S^{2}REd: A Semantic Web Rule Editor

Thetida Zetta[1], Efstratios Kontopoulos[2], Nick Bassiliades[3]

[1] School of Computer Science & Informatics, Cardiff University, Cardiff, UK
`thzetta@gmail.com`
[2] School of Science & Technology, International Hellenic University
GR-57001, Thessaloniki, Greece
`e.kontopoulos@ihu.edu.gr`
[3] Department of Informatics, Aristotle University of Thessaloniki
GR-54124, Thessaloniki, Greece
`nbassili@csd.auth.gr`

Abstract. A key factor in the further progress of the Semantic Web is the development and wide-spread usage of rule- and logic-based applications. However, there is an evident lack of software tools that can assist end-users in developing such applications. Consequently, users usually resort to more generic tools that offer support at a syntactic level, but prove inadequate in semantically supporting the user. This paper presents S^{2}REd, a Semantic Web rule editor that introduces a supplementary layer of semantic assistance during rule base development. The tool offers semantic assistance via: (i) The Semantic Tag Mapping window that provides a meta-modeling facility for generating schemas over various rule language versions and, (ii) the Namespace Dialog window, for loading ontologies that serve as the underlying vocabulary for expressing rule atoms. S^{2}REd assists in developing RuleML rule bases, but is equally suitable for any other XML-based syntax for representing rule sets.

Keywords: Semantic Web, RuleML, Rule Bases, Rule Editor, Ontologies.

1 Introduction

Since the basic infrastructure of the Semantic Web has been established (content representation, ontologies), efforts have shifted towards logic and proofs. The development of these two aspects will allow systems to infer new knowledge from existing information, assisting them in explaining their actions, sources and beliefs and, eventually, increasing user trust. More specifically, the focus is mainly on integrating rules and ontologies (e.g. *OWL2 RL* [8]) and on standardizing rule representation in the Semantic Web (e.g. *SWRL* [4], *RuleML* [3] and *RIF* [5]).

However, although logic- and rule-based applications play a vital role in the materialization of the Semantic Web vision, there is a narrow and often explicitly domain-specific variety of appropriate tools, like e.g. generic and adaptable rule and ontology editors. Consequently, users typically resort to existing applications for purposes that by far exceed their given array of functionality. For instance, in order to

deploy a RuleML rule set, one could make use of a plain XML editor, since RuleML is in essence an XML-based language and there exist no dedicated RuleML editors.

An XML editor indeed offers support at a syntactic level during the development of XML-based documents, but there is an evident lack of parallel semantic support. This lack of semantic support involves the rule structure itself, as well as the data model, to which these rules refer to. The problem becomes even more apparent, when moving "upwards" to the domain model, with rules containing class and attribute names that drastically raise the need for semantic support during development.

This paper presents S^2REd (*Syntactic-Semantic Rule Editor*), a rule authoring tool that adds a supplementary level of semantic support during rule base development, which cooperates with established techniques and tools for syntactic support.

2 System Description

This section thoroughly describes the S^2REd tool[1], focusing on its two novel modules: The *Semantic Tag Mapping window* (*STMw*) and the *Namespaces Dialog window* (*NDw*). The section also includes the rest of the system features.

2.1 Underlying Rule Languages

Currently, S^2REd operates with RuleML and OO RuleML, but can seamlessly operate with any other XML-based rule language as well (e.g. SRML, XRules, XRML, JessML etc.). The latest RuleML version 1.0 is supported, along with recent previous versions 0.89, 0.90 and 0.91. SWRL is not yet supported, as it is tightly coupled with OWL ontologies, while S^2REd supports RDF/S representations. SWRL support is nevertheless included in our plans for future improvements.

2.2 Architecture

Fig. 1 displays the architecture of the system. The main module is the *Rulebase Editor*, the subsystem interacting with the end-user. The Rulebase Editor is tightly integrated with the *Semantic Tag Mapping window* (*STMw*) and the *Namespaces Dialog window* (*NDw*) that constitute the main means for semantic support.

As illustrated in the figure, S^2REd permits loading external schemas that define the rule language syntax. The *Syntax Loader* module loads an external schema and feeds it to the parser. The latter detects all elements and attributes and temporarily stores them in memory, so that they can be used later. During this phase, the system detects `<include>` and `<redefine>` elements, following the corresponding URLs and collecting the retrieved elements and attributes at these latter locations.

