

# セマンティック Web のためのメタモデル管理ツール

## A Meta-Model Management Tool for the Semantic Web

森田武史\*1      和泉憲明\*3      福田直樹\*2      山口高平\*2  
 Takeshi Morita      Noriaki Izumi      Naoki Fukuta      Takahira Yamaguchi

\*1 静岡大学大学院情報学研究科      \*2 静岡大学情報学部  
 Graduate School of Informatics, Shizuoka University      Faculty of Informatics, Shizuoka University

\*3 産業技術総合研究所 サイバーアシスト研究センター  
 Cyber Assist Research Center, National Institute of Advanced Industrial Science and Technology

We propose a tool to manage several sorts of relationships among RDF and RDFS. Our tool consists of three main functions: "Graphical Editor of RDF contents", "Graphical Editor of RDFS contents", and "Meta-Model Management Facility". In this paper, meta-model is the model expressing the type of RDF resource and RDF property which are model composition elements. Meta-model management facility supports maintenance of relationship between RDF and RDFS contents. The above facilities are implemented based on plug-in system. We provide basic plug-in modules for consistency checking of RDFS classes and properties. The prototyping tool, called *MR*<sup>3</sup> (Meta-Model Management based on RDFS Revision Reflection), is implemented by JAVA language. Through the experience of using *MR*<sup>3</sup>, we've confirmed *MR*<sup>3</sup> contributes the Semantic Web paradigm from the standpoint of RDFS contents management.

### 1. はじめに

セマンティック Web は, W3C によって標準化が行われている RDF(Resource Description Framework) 及び RDF Schema(RDF Vocabulary Description Language) を基礎とした, 次世代 Web として最も有望な候補である.

RDFS 文書の編集を支援するためのツールはいくつか提供されている [1, 2]. それらの主な機能は, RDFS 文書を RDF データモデルに基づいて視覚的に表示することである. RDF と RDFS は意味的に異なる概念をあらわすが, 同一のデータモデルで表現される. 既存ツールで RDF と RDFS が混在したコンテンツを扱う際には, 両者が混在する. RDF と RDFS を区別することはユーザの負担となる. 本稿では, 上記の問題を解決するために, RDF と RDFS の関係をモデルとオントロジーの関係としてとらえ, その管理を支援するツールを提案する. 同時に, Java 言語を用いて試作したツール: *MR*<sup>3</sup> (Meta-Model Management based on RDFS Revision Reflection)[3] についての簡単な実験及びその評価について述べる.

### 2. メタモデル管理の概念

本稿におけるメタモデルとは, モデル構成要素 (RDF リソースのタイプ及び RDF プロパティ) を表現するモデルである. RDF リソースのタイプを表現するモデルは, RDFS クラスである. RDF プロパティを表現するモデルは, RDFS プロパティである.

RDFS 文書の編集を支援するためのツールはいくつか提供されている. それらの主な機能は, RDFS 文書を RDF データモデルに基づいて視覚的に表示することである. これらのツールを用いることで, RDF コンテンツを視覚的に理解することが可能となる. RDF と RDFS は意味的には異なる概念をあらわすが, 同一のデータモデルで表現される. RDF と RDFS

が混在したコンテンツを既存ツールを用いて視覚的に表示する場合, 両者は混在する. 両者を区別するためには, RDF リソースのタイプに注目する必要がある, ユーザの負担となる.

RDF と RDFS はモデルとメタモデルの関係としてとらえることができる. 本稿では, メタモデル管理として, RDF と RDFS を分離して両者の関係を (半) 自動的に管理する. 具体的には, メタモデルにおける RDFS クラス及び RDFS プロパティと, モデルにおける RDF リソースのタイプ及び RDF プロパティを適切に対応付ける. メタモデル管理の概念を用いることで, RDF と RDFS の間の区別を明確にでき, RDF と RDFS の間の一貫性を自動的に保持することが可能となる. 図 1 は, メタモデル管理を示している. 図 1 の RDFS クラス *ex:Book* と RDF リソース *The Society of Mind* のタイプは対応関係にあることを示している. RDFS クラス *ex:Person* と RDF リソース *Marvin Minsky* のタイプ, RDFS プロパティ *ex:author* と RDF プロパティ *ex:author* も同様に, 対応関係にあることを示している. メタモデル管理では, これらを適切に対応づける.

