A Framework for Knowledge Management on the Semantic Web

Nenad Stojanovic, Siegfried Handschuh
AIFB Institute
University of Karlsruhe
Englerstrasse 11
D-76131 Karlsruhe
Germany
{nst, sha}@aifb.uni-karlsruhe.de

ABSTRACT

The Semantic Web can be a very promising platform for developing knowledge management systems. However, the problem is how to represent knowledge in the machine-understandable form, so that relevant knowledge can be found by machine agents. In this paper we present a knowledge management approach based on RDF-compatible format for representing rules and on a novel technique for the annotation of knowledge sources by using conditional statements. The approach is based on our existing Semantic Web tools. The main benefit is high improvement in the precision by searching for knowledge, as well as the possibility to retrieve a composition of knowledge sources which are relevant for the problem solving.

Keywords

Semantic Web, Conditional statements, Knowledge Management

1. INTRODUCTION

The Semantic Web is the new generation of the World Wide Web, based on the semantic network knowledge representation formalism, which enables packaging information in the form of object-attribute-value statements, so called triplets. By assuming that terms used in these statements are based on the formally specified meaning (for the community of interest), i.e. ontologies, these triplets can be semantically processed by machine agents. Most of the current Semantic Web applications are based on using such atomic statements as pure facts, which we can reason about. So, a machine agent can understand information that a concrete patient, who suffers from disease X, is treated by medicine Y. Moreover, the agent can use this information in the communication with other machine agents (e.g. to make an appointment with the doctor W), making the vision of the Semantic Web real [1]. But useful statements, which can be exchanged between agents, are not always related to concrete individuals - instances (e.g. patient X, disease Y), but also to a group of individuals with some common characteristics (e.g. statements about female patients older than 60 who suffer from disease Y). Moreover, atomic statements could be combined in a more expressive way as simple conjunction, for example in the conditional form (e.g. Precondition: the patient is male and suffers from X; Action: he has to be treated by medicine Y). On the implementation level this form can be represented using the If-Then statements, forming in that way reasoning atoms for inference- and trust- services on the Semantic Web.

From the web-user point of view, the existence of more expressible statements in a machine-readable format means possibility to find more easily such conditional statements that are relevant for current problem solving. Moreover, conditional statements can be used for "indexing" the content of web resources in a more expressive way than conjunction of keywords/concepts or general metadata (incl. relations). That will enable the formation of more sophistic, context-aware, queries and consequently, it can improve precision and recall in the searching for knowledge. For example, the well-known document-indexing problem whether the document indexed by keywords "aspirin" and "headache" is about how aspirin cures headache or how aspirin causes headache, can be resolved very easily by using proposed conditional statements.

In order to use conditional statements in a knowledge management scenario on the Semantic Web, one needs a machine-understandable representation of such statements as well as powerful mechanisms for creating and manipulating them. In our previous work we analysed the requirements for representing conditional information (i.e. knowledge) in a machine-understandable format and proposed an RDFS format for representing rules, RDFRule [2]. Briefly, that format enables the representation of Horn rules, extended by the uncertainty factor and some model/context information. Since each rule contains a set of premises and a set of conclusions (both of these sets can be empty), the proposed format can be used for the description of conditional statement in the form Precondition-Action. In this paper we present an application framework for managing knowledge sources on the Semantic Web, by using presented conditional descriptions for a more effective searching for knowledge. The framework is based on our existing Semantic Web tools (OntOMat, OntOMat-SOEP, OntOMat-REVERSE, OntOMat-CRAWL, Ontobroker)[1], which should be slightly extended in order to operate with more expressible data format (conditional statements). This framework is elaborated in the next section.