Besides offering the capability of loading custom external schemas, the software also gives the option of loading predefined RuleML schemas (v.0.89-v.1.0). The process of following the `<include>` and `<redefine>` elements described above has

---

already been executed for the predefined schemas and the corresponding structures that contain the collected elements and attributes have been stored in respective files. When loading one of these schemas, users save time by not needing to reinitiate the detection phase. Lastly, the application commences each time having as default schema the one that was loaded during the last time S^2REd was executed.
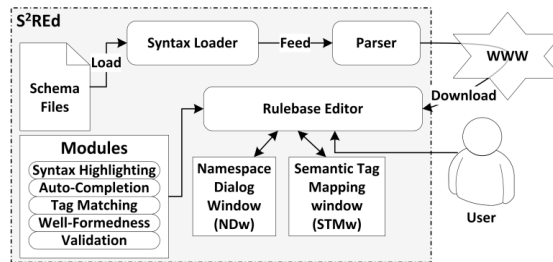

Fig. 1. S^2REd Architecture.

2.3   Functionality

Fig. 2 displays the main window of the program. The menu bar on the top consists of the menus that launch the main functionalities of the tool. The most important menus are the *File* and *Properties* menu:
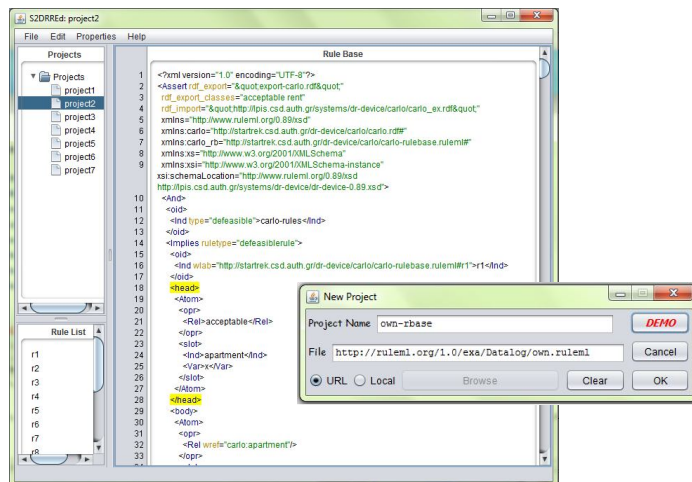

Fig. 2. S^2REd Main Window and New Project Window.

- The *File* menu contains options for managing projects and rule bases. Each project is assigned a distinct folder space in memory, containing the rule base, along with additional files that contain custom, project-specific properties.
- The *Properties* menu contains the more sophisticated functionalities for syntactic and semantic support during the development of rule bases. This menu allows us-

ers to load schema files for the rule base under development (see previous section). There is also the option of checking the well-formedness and validity of the RuleML file, while the last option launches the STMw and NDw windows that constitute the backbone of the system's semantic support and are described later.

The bulk of the main window is occupied by the rule base editor itself, while the two side panels on the left-hand-side are the *Projects Tree* and the *Rules List*. The former displays a tree of available projects, while the latter features a list of all rules contained in the loaded rule base. By clicking on a rule name, the cursor moves to the corresponding rule in text area. As new rules are added into the rule base, the rule list is automatically refreshed with the new rule names.

New S^{2}REd projects are created via the New Project window (see Fig. 2) that requests a project name and a rule base assigned to the project. Only one rule base is assigned to each project, which may exist on the user's local disk or on the Web; in the latter case the system downloads the remote file and locally stores a copy. After clicking the OK button, the project folder is created containing the rule base named after the project. The existing projects appear by expanding the Projects node in the main window of the system (see Fig. 2).

2.4 Code Editor and XML-Centered Features

Since S^{2}REd is in essence a text-based code editor, emphasis was given in incorporating a variety of popular features that improve user-friendliness and efficiency: syntax highlighting, spell checking, brace matching, pretty-printing, code indentation and code completion. Two additional features that have not yet been integrated into S2REd are: code folding and code refactoring, which are planned for addition in the upcoming versions of the software.

Additionally, S^{2}REd integrates XML-authoring functionalities. The corresponding modules for checking the *well-formedness* and *validity* of an XML document are available via the *Properties* menu, ensuring that the document is syntactically correct and complies with an external schema that defines its grammar. Only a well formed rule base can be saved; otherwise, the system displays an error message. Also, the status bar at the bottom of the main window verifies the validity of the rule base, by displaying the message "Valid rule base", otherwise a corresponding error message is shown. A rule base must be saved before performing the validation.