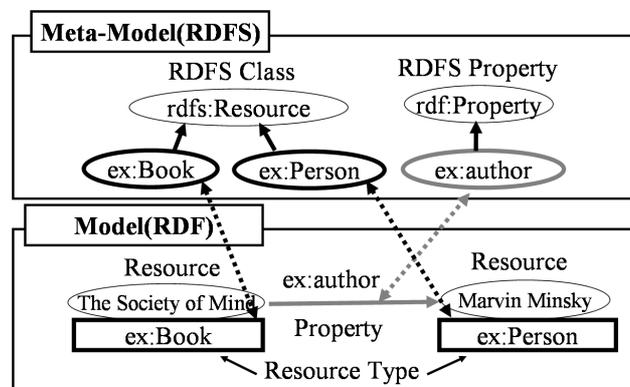


図 1: メタモデル管理

連絡先: 森田武史, 静岡大学大学院情報学研究科, 〒 432-8011  
 静岡県浜松市城北 3-5-1, Tel: 053-478-1473, FAX 053-473-6421, e-mail: morita@ks.cs.inf.shizuoka.ac.jp

### 3. メタモデル管理ツールの設計

メタモデル管理ツールには、主に3つの機能がある。以下で、それぞれについて説明する。本稿におけるデータグラフとは、データモデルを視覚的に表現したものである。RDF要素とは、RDF リソース、RDF プロパティ、RDF リテラルをあらわす。RDFS要素とは、RDFS クラス及びRDFS プロパティをあらわす。

#### 3.1 機能概要

本ツールの機能概要を図2に示す。図2の(1)は、RDF要素の視覚的編集機能である。本機能は、RDFデータモデルに基づいて、RDFにおけるリソース、プロパティ、リテラルの編集を支援する。また、RDF文書からRDFデータグラフへの変換及びRDFデータグラフからRDF文書へ変換する機能をもつ。図2の(2)は、RDFS要素の視覚的編集機能である。本機能は、RDFSのデータモデルに基づいて、RDFSにおけるクラス及びプロパティの階層関係、プロパティのdomain, rangeなどの編集を支援をする。また、RDFS文書から、RDFSデータグラフへの変換及びRDFSデータグラフからRDFS文書へ変換する機能をもつ。図2の(3)は、メタモデル管理機能である。メタモデル管理機能とは、RDF要素及びRDFS要素を編集しても、RDFS要素の一貫性を保持する機能である。3.2節でメタモデル管理機能の具体例を示す。

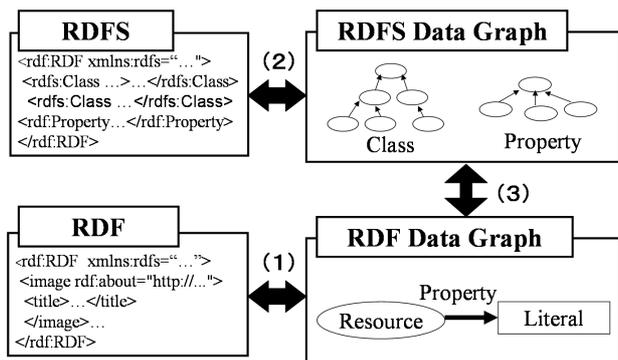


図2:  $MR^3$  の機能

#### 3.2 メタモデル管理機能

メタモデル管理機能は、複数の副次機能から構成される。以下に、メタモデル管理機能の具体例を2つ示す。

##### 例1: RDFS クラス名の変更

RDFS クラス名を変更する場合、変更するRDFS クラスを参照しているRDF リソースのタイプ名 (URI) も同時に変更される。図3は、RDFS クラス名を変更する場合の具体例である。図3の左側はRDFS クラス `ex:Book` を変更する前の状態を表しており、右側はRDFS クラス `ex:Book` を `ex:Biblio` に変更した後の状態を表している。`The Emotion Machine` と `The Society of Mind` は、`ex:Book` をタイプとするRDF リソースである。RDFS クラス `ex:Book` を `ex:Biblio` に変更すると、`ex:Book` と対応するRDF リソースのタイプにも変更が反映される。`The Emotion Machine` と `The Society of Mind` のタイプは、`ex:Biblio` に変更される。

