Wikipediaからの再利用可能な構造化データの抽出手法

森　竜也† 増田　英孝† 中川　裕志†† 清田　陽司†††

† 東京電機大学未来科学研究科情報メディア学専攻　〒101–8457 東京都千代田区神田錦町 2–2
†† 東京大学情報理工学系研究科数理情報学専攻　〒113–0033 東京都文京区本郷 7-3-1
††† 東京大学情報基盤センター学術情報研究部門　〒113–0033 東京都文京区本郷 7-3-1
E-mail: †mori@cdl.im.dendai.ac.jp, ††masuda@im.dendai.ac.jp, †††n3@dl.itc.u-tokyo.ac.jp,
††††kiyota@r.dl.itc.u-tokyo.ac.jp

あらまし　Web 上のフリー百科事典 Wikipedia は日本語版だけでも 70 万件を超える記事が掲載されている．しかし Wikipedia データの使われ方のほとんどは人が読むことであり，膨大な電子データである利点は活用しきれていない．これは Wikipedia データが半構造的なデータで，汎用的かつ統一的に解析する手法がないことに起因する．そこで本研究では再利用性を高めることを目的に，Wikipedia データを XML 文書化する手法を提案する．その結果， Wikipedia 特有の文法やプログラムに束縛されなくなり，容易に Wikipedia データを二次的に利用できる．

キーワード　Wikipedia, 構造化, XML, データの再利用, MediaWiki

Extraction of Reusable Structured Data from Wikipedia

Tatsuya MORI†, Hidetaka MASUDA†, Hiroshi NAKAGAWA††, and Yoji KIYOTA†††

† Graduate School of Science and Technology for Future Life, Tokyo Denki University
2–2 Kanda-Nishiki-cho, Chiyoda, Tokyo 101–8457 Japan
†† Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo, 113–0033 Japan
††† Information Technology Center, The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo, 113–0033 Japan
E-mail: †mori@cdl.im.dendai.ac.jp, ††masuda@im.dendai.ac.jp, †††n3@dl.itc.u-tokyo.ac.jp,
††††kiyota@r.dl.itc.u-tokyo.ac.jp

1. はじめに

Web 上のフリー百科事典 Wikipedia は，日本語版以外にも世界中の多数の言語版で運営されており，2011 年 1 月現在 250 を超える言語版が存在する．日本語版 Wikipedia は約 72 万件，英語版は約 350 万件の記事が掲載されている．全ての言語版を合計すると 1000 万件を超える．これらの記事の題材は基本的にユーザの自由に任せられているので，通常の百科事典には掲載されない傾向の話題の記事も Wikipedia には多数掲載されている．

Wikipedia データの使われ方のほとんどは，Wikipedia のサイトへ Web ブラウザでアクセスして人が記事を読むことである．Wikipedia は世界の Web サイトの中でもアクセス数が上位のサイトであり，非常に多くの人々に記事が読まれている．しかし Wikipedia のデータは，人が目で読む以外の用途にも有用そうな特徴を持っている．その点に着目して Wikipedia データを解析し，様々な情報を抽出したり，他のアプリケーションで利用したりする研究が行われている．

しかし Wikipedia データを利用する研究を行うに当たって障害となるのが，Wikipedia データを操作・解析する汎用的かつ統一的な手法が公式に用意されていない点である．そのため研究者は各々が解析プログラムを自前で作成することになっている．これは時間と労力を要する作業であり，研究への参入を難しくしている．

そこで本研究では Wikipedia データを構造化データである XML 文書へ変換する手法を提案する．その結果，研究者は自前の解析プログラムではなく，DOM や SAX などの既存の XML 文書解析 API を通して，Wikipedia データを操作することができるようになる．DOM や SAX は幅広いプログラミング言語でサポートされているのも利点である．

