, A., Hinkelmann, K.,Kühn O., Sintek, M. 2001 Towards a Technology for Organizational Memories. IEEE Intelligent Systems & Their Applications 13(3) May/June 1998

Boley, H., Tabet, S., Wagner, G. 2001 Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In Proc. Semantic Web Working Symposium (SWWS'01). Stanford University, July/August 2001.

DAML 2001, DAML Language, http://www.daml.org/2001/03/daml+oil-index.html, 2001

Erdmann, M., and Studer, R. 2000 How to structure and access XML documents with ontologies. Data and Knowledge Engineering, DKE(36) (3): 317-335, 2000

Gil, Y., Ratnakar, V., 2001 „TRELISS: An Interactive Tool for Capturing Information Analysis and Decision Making", Internal project report, USC/ISI, 2001

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P. F. 2001 OIL: An ontology infrastructure for the Semantic Web. IEEE Intelligent Systems, 16(2):38–44, 2001

W3C Working Group 1999, W3C Resource Description Framework (RDF) Model and Syntax Specification. http://www.w3.org/TR/REC-rdf-syntax/, February 1999.

Brickley, D., Guha. R.V. 2000, Resource Description Framework (RDF) Schema Specification 1.0. Technical report, World Wide Web Consortium (W3C), 2000.

Sintek, M., Decker, S. 2001, TRIPLE - an RDF Query, Inference, and Transformation Language. DDLP'2001

Sowa, J. 2000, Knowledge Representation, Brooks/Cole, 2000

Staab, S., Schnurr, H-P., Studer, R., Sure Y. 2001 Knowledge Processes and Ontologies. IEEE Intelligent Systems. 16(1), January/February 2001.

Stojanovic, N., Maedche, A., Staab, S., Studer, R., Sure, Y. 2001, SEAL — A Framework for Developing SEmantic PortALs. ACM K-CAP 2001. October, Vancouver.

Berners-Lee, T. 2001a Comments on webizing applications, 2001, http://www.w3.org/DesignIssues/Webize.html

Berners-Lee, T., Handler, J., Lassila, O., 2001, The Semantic Web, Scientific American, May, 2001,

Berners-Lee, T. 2001b, Using XML for Data, W3C Design Issues, 2001

W3C 2000, Extensible Markup Language (XML) 1.0 (Second Edition) World Wide Web Consortium Recommendation; http://www.w3.org/TR/REC-xml.

Conen, W., Klapsing, R. 2000, A logical interpretation of RDF. In Linkoping Electronic Articles in Computer and Information Science, Vol 5. Linkoping University Electronic Press, December, 2000.