

KawaWiki: Semantic Wiki による RDF 生成

KawaWiki: Generating RDF by Semantic Wiki

河本健作

Kensaku KAWAMOTO

北村泰彦

Yasuhiko KITAMURA

関西学院大学理工学部

School of Science and Technology, Kwansai Gakuin University

1 はじめに

Semantic Web とは、W3C の Tim Berners-Lee 氏により提唱され、次世代 Web システムの標準として期待されている枠組みである [1,2]。Web 情報に意味タグを付け加えることで、コンピュータによる自動的な情報の収集や検索を正確かつ効率的に処理することが可能になる。従来の Semantic Web システムは一般ユーザが使用することが困難であるために、学術機関や企業などによる Web サービスを目的としたものが主流であった。しかし今後は一般ユーザでも簡単に Semantic Web 情報を生成し、発信できるような仕組みが必要になる。

WikiWikiWeb (以後 Wiki) とは 1990 年代に Ward Cunningham 氏によって考え出された Web ページ共同制作のためのシステムである [3]。不特定多数の人がブラウザを介して Wiki ページと呼ばれる Web ページを自由に作成したり書き換えたりすることで、Web サイトを共同で制作することができる。Semantic Wiki は Semantic Web 情報の生成・発信に Wiki の枠組みを利用したものである [4]。既存の Semantic Wiki の多くは、Semantic Web 情報の生成のために RDF/XML 構文や構文記述言語を記述する必要があるため、一般ユーザ向けではない。

Wiki は不特定多数の編集者が存在し、任意の時間に自由な編集活動を行うことができるという「オープンな編集」というコンセプトを持つ。これは、Wiki 上の情報が削除されたり、Wiki 上に存在する情報の信頼性が低下したり、Wiki ページ内の情報に矛盾が発生する可能性があることを意味する。このような枠組みを、情報を厳密に取り扱う Semantic Web 情報の生成・発信のために利用する場合、その情報の整合性をいかに保つかという点が重要になる。

本論文では、テンプレートシステムを用いることで一般ユーザでも簡単な操作で RDF の生成ができる KawaWiki について紹介する。また、KawaWiki 上に存在する Semantic Web 情報の整合性を保つ方法を提案する。

2 KawaWiki

KawaWiki は一般ユーザでも簡単な操作で RDF データの生成を可能にするためのテンプレートシステムと、RDF データの整合性を保つ仕組みを持つ。図 1 に KawaWiki の概要を示す。KawaWiki はテンプレートを元に RDF データと HTML ページを生成する。また、RDF データの整合性を保つためにテンプレートと RDF スキーマとの整合性を保っている。

この章では、KawaWiki のテンプレートシステムと整

合性維持の仕組みについて述べる。

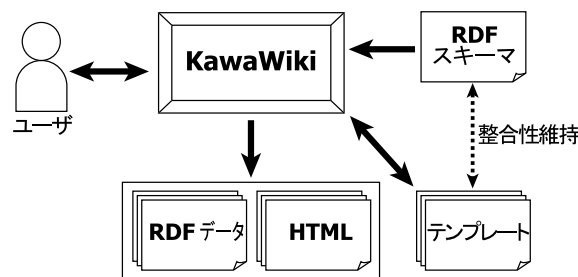


図 1 KawaWiki の概要

2.1 テンプレートシステム

KawaWiki では、簡単な操作で Semantic Web 情報の作成を可能にするために、テンプレートシステムを用いている。テンプレートシステムでは専門知識を持ったユーザが記述したテンプレートに対して、一般ユーザが値やリソースなどの具体的なデータを穴埋めするだけで RDF データと HTML ページ (Wiki ページ) が生成される。一般ユーザによるデータの inputs は、複雑な RDF 構文を覚える必要がないように、Web 上のアンケートや個人情報の入力画面などで見られるような、属性名に対応する属性値をフォームから選択したり、直接入力する方式で行う (図 2)。これにより、専門知識を持ったユーザと持っていない一般ユーザの役割分担を行うことができ、多くのユーザが積極的に Semantic Wiki による情報発信が可能になる。

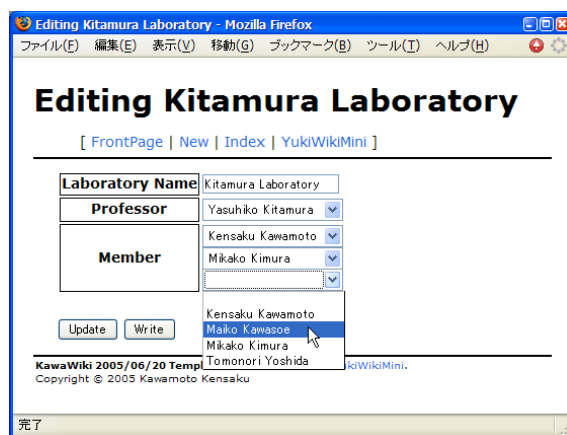


図 2 編集フォームの例

テンプレートは RDF/XML 構文で記述する (図 3)。テンプレート記述専用言語などを使用する必要が無いため、Semantic Web 開発者にとっては記述は容易であ

る。テンプレートは名前空間の定義、変数の宣言、テンプレート本体の三部から構成される。名前空間の定義は、一般的な RDF 文書に記述されるものと同様である。変数の宣言は、値入力フォームの表示、ユーザがフォームに入力した値をテンプレート本体に埋め込むための仲介、値の取り得る範囲や型などの制約に使用される。