2. Wikipedia データの現状

本章では Wikipedia データの XML 文書化に当たって，Wikipedia やそのサービス基盤ソフトウェア MediaWiki について説明する．

2.1 Wikipedia データの特徴

Wikipedia のデータは，人が目で読む以外の用途にも有用そうな特徴を持っている．まず機械処理可能な形式でデータが公開されている．データは自由にダウンロードでき，一定のライセンスのもとに使用できる．

Wikipedia は客観的な視点による百科事典サイトを標榜しているため，文章の文体がある程度は統一されており，俗語や絵文字のような処理しにくい表現が極力排除されている．

またページを分類するためのカテゴリ情報などのメタデータの付与もサポートされており，通常の Web ページよりも機械処理可能な情報資源として扱いやすい．

また Wikipedia は MediaWiki という Wiki 管理システムで運営されているため，ページは全て MediaWiki で解析可能な文法に従って書かれている．MediaWiki が導入している文法を Wiki 記法といい，Wiki 記法で書かれたテキストを Wiki テキストという．Wiki 記法は編集者にとって修得が容易であることと，MediaWiki を通して HTML 文書へ変換することが求められるため，テキストの論理構造を記述する文法となっている．

2.2 Wikipedia データを使用した研究

情報源として有用な性質に着目し，Wikipedia データを解析して役立つ情報を抽出する研究が行われている．Auer らの研究 DBPedia [1] は Wikipedia データからセマンティック Web で利用可能なデータセットを生成している．これにより Wikipedia という 1 サイト内でのみ完結していたデータをセマンティック Web というより広い領域へ接続することを目指している．また Erdmann らの研究 [2] は異なる言語版の対応する記事同士がリンクで接続されている点に着目し，単語の翻訳辞書を自動構築する研究を行っている．坂井らの研究 [3] は Wikipedia のページの分類構造を，図書館の図書の分類構造と組み合わせることで相互に補完されたオントロジーを構築し，情報検索の利便性の向上を狙っている．

2.3 Wikipedia のページ構造

Wikipedia は MediaWiki 上の Web サイトであり，ページがサイトを構成する単位になっている．ページは編集者によって作成・編集される．

2.3.1 タイトル

全てのページはタイトルを持つ．MediaWiki 上のサイトに対して，以下のようなタイトルを含む URL から個別のページにアクセスすることができる．

```
http://ja.wikipedia.org/wiki/タイトル
```

タイトルは URL に組み込まれる仕様上，一意に識別することが可能であり，サイト内でユニークである．タイトルは以下のように名前空間文字列とページ名で構成されている．ページには役割によっていくつかの種別があり，タイトルの名前空間によって区別される．

```
名前空間:ページ名
```

通常の Web ページと異なり Wikipedia のページには必ずタイトルが付いているので，ページ内の情報の主題が常に明確である．

2.3.2 記　　事

記事は Wikipedia の通常のページである．1 件の記事に 1 つの主題を持つ．記事だけは他のページと異なり名前空間文字列がなく，ページ名のみでタイトルが構成される．

2.3.3 カテゴリ

カテゴリはページを分類するためのページである．名前空間文字列は Category である．ページ中に以下のようにタグ付けをしたテキストを記述することで，そのページをカテゴリに含めることができる．

```
[[Category:カテゴリ名]]
```

カテゴリ構造を構成するページを取得することで，タイトルの語の上下関係を抽出することができる．図 1 は「イヌ」からカテゴリ構造を追跡して得た語の上下関係の例である．

図 1　カテゴリ構造から得られる語の上下関係の例

2.3.4 テンプレート

テンプレートはプログラミング言語におけるメソッドやサブルーチンのような機能を実現するページである．名前空間文字列は Template である．テンプレートは次のように他のページから呼び出され，文字列を返す．

```
{{テンプレート名
|引数名 1 = 引数 1
|引数名 2 = 引数 2
    …
|引数名 n = 引数 n
}}
```

MediaWiki は呼び出し元のテンプレートの記述を，テンプレートが返した文字列へ置換する．テンプレートは任意の個数の引数を取ることができる．また四則演算や条件分岐，ルー