2.5 Semantic Tag Mapping (STM)

After loading a schema (see previous subsection) S^{2}REd displays the *Semantic Tag Mapping window* (*STMw* – See Fig. 3), where the user manually defines the correspondence between the tags of the loaded vocabulary and the various concepts of the rule base. In essence, STMw provides a meta-modeling facility for generating schemas over various RuleML language versions. A meta-model provides a schema for semantic data, specifying what elements may be contained in the model and how they

relate to one another [2]. In reality, the meta-model is a specification of a domain-specific modeling language.

An STM meta-model is a collection of notions that are the main "building blocks" (i.e. vocabulary), via which an understanding of the given domain (various RuleML versions) is provided. It provides a means of communication between S^2REd and the end user, by facilitating common grounds of "understanding" over the notions of "rule", "rule head", "rule body" etc.
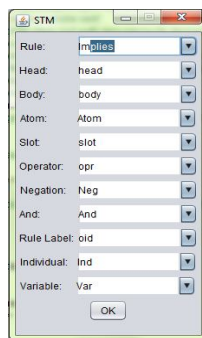

Fig. 3. The Semantic Tag Mapping window.

STM is a primary, novel contribution of this work, offering a wide range of notions to handle, while the list is easily customizable as well. The user is given the option of choosing the tag/element name he/she considers more suitable from a list that contains all the elements and attributes of the loaded schema. The latter list promotes user-friendliness, by also providing auto-completion features during typing.

For the schemas already included (v.0.89-v.1.0 – see previous subsection) the STMw table is already filled; the process of manually matching tags and concepts is necessary only when loading a custom schema for the first time.

2.6 The Namespace Dialog Window

The second means for semantic assistance offered by the tool is represented by the *Namespace Dialog window* (*NDw* – see Fig. 4) that lets the user determine which RDF/XML namespaces will be used by the rule base. Inspired by previous work of ours [6], S^2REd treats namespaces as addresses of input RDF/S ontologies that contain the vocabulary for populating rule atoms.

Namespaces can be manually entered by the user through the NDw. Note that the window displays only those namespaces that actually correspond to RDF and/or RDF/S documents. The added documents are checked for syntactic consistency and are confirmed to contain RDF/S declarations. The user can also manually "discover" more namespaces, by pressing the "…" button next to each namespace entry. Similarly to the process described before (following the `<include>` and `<redefine>` elements in XSD schemas – see Section 2.2), S^2REd downloads the namespace documents contained within this document and repeats the above namespace hunting pro-

cedure. When a new namespace is discovered that is not already contained in the NDw, it is displayed unchecked.

Fig. 4. The Namespace Dialog window.

The namespaces that have been manually selected (i.e. checked) by the user to be included by the system are analyzed and all allowed class and property names are retrieved. These names are then used throughout the authoring phase of the RuleML rule base, constraining the corresponding allowed names that can be applied and narrowing the possibility for errors on behalf of the user.

S²REd distinguishes properties in *owned*, *inherited* and *subsumed*: owned properties of a class are those properties that explicitly have this class in their domain set. Inherited properties of a class are those properties that have (at least) a superclass of this class in their domain. Finally, subsumed properties of a class are those properties that have (at least) a subclass of this class in their domain. This distinction may not seem too practical, but although the domain of a subsumed property of a class C is not compatible with class C itself, it can still be used in the rule condition for querying objects of class C. This implies that the matched objects will belong to some subclass C' of class C, which is compatible with the domain of the subsumed property. For example, consider two classes A and B, where B is a subclass of A and a property P, whose domain is B. It is allowed to query class A, demanding that property P satisfies a certain condition; however, only objects of class B can possibly satisfy the condition, since direct instances of class A do not even have property P.

2.7 A Use Case Example

This subsection describes a use case example that illustrates S²REd's functionality:

1. Create a new web-based (not local) project in S²REd.
2. Give a project name and a rule base URL (you can choose a sample rule base from http://ruleml.org/1.0/exa/, where a list of RuleML examples is maintained).
3. Make sure the schema for the respective RuleML version (v.1.0) is loaded.
4. Open STMw and verify that the correct tags are associated with the appropriate rule base notions.
5. If ontology terms are needed during rule base development, open NDw and add a new namespace (a prefix plus a URL, where the RDF/S ontology can be found).

6. Start editing the rule base!

3 Discussion and Future Work

This paper argues that, although logic- and rule-based applications will play a vital role in the realization of the Semantic Web vision, there is a lack of appropriate software tools that can assist end-users towards this direction. Consequently, users often use applications for purposes that exceed their given functionality. The paper presented S^2REd, a rule authoring tool that adds a level of semantic support during rule base development. Currently, S^2REd operates with RuleML and OO RuleML, but can seamlessly operate with any other XML-based rule language as well. As a sample domain of application, S^2REd has been deployed [7] as the rule editor module for DR-DEVICE [1], a defeasible logic reasoner for the Semantic Web. Our plans for the future of the software involve overall improvements in the tool's functionality, as well as an extensive user evaluation along with investigating the scalability of the system.

