社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DBS－144 (8)
2008－GN－ 66 (8)
2008／1／24

GRDDLによるコンテンツ変換を用いたオントロジー構築

柳田 憲士郎† 塚本 享治†
東京工科大学大学院†

インターネット上の情報を検索する手法としてキーワードによる全文検索が用いられている。しかし、表記のずれにより求めている情報に到達できない。その解決方法としてセマンティックウェブが提唱されている。セマンティックウェブの現在の状況は、RDF/OWL というメタデータが標準化され、メタデータの構築が各所で進められている。

しかし、現実にはメタデータの構築はあまりうまくいっているとは言いがたい。実在するデータに対してＯＷＬ記述を行うには、ある程度まとまった量のデータが必要となり、そのデータ量の多さの為にオントロジー構築に時間がかかってしまうからである。

GRDDL と SPARQL を用いて、すでに電子化されている既存のデータからオントロジー構築をする方法が推定されている。本研究ではこの方法の実用性を確かめるためにシステムを構築し、オントロジー構築を行ったので、その方法と実行結果について報告する。

Ontology construction using contents conversion by GRDDL

Kenshiro Yanagida† Michiharu Tsukamoto†
Tokyo University of Technology†

The full-text search by keyword is used as a technique for finding information on the Internet. However because of the character string differences it is unable to find the information. Semantic web is proposed to solve this problem. A present situation of the semantic web is the metadata named RDF/OWL is standardized and constructed in each place.

However the construction of the metadata doesn't actually go well. It takes time to construct the ontology for the existing data to describe in OWL.

The method of constructing ontology form existing data using GRDDL and SPARQL is presumed. This paper describes the method and the test result of constructing the system and ontology.

1　はじめに

現在、インターネット上の情報を検索する手法としてキーワードによる全文検索が用いられている。しかし、表記のずれにより求めている情報に到達できないという問題がある。その 1 つの解決方法としてセマンティックウェブが提唱されている。セマンティックウェブは、RDF [1] /OWL [2] というメタデータが標準化され、メタデータの構築の段階に入りつつある。

しかし、現実にはメタデータの構築が進んでいるとは言いがたい。理由はいくつか考えられるが、その一つとして、実在するデータに対して OWL 記述を行うには、ある程度まとまった量のデータが必要となり、そのデータ量の多さの為にオントロジー構築に時間がかかってしまうということがあげられる。

本研究では、GRDDL [3] と SPARQL [4] を用いて、すでに電子化されている既存のデータから情報を再利用することにより、オントロジー構築の一部を自動化するシステムを構築し、その方法の実用性について検証した。

2　オントロジー構築の現状と課題

OWL 記述によるオントロジー構築には Protégé [5] や SWOOP [6] が用いられることが多い。これらのソフトウェアは GUI を用いて手作業でオントロジー構築を行うことに重点をおかれたソフトウェアであり、これらのソフトウェアを用いて、まとまった量のデータをもったオントロジー構築を行う際には膨大な時間が必要となる。

この問題を解決するためには、大量のデータ操作に適したオントロジー構築ツールが必要となる。

本研究では RDB、HTML、XML といった、現在広く普及しているフォーマット形式のデータを用いて、オントロジー構築処理の自動化を図った。

3　本研究が対象としたオントロジー構築自動化部分

オントロジーを構築は主に次の４つのプロセスに分けられる。

(1) クラス設計プロセス
(2) データタイププロパティ設計プロセス
(3) オブジェクトプロパティ設計プロセス
(4) インスタンス入力プロセス

(1) から (3) までのプロセスには、構築したいオントロジーの分野及び OWL について深い知識が必要である。(4) は実データについての部分であり、実用的なオントロジーとして利用するには、(4) のプロセスで生成されるデータが大量に必要となる。

本研究では、(1) から (3) までのプロセスはすでに行ってある OWL 記述オントロジーを対象とする。これらオントロジーに対して (4) のプロセスについて自動化を試みた。