図 2 「基礎情報_会社」テンプレート

プなど簡単なプログラミング言語のような機能を備えている．Wikipedia ではテンプレートはインフォボックスと呼ばれる表を生成するために使われているのが目立つ．例えば「基礎情報_会社」は企業に関する情報をまとめるインフォボックスを生成するためのテンプレートであり，Web ブラウザ上では図 2 のように表示される．

表の項目はテンプレート呼び出し時の引数によって作られる．そのためテンプレートの引数を参照することで，記事の主題の事柄に関する属性名と値のペアを取得することができる．「基礎情報_会社」の引数の一部には以下のようなものがある．

- 社名
- 市場情報
- 本社所在地
- 設立
- 売上高
- 従業員数
- 主要株主

「基礎情報_会社」にも国，人物，生物，学問，娯楽など幅広い事柄に関するテンプレートが存在し，日本語版 Wikipedia には約 3 万 8 千種類のテンプレートが存在する．ただし全てのテンプレートがインフォボックスを生成するためのものとは限らない．

2.3.5 言語間リンク

ページに対して，他の言語版に存在する同一概念のページへの参照情報を設定する機能が言語間リンクである．言語間リンクを設定するには，ページに以下のような Wiki テキストを記述する．

```
[[言語プレフィックス:タイトル]]
```

言語プレフィックスとは各々の言語版を示す文字列である．例えば日本語版には ja，英語版には en という言語プレフィックスが与えられている．

言語間リンクで結ばれたページのタイトルは翻訳関係になっていることが知られている．日本語版の記事「イヌ」の言語間リンクを追跡することで，表 1 に示した翻訳語を含めて，計 169 言語での「イヌ」の翻訳語を取得することができる．

表 1 言語間リンクから取得できる「イヌ」の翻訳語

言語	翻訳語
英語	Dog
ドイツ語	Haushund
フランス語	Chien
ロシア語	Собака
中国語	犬

2.4 Wikipedia の配布データ

Wikipedia では SQL ダンプ形式と XML 形式の 2 種類のデータファイルが配布されている．配布データは数十種類あるが，大半は SQL ダンプ形式である．それらは MediaWiki が使うデータベースのテーブルをエクスポートしたもので数 KB から数十 MB の比較的小さなファイルである．XML 形式のデータファイルは各ページの情報や内容を収録したものである．日本語版の場合は jawiki-latest-pages-meta-current.xml というファイルが配布されており，この中に全ページの内容が収録されている．ファイルサイズは 2011 年 2 月の版で約 7GB である．

XML ファイルはリスト 1 のように mediawiki 要素をルートとして，siteinfo 部分と page 部分に分かれている．siteinfo 部分にはサイト情報や名前空間文字列の定義などが書かれている．page 部分は個別のページのタイトルや Wiki テキストなどの情報が書かれている．

リスト 1 XML ファイルの構造

```
<mediawiki>
  <siteinfo>
    サイト情報、名前空間定義
  </siteinfo>
  <page>1 ページ目のデータ </page>
  <page>2 ページ目のデータ</page>
    ……
  <page>n ページ目のデータ</page>
</mediawiki>
```

この XML ファイルは MediaWiki のインポート機能で取り込み，データベースへ復元することができる．ただしこのインポート機能は低速であり，日本語版の XML ファイルをインポートするのに数日かかる．これを解決するために xml2sql [4] や MWDumper と [5] いった専用の変換ツールで SQL ダンプへ変換し，データベースへ登録する方法が Wikipedia では公式に説明されている．

データベースに格納されるページデータの本文部分はリスト 2 に示すような生の Wiki テキストである．Wikipedia では数十種類のデータファイルが配布されているが，ページ本文はこのような生の Wiki テキストでのみ入手できる．テンプレートも本文中に記述されているため情報を取り出すには Wiki テキストを解析する必要があるが，Wiki テキストは MediaWiki でのみ使われる独自の形式のマークアップテキストであり，解析するには別のプログラムが必要である．

