Building up RDF and RDFS Descriptions with Meta-Model Management

Takeshi Morita[1], **Noriaki Izumi**[2], **Naoki Fukuta**[3], **and Takahira Yamaguchi**[1]
[1]Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama-shi, Kanagawa, 223-8522, Japan
[2]National Institute of Advanced Industrial Science and Technology, Japan
[3]Shizuoka University, Japan
{t_morita, yamaguti}@ae.keio.ac.jp

1 Introduction

We propose a tool to manage several sorts of relationships among RDF and RDFS. Our tool consists of three main functions: graphical editing of RDF descriptions, graphical editing of RDFS descriptions, and meta-model management facilities. We focus on the meta-model management as a key concept, which is defined as appropriate management of the correspondence between a model and its meta-model: especially, class and property in a meta-model, and the type of an RDF resource and property in a model. The prototyping tool, called MR^3 (Meta-Model Management based on RDFs Revision Reflection), is implemented by Java language.

2 Concept of Meta-Model Management

In this paper, meta-model is defined as a model expressing components of models: especially the type of RDF resources and RDF properties. RDFS class is a model expressing the type of RDF resource. RDFS property is a model expressing RDF property.

For the most of existing approaches of RDF and RDFS management, visualized representations of RDF data model try to help us to maintain the XML based RDF descriptions. As RDF and RDFS descriptions are described in the same syntax of RDF, there is no clear distinction between RDF and RDFS parts. This forces us to observe the type of resources, in order to pick up the single part of RDF or RDFS.

Since RDF and RDFS can be regarded as relationship between a model and a meta-model, in the concept of meta-model management, RDF and RDFS can be managed separately and maintained their relationship automatically. In some logical framework described before, meta-model concept seems to capture the above correspondence. Such logical framework reminds us to distinguish RDF and RDFS clearly, and can be expected to bring (semi-)automated support of consistency among RDF and RDFS.

Considering the processes to develop RDF and RDFS, each process can be reflected each other. Some modification in RDF models can cause the modification of RDF Scheme and vice versa. So the meta-model management facilities support a user in developing RDF and RDFS in such reflective processes.

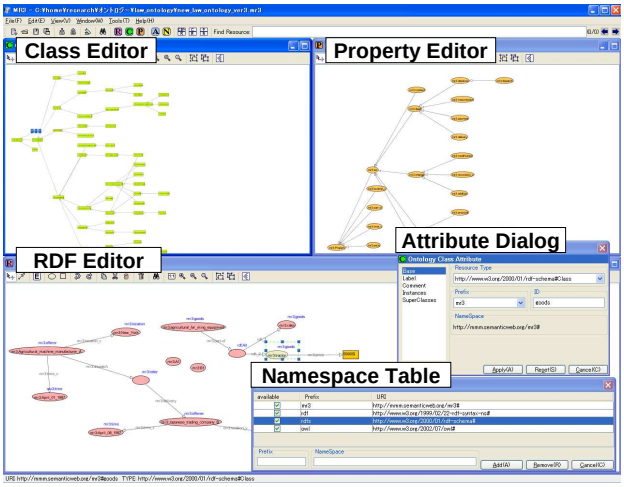

Figure 1: Typical Screen with MR^3

3 Implementation

Figure 1 shows typical screen with MR^3. MR^3 consists of five main windows; RDF Editor, Class Editor, Property Editor, Attribute Dialog, and Namespace Table. The attribute of each element can be displayed and edited via Attribute Dialog. In RDF Editor, the user can edit relationships among an RDF resource, an RDF property, and an RDF literal using a labelled directed graph. In Class editor and Property editor, the user can edit hierarchies for RDFS classes and properties. The user can replace namespaces with a prefix via Namespace Table.

4 Comparison with Related Other Tools

In [Denny, 2004], the survey results of 94 ontology editors currently available to the ontology building community are presented. Our study is also introduced in the survey results. We selected 5 related other tools from the survey results, and compared them with our study from the viewpoint of RDF models construction.