2. THE KNOWLEDGE MANAGEMENT FRAMEWORK

The main process in a knowledge management system is the possibility to find knowledge sources, which are relevant for the problem at hand, as well as the process of providing knowledge sources, which can be used in resolving some problems. From the point of view of the knowledge formalisation, these knowledge sources can be divided into two categories: formal expert rules and (multi-media) documents. In order to enable more efficient searching for the knowledge that is contained in this second category, the content of the documents is indexed by using some ontology-based statements. In our approach these statements have conditional form: Precondition-Action, which enables us to use the same logical mechanisms in the management of both categories of knowledge sources. Moreover, a searching for relevant knowledge can result in some expert rules and/or some documents.

The Figure 1 sketches the proposed framework for knowledge management on the Semantic Web, which reflects the variety of knowledge transformations in this distributed environment: knowledge can be collected from various sources and in different formats, then stored in the common representation formalism, processed in order to compute interdependencies between knowledge items or to resolve conflicts, shared/searched and finally used for problem solving.

Therefore, our knowledge management approach encompasses the following processes [3]: 1. Knowledge Capturing, 2. Knowledge Representation, 3. Knowledge Processing, 4. Knowledge Sharing and 5. Using of Knowledge. All processes are related somehow to domain ontology. Since ontology is a domain model, it contains a set of domain axioms which are used for deriving new information – that is the task of an inference engine. In the following we describe these processes.

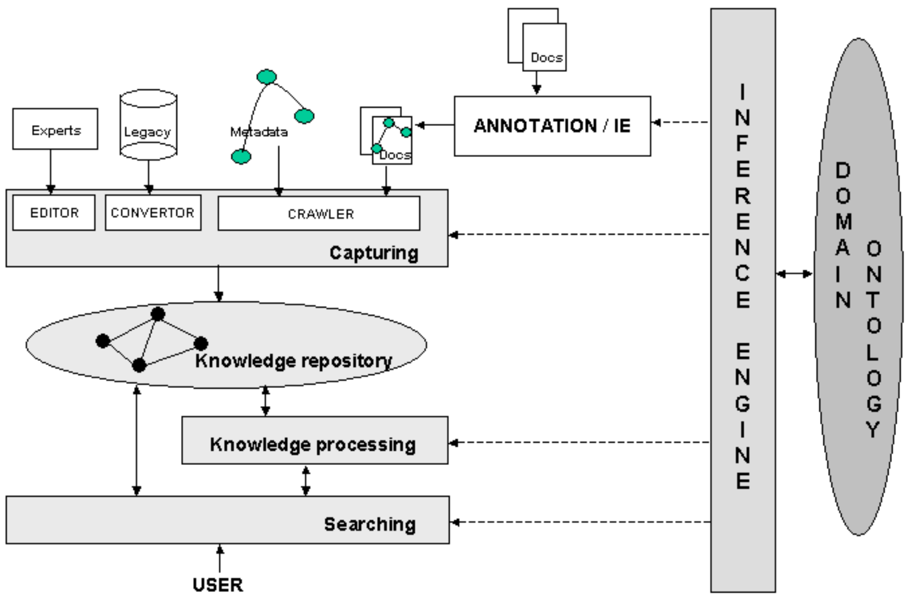

Figure 1: Proposed knowledge management framework

1. We identify four types of knowledge sources, which could be treated in the knowledge capturing phase: expert knowledge, legacy (rule-base) systems, metadata repositories and documents. For each of them we associate our Semantic Web tools:

a) Expert knowledge in the form of rules can be captured using our Simple Ontology Editor Plug-In – OntOMat SOEP, an ontology editor that is extended with rule-editing capabilities. SOEP provides structure as well as vocabulary, i.e. lexical layer of the domain ontology, for the rule creation. Although these rules are related to domain ontology, they are not treated as axioms in that ontology. The ontological axioms should be always-true statements, which is not the case for expert rules. SOEP saves rules directly in the RDFRule format.

b) Legacy rule-bases are very valuable sources of sharable knowledge, which can be consulted in solving some problems, either for free or for some price. The focus is not on collaborative problem solving via querying the federation of