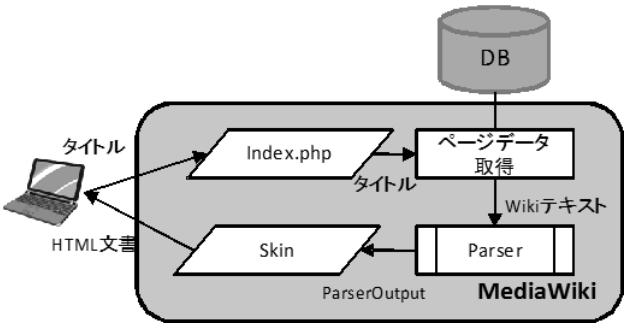

図 3　MediaWiki の処理の流れ

リスト 2 Wiki テキストの例

```
'''任天堂株式会社'''（にんてんどう、{{Lang-en-short|''Nintendo Co., Ltd.''}}）は、[[玩具]]・[[ゲーム]] を製造する [[日本]] の [[株式会社（日本）|株式会社]]。[[麻雀]]、[[囲碁]]、[[将棋]]、[[花札]] 用具のメーカーでもある。
{{lang|en|'''Nintendo'''}}（ニンテンドー）は、[[1990 年代]] まで主に [[北アメリカ|北米]] で [[テレビゲーム|ビデオゲーム]] 一般を指す [[俗語]] としても使われた。
== 会社概要 ==
本社は [[京都市]] に所在する。[[1889 年]][[9 月 23 日]] 創立。
```

当然 MediaWiki 自体が Wiki テキストの解析器を持っている．MediaWiki は図 3 のような流れの中で，Wiki テキスト解析器 Parser を使っている．最終的に Wiki テキストは HTML 文書へ変換され．それを一般のユーザは Web ブラウザを通して閲覧することになる．しかし HTML 文書へ変換されたデータは元の Wiki テキストが含んでいたデータが失われている。例えばテンプレート呼び出しの記述は戻り値に変換されるため，記事中のある文章に着目した時，元々そこにどのテンプレートが記述されていたか，あるいはテンプレートを介さず直接書かれた文章なのか，判断が付かない．

このように MediaWiki は Web ブラウザを通して Wiki テキストを編集し，HTML 文書として表示することに特化しており，蓄積したページデータを再利用する仕組みに乏しい．そこで本研究では Wikipedia データの再利用性を高めるための手法を提案する．

ひとつは XML 形式のデータファイルから，特定の情報を抽出するツールを開発することである．こちらの手法は分散処理プラットフォーム Hadoop を利用した手法を以前に発表した [6]．もうひとつの手法は Parser の出力結果を HTML 文書ではなく，オブジェクトの内容を永続化したような形式の XML 文書へ変換することである．本稿では後者の手法について述べる．

3. Wikipedia データの構造化

本章では Parser の出力結果を XML 文書へ変換する手法を説明する．

3.1 Wikipedia データの構造化の利点

Wiki テキストの文法は基本的には平易だが，テンプレートやテーブルの記述などで複雑な書式になる場合がある．MediaWiki は明確な仕様がないため，正確に動作することが保証されている Wiki テキスト解析器は純正の Parser だけである．また Wikipedia で使われている MediaWiki には多数の機能拡張プログラムが組み込まれており，Wikipedia の配布データには機能拡張向けの文法で書かれたテキストが含まれているため，サードパーティ製の Wiki テキスト解析器が必ず正しく Wikipedia データを解析することは期待できない．そのため Wiki テキスト解析器には MediaWiki 純正の Paser を使うことになる．しかし MediaWiki 純正の Parser には 2 点の問題がある．

ひとつは MediaWiki は PHP での実装のみが開発されており，Parser の解析結果も PHP プログラム上のオブジェクトである点である．このため Wikipedia データは MediaWiki というソフトウェアと強く結合しており，他のプログラムでの利用することが困難である．