Table 1 shows a comparison of MR^3 with related other studies. The feature of MR^3 is to have the graphical facilities with RDF models and meta-model management facilities. The tools in Table 1 are classified into an RDF based tool or

Table 1: A comparison of MR^3 with related other studies

	Name	Source	Graphical Facilities with RDF Models	Meta-Model Management		Easy to Define RDF properties	Import/Export/Edit		
				O→M	M→O		RDF	RDFS	OWL
RDF based	IsaViz	W3C	△	×	×	◯	◯	×	×
	RDFAuthor	Demian Steer	◯	×	×	◯	◯	△	×
	MR^3	Keio University, Shizuoka University, and AIST	◯	◯	◯	◯	◯	◯	△
Ontology based	KAON OI-modeler	FZI Research Center & AIFB Institute, University of Karlsruhe	△	◯	×	×	△	◯	×
	OntoEdit	Ontoprise GmbH	△	◯	×	×	△	◯	×
	Protégé OWL Plugin	Stanford Medical Informatics, Stanford University	×	◯	×	×	△	◯	◯

an ontology based tool. IsaViz, RDFAuthor, and MR^3 are classified into an RDF based tool and they are concentrating on constructing RDF models. Therefore they have the enhanced graphical facilities with RDF models. KAON OI-modeler, OntoEdit, and Protégé OWL Plugin are classified into an ontology based tool and they are concentrating on creating and managing ontologies. Therefore they partially have the graphical facilities with RDF models, or don't have it. IsaViz and KAON OI-modeler mix and display RDF and RDFS model. Because it is difficult for the user to distinguish RDF and RDFS elements, a burden of the user in RDF model construction is big. OntoEdit displays an ontology and a model separately. However OntoEdit cannot edit RDF elements while displaying the whole RDF model. MR^3 and RDFAuthor displaying an ontology and a model separately.

O→M and **M→O** in Table 1 mean sub facilities for the meta-model management. **O→M** is the facilities to reflect the change in an ontology (RDFS class and property) in a model (the type of an RDF resource and RDF property). **M→O** is the facilities to reflect the change in a model in an ontology. IsaViz and RDFAuthor don't have the meta-model management facilities. Ontology based tools have the facilities of **O→M**. However they don't have the facilities of **M→O**. The feature of MR^3 is to be an RDF based tool and have the meta-model management facilities. Especially, **M→O** is original facilities of MR^3.

Easy to Define RDF properties in Table 1 means whether the tool can define RDF properties without defining the values of `rdfs:domain` and `rdfs:range`. In ontology based tools, it is necessary to define the values of `rdfs:domain` and `rdfs:range` beforehand in order to define an RDF property. RDF based tools can define an RDF property without defining the values of `rdfs:domain` and `rdfs:range`. For RDF model construction, it is consider that the user should be able to define an RDF property freely.

There are two methods to construct an RDF model. One is the method to construct an RDF model using a pre-defined ontology. The other is the method to construct an RDF model without referring to an ontology. A user can construct an RDF model by the latter method using the facility of **M→O** and **Easy to Define an RDF property**. Thanks to the facility of **M→O**, when a user uses a class or a property which is not defined in an ontology in order to construct an RDF model, the class or property is defined in the ontology automatically.

Import/Export/Edit in Table 1 means the function of import, export, and edit for the Semantic Web contents. In this function, a big difference between an RDF based tool and an ontology based tool is about RDF. Ontology based tools can't treat RDF models including anonymous resources. Anonymous resources are often used to describe groups of things. Because most of existing RDF descriptions (e.g. RSS) includes anonymous resources, treating anonymous resources is important to construct RDF models. RDF based tools can treat anonymous resources.

5 Conclusions and Future Work

In order to open our technology to the community of the Semantic Web, the system and its source code of MR^3 is provided via our web site [MR3, 2003]. At present, there are about 250 user registrations and some users gave comments to MR^3. We'd like to reflect comments as our future work.

We are developing an additional plug-in module for MR^3 corresponding to our another work [Morita *et al.*, 2004]. In [Morita *et al.*, 2004], we propose a domain ontology development environment called DODDLE-OWL. DODDLE-OWL generates domain ontologies semi-automatically in OWL format. In the future, we'd like to support constructing OWL Lite level ontologies and its model by integrating MR^3 and DODDLE-OWL.

References

[Denny, 2004] Michael Denny. Ontology tools survey, revisited, 2004. http://www.xml.com/pub/a/2004/07/14/onto.html.

[Morita *et al.*, 2004] Takeshi Morita, Yoshihiro Shigeta, Naoki Sugiura, Naoki Fukuta, Noriaki Izumi, and Takahira Yamaguchi. Doddle-owl: Owl-based semi-automatic ontology development environment. *Evaluation of Ontology-based Tools*, 2004. http://mmm.semanticweb.org/doddle/.

[MR3, 2003] Web site mr^3, 2003. http://mmm.semanticweb.org/mr3/.