[1] http://kaon.semanticweb.org, http://ontobroker.semanticweb.org

rule bases, but in the creating high-specific expert bases, by importing relevant (for the given task) rule chains from those rule bases. The prerequisite is to have a mechanism to convert legacy rule-bases into rule-interlingua. The potential candidate for the common-accepted rule markup is RuleML [4]. We plan to use our OntOMat-REVERSE, the tool which translates the content of a relational database into an ontology represented in the RDF, for the support of this translation into RDFRule.

c) Metadata dispread on the web should be the primary knowledge source for sharing knowledge in the future. In order to make that sharing more efficient some mechanisms for knowledge packaging and knowledge trading/pricing are needed [5]. Our OntOMat-CRAWL, ontology-focused and metadata-aware crawler already has the capabilities to collect web documents (metadata) that fit the given “knowledge” model, so that the adaptation to rule-crawling is straightforward.

d) The previous three types are related to formally stated knowledge, which can be processed by machine agents. Knowledge in the documents is informally represented, but the content of a document can be formally stated by ontology-based indexes. The underlying process is called semantic annotation and it is supported by our OntOMat annotation framework [6]. By using some information extraction (IE) techniques it is possible to make annotation semi-automatically. For example, we plan to extend OntOMat with techniques for the extraction of the tabular content in order to convert tabular information into a set of rules automatically.

2. Knowledge repository is a relational database organised in a way that enables efficient storing and access to RDF metadata. This repository can be seen as a RDF repository.

3. Knowledge processing component enables efficient manipulation with the stored knowledge, especially graph-based processing for the knowledge represented in the form of rules, e.g. deriving dependency graph or consistency checking

4. Knowledge sharing is realised by searching for rules that satisfy the query conditions. In the RDF repository rules are represented as reified RDF statements and while in RDF any statement is considered to be an assertion, we can view an RDF repository as a set of ground assertions in the form (subject, predicate, object). Rules are also related to domain ontology, which contains domain axioms used for deriving new assertions. Therefore the searching is realised as an inferencing process. We use Ontobroker, main memory, deductive, object oriented database system, which inferences using RDF inputs also. To note that our system treats facts and queries as rules without the rule body and the rule head, respectively. This facility enables using the SOEP editor as a query interface.

5. Using of the knowledge is related to our semantic web-enabled knowledge portals scenarios [3], [7]. The main advantage of our approach is using a conditional statement for the semantic annotation of knowledge sources. In that way we put statements used in the annotation into the context of each other, which consequently leads to efficient searching for knowledge. Moreover, annotating knowledge resources using Precondition-Action statements enables semantic hyperlinking of each two resources, which satisfies the condition that the Precondition part of one annotation, subsumes the Action part of the annotation of another resource. In that way querying for a problem can result in a composition of documents, which cover problem solving. This is a very important process in knowledge management or e-learning search.

3. CONCLUSION

In this paper we presented an application framework for realizing a knowledge management system on the Semantic Web. The proposed framework is mainly based on our existing Semantic Web tools. The approach introduces two new aspects, which could enhance applicability of the Semantic Web tools in real-world applications: (i) rules as the first class citizens on the Semantic Web and (ii) semantic annotation by using conditional statements. The benefits of the proposed approach are manifold: an integration platform for various rules sources and rules format, more precise search for knowledge sources by using conditional statements, machine-processable description of the content of the tabular- and graphic-based resources, a possibility to compose various knowledge sources in solving some rare difficult tasks.

4. REFERENCES

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

2. N. Stojanovic, Lj. Stojanovic. Searching knowledge in the Semantic Web, accepted for FLAIRS 2002, Special track on Semantic Web

3. S. Staab, H.-P. Schnurr, R. Studer, Y. Sure: Knowledge Processes and Ontologies. IEEE Intelligent Systems. 16(1)

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

5. S. Handschuh, S. Staab, A. Maedche. CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. ACM K-CAP 2001. October, Vancouver.

6. D. Skryme. *Capitalizing on Knowledge: From e-business to k-business*, Butterworth-Heinemann, 2001

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