もうひとつの問題点は 2.4 節で述べた，テンプレートの引数が解析結果から失われてしまう点である．

これらの問題点を解決するために，テンプレートの処理を行っている部分の MediaWiki のコードを書き換え，テンプレートの名前と引数名・値を解析結果に埋め込むように変更した．また改良した MediaWiki の解析結果を XML 文書として出力するプログラムを作成した．

その結果 DOM や SAX などの既存の XML 解析 API を通した操作のみでテンプレートを含む Wikipedia データにアクセス可能となった．さらに元の Wiki テキストでは記事本文にカテゴリや言語間リンクなどのメタデータが埋め込まれていたが，XML 文書化する際に本文とは別の要素にメタデータを記述したので，より再利用性が改善できた．

3.2 Wikipedia データの構造化の実装

図 3 の Parser は MediaWiki を構成するクラスであり，生の Wiki テキストを解析し，ParserOutput というクラスのオブジェクトへ変換する機能を持っている．ParserOutput は HTML 文書へ変換された本文のほか，カテゴリや言語間リンクなどのメタデータを保持する．Skin によって ParserOutput を最終的な出力結果へ加工する．

提案手法を実装するに当たって，まず MediaWiki を導入する必要がある．しかし 3.1 節で述べたように Wikipedia の MediaWiki には多数の機能拡張が組み込まれているため，標準の MediaWiki を導入しても Wikipedia の配布データを正しく解析することができない．そこで wmf4 [7] という MediaWiki のディストリビューションを導入する．wmf4 は Wikipedia で使用されているのと同様の機能拡張が予め組み込まれた MediaWiki である．

3.2.1 テンプレート引数の解析結果への埋め込み

MediaWiki ではテンプレートの処理を Preprocessor_DOM および Parser クラスで行っている．中でも重要な部分はリスト 3 およびリスト 4 に示したコードである．これらのコードに

よってテンプレートが返す文字列が再帰的に生成されている．

これらのコードをリスト 5 およびリスト 6 のように変更することで通常のテンプレート解析を回避し，リスト 9 に示す形式の XML 要素を返すようになる．テンプレート名や引数名が要素の属性として記述されているため，XML 文書に対する操作のみでテンプレート情報にアクセスすることができる．

リスト 3 Preprocessor_DOM のコード (936〜967 行)

```
$ret = $this->parser->braceSubstitution( $params, $this );
if ( isset( $ret['object'] ) ) {
    $newIterator = $ret['object'];
} else {
    $out .= $ret['text'];
}
```

リスト 4 Parser のコード (3040〜3045 行)

```
if ( isset( $this->mTplExpandCache[$titleText] ) ) {
    $text = $this->mTplExpandCache[$titleText];
} else {
    $text = $newFrame->expand( $text );
    $this->mTplExpandCache[$titleText] = $text;
}
```

リスト 5 Preprocessor_DOM の改良コード

```
$ret = $this->parser->braceSubstitution( $params, $this );
if ( isset( $ret['object'] ) ) {
    $newIterator = $ret['object'];
} else {
    $str_title = trim($this->expand($params['title']));
    $obj_title = Title::newFromText($str_title, NS_TEMPLATE);
    $template = "";
    if (isset($obj_title)) {
        $template .= "<div class=\"template\" title=\"".$obj_title->getDBKey()."\">";
        $template .= $ret['text'];
        $template .= "</div>";
    }
    $out .= $template;
}
```

リスト 6 Parser の改良コード

```
if ( isset( $this->mTplExpandCache[$titleText] ) ) {
    $text = $this->mTplExpandCache[$titleText];
} else {
    $text = "";
    foreach ($newFrame->getArguments() as $key => $value) {
        $text .= "<div class=\"argument\" title=\"$key\">";
        $text .= $value;
        $text .= "</div>";
    }
}
```