図3は、研究室の Wiki ページ (RDF データ) を作成するためのテンプレートを示している。変数は研究室名 (Laboratory Name) を示す lab_name (10~14 行目)、担当教授 (Professor) を示す prof (15~19 行目)、配属学生 (Member) を示す student (20~24 行目) の3つが宣言されている。このテンプレートから生成されたデータ入力フォームが図2に示してある。変数 lab_name の取り得る範囲は文字列とされており (13 行目)、テンプレート本体ではリテラルノードとして扱われている (29 行目)。変数 prof は取り得る範囲が professor クラス (すなわち、professor テンプレートを利用して作成された Wiki ページの集合) となっており (18 行目)、テンプレート本体ではリソース参照されている (30 行目)。同様に変数 student は student クラスで (23 行目)、テンプレート本体ではコンテナモデル rdf:Bag でグループ化されている (31 行目)。

```

1: <?xml version="1.0"?>
2: <rdf:RDF
3:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4:   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5:   xmlns:template="http://localhost/template.rdf#"
6:   xmlns:lab="http://localhost/lab.rdf#"
7: >
8:   <rdf:Description rdf:ID="template">
9:     <template:define rdf:parseType="Resource">
10:      <template:variable rdf:parseType="Resource">
11:        <template:varname>lab_name</template:varname>
12:        <template:varLabel>Laboratory Name</template:varLabel>
13:        <template:vrange>.*</template:vrange>
14:      </template:variable>
15:      <template:variable rdf:parseType="Resource">
16:        <template:varname>prof</template:varname>
17:        <template:varLabel>Professor Name</template:varLabel>
18:        <template:vrange>professor</template:vrange>
19:      </template:variable>
20:      <template:variable rdf:parseType="Resource">
21:        <template:varname>student</template:varname>
22:        <template:varLabel>Member</template:varLabel>
23:        <template:vrange>student</template:vrange>
24:      </template:variable>
25:    </template:define>
26:    <template:body>
27:      <!CDATA[
28:      <lab:Laboratory>
29:        <lab:Lab-Name rdf:parseType="Literal">$lab_name</lab:Lab-Name>
30:        <lab:Prof rdf:resource="$prof"/>
31:        <lab:Belong>&Bag{$student}</lab:Belong>
32:      </lab:Laboratory>
33:    ]>
34:    </template:body>
35:  </rdf:Description>
36: </rdf:RDF>

```

図3 テンプレートの例

2.2 KawaWiki における整合性検証

RDF は、Semantic Web 情報を記述するための仕様の一つである。RDF で使用することができる語彙の定義を行うために RDF スキーマが用いられる。RDF スキーマでは、RDF で用いるプロパティの定義、主語と目的語の値の範囲 (ドメイン, レンジ) の定義、リソースをグループ化するクラスの定義などを行う。RDF スキーマをあらかじめ記述し、定義しておくことで、整合性の検証を行うことができる。

研究室についての RDF データを作成した場合の整合性検証の例を示す。あらかじめ RDF スキーマで“研究室に所属する人物 (研究室メンバー) は教授と学生である”と記述されていると仮定する。ある研究室についての RDF データのメンバーに教授と学生を追加した場合、RDF スキーマとの整合性検証で矛盾は発生しない。だ

が、メンバーに事務員を追加した場合、教授と学生以外である事務員が追加されたために矛盾が発生する。

KawaWiki 上には Wiki 全体の整合性を保つための RDF スキーマが存在する。テンプレートの作成・更新時にスキーマとの整合性の検証が行われ、矛盾が発生した場合は矛盾箇所をエラー表示し、解消されるまではそのテンプレートの作成・更新の操作ができなくなる。整合性の検証には OWL,RDFS 検証ツールである vOwLidator [5] を使用する。

テンプレートと Wiki ページの具体的な処理について説明する。テンプレート内の変数が変更、削除、追加された場合はテンプレートと Wiki ページとの間に矛盾が発生し、テンプレートの制約に合わせるために、Wiki ページを編集することが必要になる。KawaWiki では矛盾の発生した Wiki ページを目立つ色などで表示することで、ユーザにデータの更新を促す。テンプレート自体が削除された場合も矛盾が発生する。この場合は当該のテンプレートを使用している Wiki ページだけでなく、その Wiki ページにリソース参照を行っている Wiki ページについても矛盾が発生することになるため、同様にユーザにデータの更新を促す。

3 おわりに

本論文では、テンプレートシステムを用いた Semantic Wiki である KawaWiki の紹介を行った。テンプレートシステムを用いることで、専門知識を持っていない一般ユーザでも Semantic Web 情報の生成・発信が可能になる。また、RDF スキーマを利用することで KawaWiki 上に存在する Semantic Web 情報の整合性を保つ方法の提案を行った。

現在の KawaWiki では、RDF データ間 (Wiki ページ間) のリソース参照は単独サイトの閉じた KawaWiki 内でのしか行うことができない。今後の課題として、これを InterWiki のような仕組みを用いることで、他サイトの KawaWiki 上の Wiki ページとのリソース参照を可能にするということが考えられる。

参考文献

- [1] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, May 17, 2001.
- [2] 神崎正英. セマンティック・ウェブのための RDF/OWL 入門. 森北出版, 2005.
- [3] Bo Leuf and Ward Cunningham. *Wiki Way*. ソフトバンクパブリッシング, 2002.
- [4] Roberto Tazzoli, Paolo Castagna, and Stefano Emilio Campanini. Towards a Semantic WikiWiki-Web. *Poster Track, 3rd International Semantic Web Conference* pp. 7-11, November 2004.
- [5] BBN. vOwLidator. <http://owl.bbn.com/validator/>