4　GRDDL によるコンテンツ変換

4.1　GRDDL について

GRDDL は、W3C によって提唱された XML ドキュメントが RDF に準拠したデータに変換可能であることを宣言し、XSLT を用いて XML ドキュメントから RDF データを抽出するための規格である。XML ルートタグの要素部分に図 1 のような記述を行うことにより、XSLT を用いて RDF 形式に変換する。また、GRDDL では図 2 のように複数の XSLT を指定することにより、複数の RDF 変換を同時に行うことができ、ひとつの RDF にまとめることが可能である。

```
<root xmlns:grddl='http://www.w3.org/2003/g/data-view#' grddl:transformation="convertxsl">
```

図 1　GRDDL 変換記述

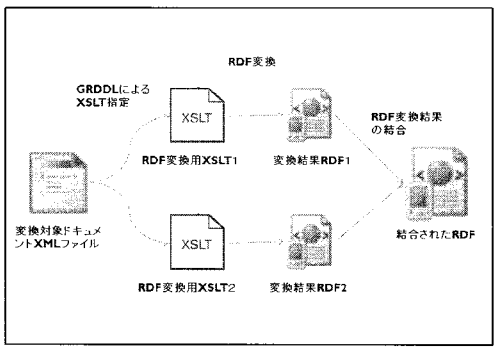

図 2　GRDDL による RDF 変換の流れ

4.2 GRDDL 記述の挿入

GRDDL による変換を行う際には、変換対象ドキュメントに 4.1 の記述を加える必要がある。本研究では変換対象ドキュメント URI と変換用 XSLT ファイル URI の対応関係をリストで持たせた。このリストを元に変換対象となるコンテンツを DOMAPI で読み込み、ルートタグに GRDDL の名前空間、変換用 XSLT ファイルの記述を追加した。複数の変換用 XSLT を指定する場合は XSLT ファイル名の後に改行コードを挿入し、別々の XSLT ファイルであることを認識させる。

4.3 各種フォーマットにおける汎用的なコンテンツ変換

GRDDL を用いて RDF 変換を行う際には、変換対象データに沿った XSLT を記述する必要がある。本節では、各種変換対象フォーマットに対してどのような XSLT を記述したかについて述べる。

4.3.1 XML の変換

XML の RDF 変換を行う XSLT を記述する際には、以下の記述ルールを追加した。図 3 は、下記の制約を設けた際の RDF 変換例である。

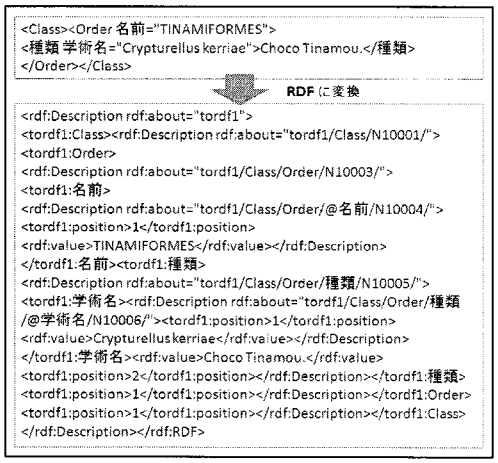

図 3 XML の RDF 変換例

制約内容：
(1) XML のタグ一つ対して、一つのユニークな RDF リソース URI を持たせる。
(2) URI は「XML 参照元 URL/XML の階層パス/ユニークな乱数」という形で指定する。
(3) XML のタグ出現順序の情報を保持するために、「tordf:position」という専用のプロパティを設け、RDF に記述する
(4) XML の属性については、XPATH の指定のように、URI の属性名表記に「@」をつける。
(5) タグの要素が文字列の場合は「rdf:value」プロパティで指定する。

4.3.2 HTML の変換

HTML に対して XSLT を用いて RDF 変換を行うには、HTML を XML 形式としてパースできる形に変換する必要がある。本研究では NekoHTML [7] を用いて XHTML 形式に変換した。この XHTML の head タグに対して図 4 のようなタグを挿入することにより、GRDDL 変換に対応させた。

```
<head profile="http://www.w3.org/2003/g/data-view">
<link rel="transformation" href="convert.xsl" />
```

図 4 GRDDL 対応ヘッダ記述

なお、XHTML からの RDF 変換については、4.3.1 で用いた XSLT を利用した。

4.3.3 RDB の変換

RDB の変換には XSLT の SQL アクセスを行う拡張関数を用いた。変換対象 XML ファイルとして、変換対象データベースのアクセスに必要な情報を記述し、XSLT 変換を行うことにより、RDB から GRDDL を用いて RDF 変換を行うことを可能とした。