4 Acknowledgements

5 References

1. Bassiliades, N., Antoniou, G., Vlahavas, I.: A Defeasible Logic Reasoner for the Semantic Web. In: Sheth, A., Lytras, M. (Eds.) Int. Journal on Semantic Web and Information Systems, IDEA Group Publishing, Vol. 2, No. 1, pp. 1-41, 2006.
2. Bézivin, J., Gerbe, O.: Towards a Precise Definition of the OMG/MDA Framework. In: Proc. ASE'01, IEEE Computer Society Press, USA, pp. 273-280, 2001.
3. Boley, H.: The RuleML Family of Web Rule Languages. In: Alferes, J. J., Bailey, J., May, W., Schwertel, U. (Eds.) Proc. 4th Int. Conf. on Principles and Practice of Semantic Web Reasoning (PPSWR'06), pp. 1-17. Springer-Verlag, Berlin, Heidelberg, 2006.
4. Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004, Available at: http://www.w3.org/Submission/SWRL/, last access: Oct-12.
5. Kifer, M.: Rule Interchange Format: The Framework. In: Calvanese, D. and Lausen, G. (Eds.) Proc. 2nd Int. Conf. on Web Reasoning and Rule Systems (RR '08), Springer-Verlag, Berlin, Heidelberg, pp. 1-11, 2008.
6. Kontopoulos, E., Bassiliades, N., Antoniou, A.: Deploying Defeasible Logic Rule Bases for the Semantic Web. In: DKE, Elsevier, 66 (1), pp. 116-146, 2008.
7. Kontopoulos, E., Zetta, T., Bassiliades, N.: Semantically-enhanced Authoring of Defeasible Logic Rule Bases in the Semantic Web. In: Proc. 2nd Int. Conf. on Web

Intelligence, Mining and Semantics (WIMS'l2), ACM, Article 56, pp. 489-492, Craiova, Romania, June 13-15, 2012, 2012.
8. Reynolds, D.: OWL 2 RL in RIF. In: W3C Working Group Note 22 June 2010. Available at: http://www.w3.org/TR/rif-owl-rl/#Background_OWL_2_RL, last access: Oct-12.

6 Appendix: Addressing the Evaluation Requirements

6.1 Minimal requirements

- Requirement 1: S^2REd is primarily addressed to domain experts, i.e. rule base developers, offering a highly versatile tool for XML-based rule base authoring.
- Requirements 2.1-2.3: S^2REd uses a variety of Web-based information sources, like XSD files, RDF/S ontologies and RDF files and RuleML and other XML-based rule base representations, covering requirements 2.1 and 2.2. So far, no practical need has emerged for really big rule bases (requirement 2.3), but this is a criterion we will address in the upcoming evaluations of the system.
- Requirements 3.1-3.3: The tool is relying heavily on Semantic Web technologies (XML, RuleML, Jena Semantic Web Toolkit, ontologies) and the same happens with the underlying data and its meaning (requirement 3.1). The processing of the input RDF/S ontologies produces – among others – useful information (owned/inherited/subsumed properties) (requirement 3.2), while semantic technologies provide to the system a functionality that would not be otherwise possible; without semantic information processing, S^2REd would be just another XML editor (requirement 3.3).

6.2 Additional Desirable Features

- Although the application is not Web based, its interface is user-friendly and intuitive; there has been substantial effort to reduce complexity as much as possible.
- The scalability of the software has not been evaluated yet (see also requirement 2.3), but it constitutes a priority for the upcoming evaluations of the tool.
- S^2REd is novel in the sense that it attempts to fill an existing gap in rule authoring tools for the Semantic Web environment. Its meta-modeling facilities as well as the assistance it offers in integrating ontology terms inside rules make the tool a highly versatile system that can be applied in various domains and rule languages.
- The tool's functionality clearly goes beyond pure information retrieval, since it is a system allowing rule base development.
- Due to S^2REd's versatility, the software can be applied in a variety of domains, demonstrating a clear commercial potential. Notice that, although the tool is at its beta stage yet, more than 330 downloads were made from SourceForge.
- The tool is implemented in Java and is, thus, platform-independent and accessible on a range of devices, but no multilingual support has been integrated yet.