3.2.2 解析結果の XML 文書化

Wiki テキストを XML 文書化するには、Parser を一連の流れから切り離し，自作のプログラムから呼び出せるようにする．リスト 7 のように commandLine.inc と Parser.php を読み込むことで，Parser を MediaWiki 本来の処理の流れから切り離して呼び出すことができる．commandLine.inc はメンテナンス用のプログラムを呼び出す際に使うファイルで，Parser のような MediaWiki に組み込まれているクラスを外部利用できるようになる．Parser に対して，本文，タイトルを示す Title，解析時のオプションを示す ParserOptions を与えると，ParserOutput が得られる．

ParserOutput は解析結果の要素に対するアクセサメソッドがある．例えばカテゴリは getCategories()，言語間リンクは getExternalLinks() で取得できる．これらのメソッドを利用して ParserOutput オブジェクトからデータを取り出し，リスト 8 のような XML 文書を出力する．

リスト 7 Parser の呼び出し

```
<?php
require_once ('maintenance/commandLine.inc');
require_once ('includes/parser/Parser.php');

$parser = new Parser();
$text = "Wiki テキスト";
$title = Title::newFromText('Title');
$options = new ParserOptions();
$output = $parser->parse($text, $title, $options);
?>
```

リスト 8 変換した XML ファイルの構造

```
<?xml version="1.0" encoding="UTF-8"?>
<page>
<title>タイトル</title>
<body>
<p>本文</p>
</body>
<categories>
  <category>カテゴリ 1</category>
  <category>カテゴリ 2</category>
  <category>カテゴリ 3</category>
</categories>
<interlangs>
  <interlang lang="de">ドイツ語版</interlang>
  <interlang lang="en">英語版</interlang>
  <interlang lang="fr">フランス語版</interlang>
</interlangs>
</page>
```

リスト 9 テンプレートを表現する XML 要素

```
<div class="template" title="テンプレート名">
  <div class="argument" title="引数名 1">
    引数 1
  </div>
  <div class="argument" title="引数名 2">
    引数 2.
  </div>
    …
  <div class="argument" title="引数名 n">
    引数 n
  </div>
</div>
```

4. XML 文書化 Wikipedia データの利用例

提案手法で XML 文書化した Wikipedia データの利用例として，テンプレートの情報から条件に適合するページを発見する処理を行う．

4.1 東証一部上場企業

Wikipedia では企業に関する情報をまとめた表を生成するためのテンプレート「基礎情報_会社」が使われている．株式の上場情報は「市場情報」という名前の引数に書かれている．XML 文書化された Wikipedia データ内の，市場情報」の要素を参照する XPath の評価式は以下のようになる．

```
/pages/page/body/div[@class="template"]
[@title="基礎情報_会社"]/div[@class="
argument"][@title="市場情報"]
```

この評価式で取得したテキストが「東証 1 部」あるいは「東証一部」に部分一致すれば，その会社は東証一部上場企業であると判定した．

この手順で XML 文書化した Wikipedia データを検索するプログラムを Java で作成した．その結果を示したものが表 2 である．東京証券取引所のサイトに掲載されている一部上場企業数よりも Wikipedia から取得できた数のほうが多いが，これは上場廃止になった企業の情報が更新されていないことがあったためである．

4.2 売上高が多く上場していない企業

株の市場情報と売上高の引数から，売上高が多いが株が上場していない企業を探した．売上高は以下の XPath の評価式で参照できる．

```
/pages/page/body/div[@class="template"]
[@title="基礎情報_会社"]/div[@class="
argument"][@title="売上高"]
```

市場情報と売上高が以下の 2 つの条件に合致した場合，その

企業は売上高が多く上場していないと判定した．

- 市場情報が空白，あるいは「未上場」「非上場」のいずれかに部分一致する
- 売上高が「兆」に部分一致する

この手順で XML 文書化した Wikipedia データを検索するプログラムを Java で作成した．結果として条件に合致した企業が 26 社見つかった．上場していない理由を調査したところ，以下の 3 つの理由に分類できた．