RDB の変換を行う XSLT を記述する際には、以下の記述ルールを設けた。

(1) XSLT による RDB アクセスはデータベース単位で行い、データベースに属している全てのテーブルを変換対象とする。
(2) RDF 変換の名前空間は「データベース参照 URL/データベース名/」となる。
(3) テーブル及びレコードに対して URI 記述を行う。テーブルの場合は「名前空間/テーブル名」となり、このテーブルに所属

しているレコードは、テーブル名のあとにユニークな乱数が付与される。

(4) レコードがテーブルに属していることを表すために「名前空間/row」というプロパティで関係性を設ける。

(5) レコードが保持している情報は、「名前空間/テーブル項目名」というプロパティで関係性を設ける。

図5は、上記の制約を元に変換された RDF のデータ構造の例である。

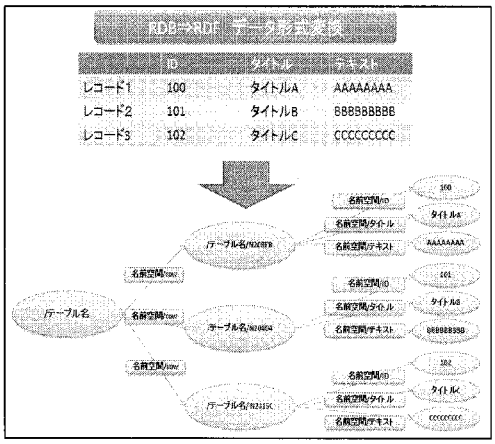

図5　RDB から RDF へのデータ変換例

4.4　専用文章フォーマットに対応したコンテンツ変換

前節では XML、RDB といったデータフォーマットに準拠していれば、どのようなデータであっても変換可能な汎用的な GRDDL 変換手法について述べた。これらの変換手法はデータの記述内容に依存はないが、余分なデータについても変換を行い、意味を持たないデータとして変換する可能性を含んでいる。

本節ではドキュメントの意味構造にできるだけ沿った GRDDL 変換について述べる。

4.4.1　OFFICE ドキュメントの変換

「Word」「Excel」「PowerPoint」といった、世間で広く用いられているドキュメントフォーマットからの変換を試みた。

まず、これらのファイルを変換するにあたって、Microsoft Office Open XML 形式[8]に変換した。変換後のファイルは XML 形式であるため、4.2.1 の手法で RDF 変換は可能である。しかしこの変換手法には以下のような問題点がある。

(1) OFFICEXML は多くの XML ファイルから構成されており、それらの XML ファイル全てに変換処理を行うことは非効率的である。

(2) ドキュメントの仕様上、デザインに関する記述が非常に多くなされており、オントロジーを構築する上では不必要なデータが大量に抽出される。

(3) 図6のように、文章中にフォントの変更があった場合、その部分は別のタグで囲まれる。このため、フォントの変更箇所だけ別の文章とみなされ、必要なデータをただしく抽出できない可能性がある。

(4) Excel の表に属している文字列は参照指定によって別の XML ファイル保存されており、4.2.1 ではこれらの参照指定を RDF に反映させることができない。

```
- <a:r>
    <a:rPr lang="ja-JP" altLang="en-US" dirty="0" smtClean="0" />
    <a:t>右の図の赤丸をクリック後</a:t>
  </a:r>
- <a:r>
  - <a:rPr lang="en-US" altLang="ja-JP" b="1" dirty="0" smtClean="0">
    - <a:solidFill>
        <a:srgbClr val="FF0000" />
      </a:solidFill>
    </a:rPr>
    <a:t>NEXT</a:t>
  </a:r>
- <a:r>
    <a:rPr lang="ja-JP" altLang="en-US" dirty="0" smtClean="0" />
    <a:t>を押す</a:t>
  </a:r>
```