##### 例2: RDF リソースのタイプの変更

ユーザが変更したRDF リソースのタイプに対応するRDFS

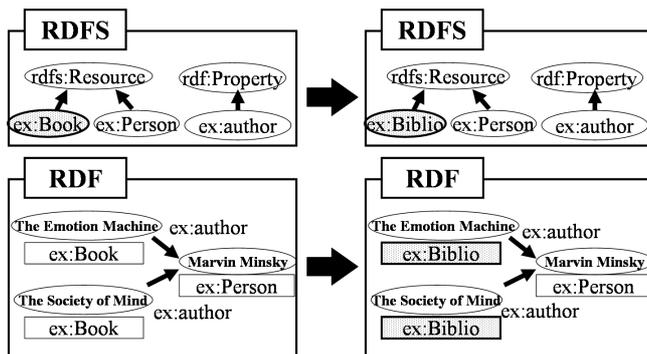


図3: RDFS クラス名の変更

クラスが定義されていない場合、メタモデル管理機能がはたらく。ユーザが変更したRDF リソースのタイプがRDFS クラスで定義されている場合は、 $MR^3$  はRDF リソースのタイプとそれに対応するRDFS クラスを対応づける。定義されていない場合には、ユーザは次の2つのどちらかを選択できる。一つは、ユーザはRDF リソースのタイプを変更する前に参照していたRDFS クラス名を変更することができる。もう一つは、ユーザは定義されていないRDFS クラスを新規に作成することができる。ユーザが変更する前のRDF リソースのタイプが空の場合には、RDFS クラス名の変更は選択できない。図4は、RDFS クラス `ex:Biblio` が定義されていない状態で、ユーザがRDF リソース `The Society of Mind` のタイプ `ex:Book` を `ex:Biblio` に変更した場合の例である。ここでは、整合性を保つためにRDFS クラス `ex:Biblio` の新規作成を行っている。

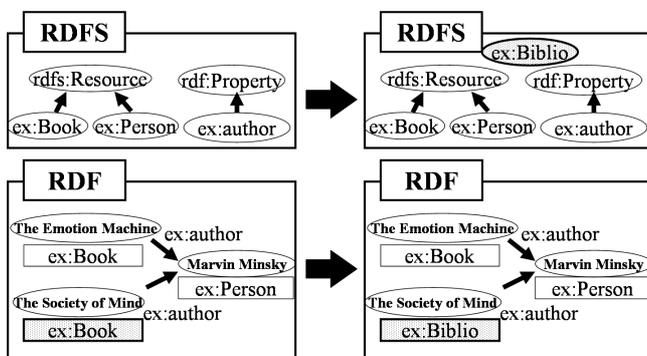


図4: RDF リソースのタイプ変更 (RDFS クラスの新規作成)

その他には、RDFS クラス及びRDFS プロパティの削除、RDFS プロパティ名及びRDF プロパティ名の変更、RDF 文書の読み込みを行った際にも、メタモデル管理機能がはたらく。また、要素名の一意性確保もメタモデル管理機能の副次機能である。

### 4. メタモデル管理ツールの実装

3. 節で述べた設計に基づいて、プロトタイプツール:  $MR^3$  (Meta-Model Management based on RDFs Revision Reflection) をJava 言語で作成した。図5は、 $MR^3$  のシステム構成を示している。ユーザは、 $MR^3$  のユーザインタフェー