（1） 単純に上場していない
（2） 親会社や持株会社がある
（3） 株式会社ではない

この分類に従って見つかった企業をまとめたものが表 3 である．

表 2　東証 1 部上場企業

情報源	数
Wikipedia	1,711
東証サイト	1,680

表 3　売上高が多く上場していない会社

分類	会社名	売上高 [円]
1	日本郵政	10 兆 979 億
	マルハン	2 兆 559 億
	サントリー	1 兆 5,129 億
	メディセオ	1 兆 3,506 億
	JTB	1 兆 2,760 億
	矢崎総業	1 兆 550 億円
	リクルート	1 兆 839 億
	竹中工務店	1 兆 394 億
2	メタルワン	3 兆 3,316 億
	セブン-イレブン	2 兆 4,987 億
	ソニーイーエムシーエス	2 兆 251 億
	JFE 商事	1 兆 8,548 億
	東日本電信電話	1 兆 9,286 億
	西日本電信電話	1 兆 7,808 億
	イトーヨーカ堂	1 兆 6,778 億
	野村證券	1 兆 5,937 億
	ソフトバンクモバイル	1 兆 5,791 億
	アルフレッサ	1 兆 4,451 億
	トヨタファイナンシャルサービス	1 兆 4,106 億
	日本アクセス	1 兆 3,425 億
	日本サムスン	1 兆 2,062 億
	富士ゼロックス	1 兆 1,633 億
	NTT コミュニケーションズ	1 兆 792 億
	パナソニック コンシューマーマーケティング	1 兆 239 億
3	日本生命保険	6 兆 6,898 億
	住友生命保険	3 兆 637 億

このようにテンプレートを含む Wikipedia データを XML 文書化することで，XML 文書の操作と簡単な文字列のマッチングだけで Wikipedia データを扱えるようになった．

5. おわりに

本稿では Wikipedia データの再利用性を高めるアプローチとして，Wiki テキストの解析結果を XML 文書化する手法を提案した．その利点をまとめると以下のようになる．

- Wikipedia データを，Wiki テキストという特定のソフトウェアと強く結合したデータ形式から，XML 文書というオープンかつ構造的なデータ形式へ変換した．その結果，PHP 以外のプログラミング言語でも Wikipedia データを扱え，その際には XPath 等の既存の API が利用できる．
- 従来の MediaWiki では扱えなかったテンプレートの引数の情報を XML 文書に埋め込んだことで，属性名と値のペアを取得できるようになった．

今後の展開としては，Wikipedia データを利用したい第三者向けに XML 文書化した Wikipedia データを公開することを考えている．また現在はテンプレートの引数を解析結果に受けこむために MediaWiki のコードを直接書き換えているが，今後のアップデート等に対応できない恐れがあるので，機能拡張の一部として同様の機能を実現できないか検討したい．

文　献

[1] Soren Auer, Chris Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, pp. 715–728, 2007.

[2] Maike Erdmann, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Extraction of bilingual terminology from a multilingual web-based encyclopedia. *Journal of Information Processing*, Vol. 16, July, pp. 68–79, July 2008.

[3] 坂井哲, 増田英孝, 清田陽司, 中川裕志. Wikipedia と図書館情報資源による調べ方自動提示システム. 情報処理学会 第 72 回全国大会 講演論文集 (3K-5), 2010.

[4] xml2sql. http://meta.wikimedia.org/wiki/Xml2sql.

[5] MWDumper. http://www.mediawiki.org/wiki/MWDumper.

[6] 増田英孝. 言語間差異を活用した web 情報資源へのアクセスシステムに関する研究. 学際大規模情報基盤共同利用・今日研究拠点第 2 回シンポジム (3K-5), 2011.

[7] 1.16wmf4. http://svn.wikimedia.org/svnroot/mediawiki/branches/wmf/1.16wmf4.