図6　フォント指定によって分割された文章

これらの問題を解決するため、以下の記述ルールに基づいて Office Open XML 専用の XSLT を制作した。

(1) ドキュメントのデザイン指定を目的とした XML は読み込まない。

(2) ドキュメント上に表示されるテキストと直接関係があるタグについてのみ RDF 記述処理を行う。

(3) PowerPoint のタイトル、Excel のセルといったタグには、XSLT に専用の処理を記述する。

(4) <a:p>タグに含まれている<a:t>内の文字

列は出現順序に沿って一つの文章として RDF 記述を行う。

4.4.2 その他ドキュメントにおける意味構造を反映した変換

インターネット上には、あるデザインに沿って機械的に生成されたドキュメントが多く存在している。これらドキュメントの意味構造をより反映させて変換を行いたい場合は、そのドキュメント専用に XSLT を作成することで対応することができる。

本研究では、e-Words[9]の WEB ページを対象とし、用語の説明や、関連用語といった情報を RDF として抽出できる XSLT 記述を行った。

5 変換データからのオントロジー構築

GRDDL によって変換された RDF データは、データ間の関連性が記述されておらず、この状態ではオントロジー（OWL）とは言えない。

本章では、これらのデータに対してどのように OWL 記述を付与し、オントロジー構築を行ったかについて述べる。

5.1 オントロジー構築の全体構成

ここでは、GRDDL を用いたオントロジー構築を行うにあたって、図 7 のようにシステムを構成した。

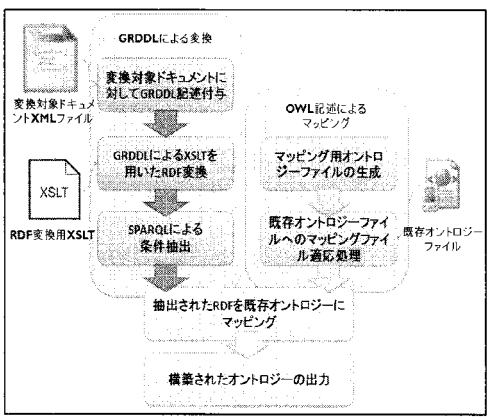

図 7 システム構成図

GRDDL による RDF 化、及び SPARQL によるクエリ処理の実装には、セマンティック WEB アプリケーション開発用フレームワークである jena[10]を用いた。

5.2 SPARQL による条件抽出

GRDDL によって変換された RDF データの中にはオントロジー構築に不必要なデータが存在する場合がある。このような場合、GRDDL の変換で用いる XSLT の記述を変更すれば、必要なデータのみの抽出が可能となる。

しかし、GRDDL によって変換される RDF が複数存在し、これら複数の RDF にまたがった条件によってオントロジーの構築を行いたい場合、XSLT による条件抽出では不十分である。

また、オントロジー構築時に関係性を記述する際に、rdf:type や owl:DatatypeProperty タグといった、クラスやプロパティの種類というものを意味構造にそって明確に指定しなければならない。

そこで、RDF クエリ言語である SPARQL の CONSTRUCT 文を用いて変換された RDF に対して条件抽出を行えるようにした。

これにより、OFFICE ドキュメントに記述されている用語に対しての説明文のみを、HTML 上から取得し RDF 化する、といったことが可能となった。

5.3 OWL によるマッピング処理

本研究では、抽出された RDF に対して関係性を記述する為に新たにマッピング処理用の OWL ファイルを生成することにより、既存のオントロジーと関連性のあるオントロジー構築を行えるようにした。

この OWL ファイルは抽出された RDF に owl:import タグを用いてマッピング処理に対応したデータとして扱えるようにした。

マッピング処理用の OWL を作成するにあたっては、以下のような方法を用いた。

(1) 5.2 で指定したクラスやプロパティと既存オントロジーとの関係性記述には、equivalentClass 及び equivalentProperty タグを用いて行う。
(2) 既存のオントロジーに対して 1 つのマッピング用 OWL ファイルを生成する。

図 8 は抽出された RDF、マッピング用 OWL

ファイル、既存オントロジーファイルの参照関係である。

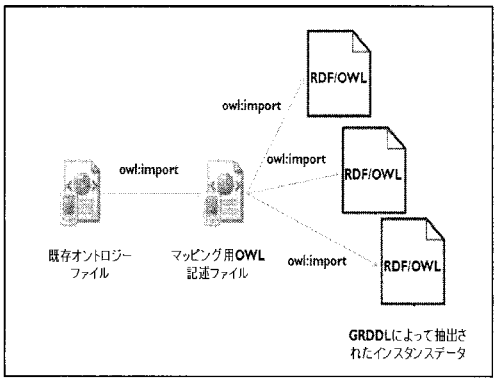

図 8　オントロジーファイル参照関係

6　実験と評価

6.1　構築したオントロジーの利用

本システムを用いて、以下の条件の下オントロジーの構築を行った。

ファイル	概要
変換対象ドキュメント	e-Words 内の用語索引ページ（アルファベット部分）
既存オントロジーファイル	用語についての情報についてのみクラスやプロパティ記述を行ったファイル（インスタンス記述なし）
変換用 XSLT	XML 用の汎用変換 XSLT