スを紹介して、視覚的に RDF コンテンツの編集を行う。システムの入力と出力は RDFs 文書である。MR<sup>3</sup> に RDFs 文書を入力すると、RDF パーサが RDFs 文書の解析を行い、RDFs 文書をモデルとして操作可能にする。MR<sup>3</sup> は、RDF パーサの出力したモデルを内部データ表現に変換し、メタモデル管理を行う。プラグインは、MR<sup>3</sup> が提供する API を利用して構築される。MR<sup>3</sup> は、内部データ表現を RDF ジェネレータの入力となるモデルに変換する。RDF ジェネレータは受け取ったモデルを RDF 文書に変換し、ファイル等に出力する。MR<sup>3</sup> では、図 5 の RDF パーサと RDF ジェネレータを実装するために、HP Labs が提供している Jena Semantic Web Toolkit[4] を利用している。RDF パーサの出力と RDF ジェネレータの入力のモデルは Jena の Model インタフェースを用いている。

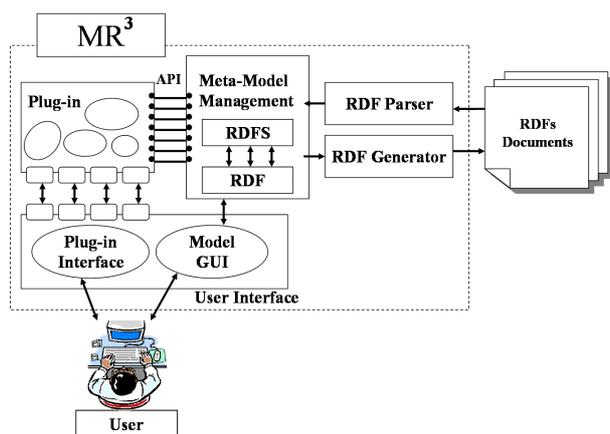


図 5: MR<sup>3</sup> のシステム構成図

ユーザインタフェースには、Model GUI と Plug-in interface の 2 つがある。図 5 の Model GUI は、MR<sup>3</sup> が基本機能として提供しているユーザインタフェースである。図 5 の Plug-in interface は、個々のプラグインが提供するインタフェースを示している。図 6 は、Model GUI のウィンドウ一覧を示している。Model GUI は、RDF エディタ、クラスエディタ、プロパティエディタ、アトリビュートダイアログ、名前空間テーブルの、主に 5 つのウィンドウから構成される。RDF エディタでは、RDF リソース、RDF プロパティ、RDF リテラルの間の関係を有向グラフで表現し、各要素の属性の編集を行うことができる。クラスエディタでは、RDFS クラス間の is-a 関係を表現し、RDFS クラスの属性の編集を行うことができる。プロパティエディタでは、RDFS プロパティ間の is-a 関係を表現し、RDFS プロパティの属性の編集を行うことができる。アトリビュートダイアログには、各要素の属性を表示し、編集することができる。名前空間テーブルでは、要素名の名前空間を指定した接頭辞で置き換えることができる。MR<sup>3</sup> では、RDF モデルを視覚的に表示するために、ユーザインタフェースモジュールとして JGraph[5] を利用している。

MR<sup>3</sup> は、Plug-in によって機能の拡張を可能としている。現在 MR<sup>3</sup> は、Plug-in 作成のための API として、MR<sup>3</sup> のグラフを Jena の Model インタフェースに変換する API と Jena の Model インタフェースを引数にとり、MR<sup>3</sup> のグラフへ変換するための API をいくつか提供している。今後は、RDF と RDFS 間のより厳密な整合性管理を管理するために必要な API など、多様なプラグイン作成を行うための API を提供し