図 9　オントロジー構築実験ファイル構成

このシステムで生成されたオントロジーについて Protégé 及び OWL 推論用ライブラリの pellet[15]における読み込み、及び OWL で記述された関係性の反映を確認した。

6.2　複数コンテンツからのオントロジー構築結果

6.1 に以下の条件を加え、オントロジー構築を行った。

(1) 変換対象ドキュメントとして、e-Words の他に、IT 用語辞典 BINARY という別の技術用語サイトを追加した。
(2) 変換を行いたい用語に関する EXCEL ドキュメントを作り、EXCEL の指定された行に存在する用語についてオントロジーに追加する。
(3) IT 用語辞典 BINARY で別名記述のある用語については、その別名記述も別名プロパティとしてオントロジーに記述する。
(4) e-Words の関連用語項目を関連用語プロパティとしてオントロジーに記述する。
(5) IT 用語辞典 BINARY と e-Words で、同一用語名が存在した場合、ひとつのインスタンスとする。
(6) 関連用語についてはインスタンス生成と用語名の記述のみを行う。

以上の条件について XSLT 及び SPARQL の記述を行った結果、以下の結果を得られた。

(1) EXCEL で指定した用語について、二つの用語情報サイトに基づき、インスタンス生成及び別名や関連用語といったつながりをオントロジーとして持たせることができた。
(2) 一部の用語について、用語サイトの表記のずれにより「AND」「AND 演算」と別々の表記により、本来は同一のものであるべきものが別々のインスタンスとして記述されてしまった。
(3) 500 語を超える用語を指定した際、システムが不安定になり変換を行えなくなった。

6.3　既存オントロジー構築との比較

前節と同様のオントロジー構築条件の下、用語 100 語に対して protégé を用いてオントロジーを手動で構築し、本システムを用いてオントロジー構築を行った場合と比較した。

比較するにあたって、オントロジー構築に必要な処理を以下のように分類し、これらの処理に必要な時間を計測した。

(1) オントロジー構築準備

本システムにおける変換用 XSLT の生成、SPARQL のマッピング用 OWL 生成が該当する。既存オントロジーファイルについてはすでに用意されている物とみなすので、この処理には含めない。

(2) 用語インスタンス生成
用語クラスのインスタンスとして OWL 記述を行う処理。用語名の記述、関連用語のインスタンス生成もこの処理に含める。

(3) プロパティ記述
用語の別名、関連用語の関係性記述を行う処理。

(4) 記述内容確認
データの重複や生成された OWL ファイルの読み込み確認といったチェックを行う処理。この処理については本システムで構築したオントロジーについても protégé を用いて手動で行った。なお、変換対象ドキュメントに記述されている内容を全て確認する作業は今回の比較対象には含めない。

図 10、図 11 は本システムと protégé による手動構築にかかった時間と上記処理内容との対応関係である。

	処理内容	時間
(1)	XSLT, SPARQL 記述に要した時間 ※用語情報サイト用 2 ファイル。EXCEL ファイルについては 4.4.1 の物を再利用	120 分
	マッピング用 OWL の生成	10 分
(2) (3)	変換処理	2 分
(4)	データの重複チェック マッピング対応の確認	25 分
	合計	**147 分**

図 10　オントロジー自動構築に要した時間

	処理内容	時間
(1)	既存 OWL ファイルインポート処理	2 分
(2)	用語インスタンスの生成	80 分
	関連用語インスタンス生成	150 分
(3)	別名プロパティ記述	30 分
	関係用語プロパティ記述	60 分
(4)	記述内容確認	10 分
	合計	**332 分**