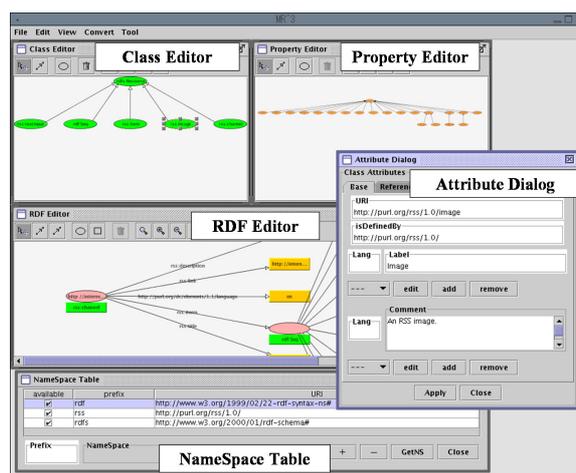


図 6: MR<sup>3</sup> のユーザインタフェース

ていく予定である。

## 5. 評価

MR<sup>3</sup> について簡単な実験を行い、それに対する評価及び考察を行った。実験は、MR<sup>3</sup> に対し RDF と RDFS が混在した文書を読み込ませた際に、適切に RDF 要素と RDFS 要素を分離して表示できることの確認を目的として行った。実験の比較対象には、IsaViz を用いた。IsaViz は、W3C が提供している RDF を視覚的に編集するためのツールで、RDF 文書を読み込んで RDF のデータモデルに忠実に視覚化することができる。本実験で IsaViz を比較対象とした理由は、IsaViz が RDF と RDFS を同様に扱うからである。IsaViz と比較を行うことで、RDF 要素と RDFS 要素を分離する場合と分離しない場合の比較を行うことができると考えられる。

### 5.1 実験と評価

本節では、実験方法とその評価について述べる。本実験では、MR<sup>3</sup> と IsaViz に対し RDF と RDFS の混在した 3 つの文書を読み込ませ、表示内容の比較を行った。1 つ目の文書は、RDFS の基本クラス及び基本プロパティである。1 つ目の文書には、リソース数が 33 個、リテラル数が 60 個、ステートメント数が 158 個含まれている。2 つ目は、Creative Commons のスキーマ [6] である。2 つ目の文書には、リソース数が 25 個、リテラル数が 33 個、ステートメント数が 66 個含まれている。3 つ目は、RSS(RDF Site Summary) のスキーマ [7] である。3 つ目の文書には、リソース数が 29 個、リテラル数が 54 個、ステートメント数が 101 個含まれている。図 7 は、IsaViz に、1 つ目の文書を読み込ませた状態でのスクリーンショットである。図 7 は、RDF データグラフをあらわしている。図 7 から、IsaViz では、RDFS 要素の階層関係を人間が目で見理解することが困難であることがわかる。また、IsaViz では、RDF リソースと RDFS 要素が共に同色の楕円で表現されているため、RDFS 要素を識別するには、RDF リソースのタイプに注目しなければならない。図 8 は、MR<sup>3</sup> に、1 つ目の文書を読み込ませた状態でのスクリーンショットである。図 8 の一番上のウィンドウでは、RDFS プロパティの階層をあらわしている。図 8 の真ん中のウィンドウでは、RDFS クラスの階層をあらわしている。図 8 の一番下のウィンドウでは、RDF データグラフをあらわしている。図 8 から、MR<sup>3</sup> が RDF 要

素と RDFS 要素を分離して表示できることが確認できた。また、 $MR^3$  では、RDF 要素と RDFS 要素の識別や RDFS クラス及びプロパティの階層関係を人間の目で見て容易に理解できることがわかった。

## 5.2 考察

IsaViz と  $MR^3$  の比較を行い、 $MR^3$  のデータグラフの方が視覚的な理解が容易であることがわかった。その理由の 1 つは、RDF 要素と RDFS 要素を分離していることだと考えられる。もう一つは、視覚的編集において編集の妨げとなる要素を表示していないことが考えられる。例えば、 $MR^3$  では、RDFS クラス及びプロパティの `rdfs:label`, `rdfs:comment`, `rdfs:domain`, `rdfs:range` 等のプロパティとその値をグラフに表示せず、編集ウィンドウで編集できるようにしている。また、RDF リソースのタイプについても同様である。特に RDF リソースのタイプは、複数の RDF リソースに参照されるため、グラフを理解する際の負担になると考えられる。上記の理由より、ユーザが最も探したいリソースに着目することができるため、データグラフを理解することが容易になると考えられる。

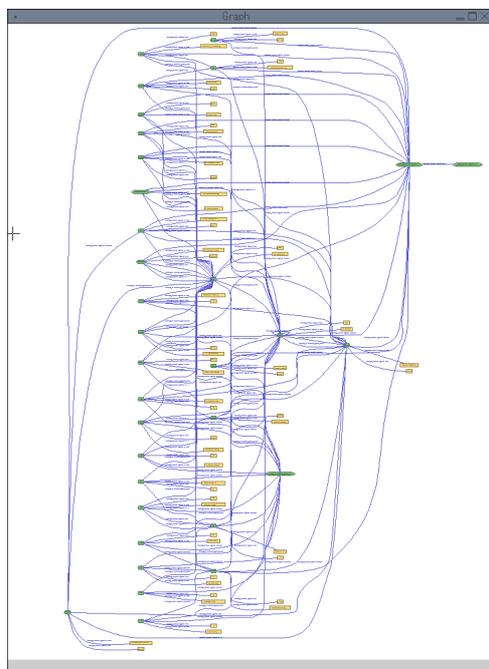


図 7: IsaViz を用いて RDFS 文書をデータグラフで表現したスクリーンショット

## 6. おわりに

本稿では、メタモデル管理の観点から RDFS 文書を管理するツール:  $MR^3$  (Meta-Model Management based on RDFS Revision Reflection) を提案した。RDF と RDFS はモデルとメタモデルの関係としてとらえることができることから、 $MR^3$  では、RDF と RDFS を分離して両者の関係を (半) 自動的に管理することを可能にした。同時に、 $MR^3$  を Java 言語で試作し、試験的な評価を行い、RDFS コンテンツの編集における効率的な支援の可能性を確認した。 $MR^3$  については、公開準備を進めているところである。

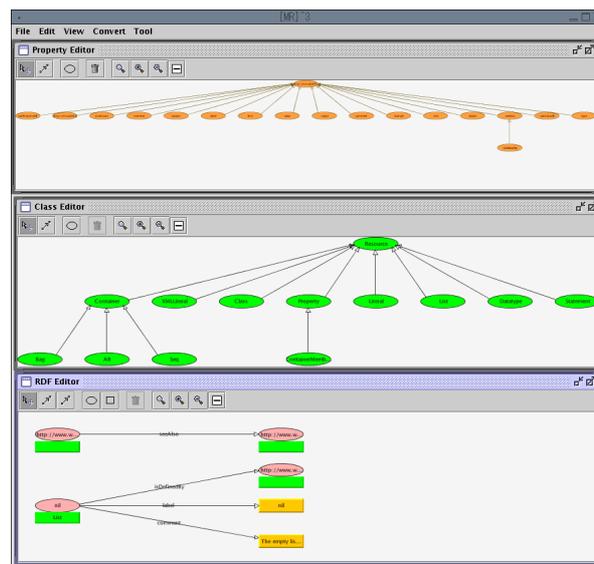


図 8:  $MR^3$  を用いて RDFS 文書をデータグラフで表現したスクリーンショット

今回は、編集作業時の表示に関するユーザの負担軽減という観点からシステム的设计と開発を行った。今後は、ユーザが RDF と RDFS を同時に構築する際、どれだけ RDFS コンテンツの関連管理を自動化できるか、という観点でシステムを発展させる予定である。このために、先行研究 [8] と関連して、計算機可読型辞書とフリーテキストを利用した、セマンティック Web におけるオントロジー構築支援技術との統合などについて考察する予定である。

## 参考文献

- [1] Emmanuel Pietriga, IsaViz, 2003, <http://www.w3.org/2001/11/IsaViz/>
- [2] Protégé, <http://protege.stanford.edu/>
- [3] Noriaki Izumi, Takeshi Morita, Naoki Fukuta and Takahira Yamaguchi, RDF-based Meta-Model Management Environment, In Proc. of ISIR 2003, pages 87-88, 2003.
- [4] HP Labs, Jena Semantic Web Toolkit, 2003, <http://www.hpl.hp.com/semweb/jena-top.html>
- [5] Gaudenz Alder, JGraph, 2003, <http://jgraph.sourceforge.net/index.html>
- [6] Creative Commons Metadata RDF Schema, <http://creativecommons.org/learn/technology/metadata/schema.rdf>
- [7] RSS Schema, <http://web.resource.org/rss/1.0/schema.rdf>
- [8] 中矢尚美, 樽松理樹, 山口高平, “既存の情報リソースを活用するオントロジー構築支援エディタ”, 第 16 回人工知能学会全国大会, 2F2-06(2002)