図 11　protégé を用いて手動構築した時間

上記の処理内容と構築時間は以下のような関係がある。

構築時間= (1) + { (2) + (3) + (4) } ×用語数

以上の結果より、本システムは 100 用語という比較的少ない用語量についても構築時間の短縮を行えた。用語数が増えた場合、手動構築は、(2) から (4) の比率が高いため、構築にかかる合計時間は用語数に比例して増える。本システムでは (2) から (4) の比率が低いため、用語数が増えた場合の処理時間の増加量は少ない。よって、変換対象のデータ量が増えるほど、本システムの利用効果が得られると言える。

6.4　課題

本システムを用いることによりオントロジー構築の一部を自動化することができたが、以下の課題が残った。

(1) XSLT 及び SPARQL 記述の複雑さ
XSLT については再利用が比較的行いやすいが、SPARQL については構築したいオントロジーによって毎回記述を行わないといけない。これらが要因となり、オントロジー構築の自動化にかかる負担が増えてしまった。

(2) 構築されたファイルの管理
本研究で用いられているシステムを複数回使用してオントロジーを構築した際、その回数文のインスタンスデータファイルが生成される。このように、オントロジー記述を追加した際のファイルの管理について改善する必要がある。

(3) 大容量データ変換時の不安定性
本研究のシステムは変換対象データ量が多いほどオントロジー構築時間短縮の恩恵を受けられるシステムである。しかし、本システムでは全てメモリ上でデータの変換及び構築を行っているため、メモリを超える容量を扱う際、動作が非常に不安定になり場合によっては変換不能に陥ることもある。

(4) HTML のリンク先情報の抽出について
本研究のシステムでは、対象ドキュメント

のリンク先について、同様に変換を行うといった処理は行えなかった。6.2 のように HTML 上の情報を取得する際には、リンク先の情報も RDF として含めたい場合がある。リンク先の情報についても同様に変換処理が行えれば、よりオントロジー構築における負荷が軽減されるのではないかと考えた。

7 おわりに

HTML、RDB、XML といったドキュメントフォーマットから自動的に OWL 記述に対応したインスタンス部分を生成し、オントロジー構築の一部を自動化することができた。

構築されたオントロジーは既存のオントロジー編集ソフト及び推論エンジンで読み取ることができ、手動でオントロジーの構築を行った場合よりも効率的にオントロジーの構築を行えた。

大容量データの扱い、及び変換に必要な XSLT、SPARQL の記述を軽減することが今後の課題である。

参考文献

[1] Resource Description Framework(RDF), W3C, http://www.w3.org/RDF/

[2] W3C "OWL Web Ontology Language" http://www.w3.org/TR/owl-features/, 2004

[3] Gleaning Resource Descriptions from Dialects of Languages (GRDDL),W3C http://www.w3.org/2004/01/rdxh/spec

[4] SPARQL Query Language for RDF,W3C http://www.w3.org/TR/rdf-sparql-query/

[5] Protégé3.3.1 http://protege.stanford.edu/, 2007

[6] SWOOP http://code.google.com/p/swoop/

[7] CyberNeko HTML Parser http://sourceforge.net/projects/nekohtml

[8] Office Open XML Formats http://www.ecma-international.org/memento/TC45.htm

[9] IT 用語辞典 e-Words http://e-words.jp/

[10] Jena – A Semantic Web Framework for Java http://jena.sourceforge.net/

[11] XML.com: Converting XML to RDF http://www.xml.com/pub/a/2004/09/01/tr.html

[12] ためしてわかるセマンティック WEB 次世代型データ活用術 伊藤 健太郎、濱崎 俊、佐藤 勇紀

[13] Rhizomik – ReDeFer XML2RDF http://rhizomik.net/redefer/

[14] RDF/XML Syntax Specification,W3C http://www.w3.org/TR/rdf-syntax-grammar/

[15] Pellet: The Open Source OWL DL Reasoner http://pellet.owldl.com/

[16] 高梨 勝敏, 佐藤 俊也, 原島 一郎: "Web コンテンツからのオントロジーの再構成方法の提案と試作", 人工知能学会論文誌, Vol. 20, No. 6, pp.417-425 (2005)

[17] Architecture of a Semantic XPath Processor. García, R and Celma, O.5th Knowledge Markup and Semantic Annotation Workshop, SemAnnot'05 CEUR Workshop Proceedings, Vol. 185, pp. 69-80, 2006