

共生コンピューティングと Semantic Wiki

北村 泰彦[†] 河本 健作^{††} 川崎 琢磨[†]

[†] 関西学院大学 理工学部 〒 669-1337 三田市学園 2-1

^{††} 村田製作所 〒 520-2393 野洲市大篠原 2288

E-mail: †{ykitamura,taku925}@kwansei.ac.jp, ††k_kawamoto@murata.co.jp

あらまし 情報通信技術は現代社会に深く浸透しているが、利便性の一方で、その障害によって大きな社会問題を引き起こすこともある。このような問題に対処する一つの枠組は、人工物と人間利用者がうまく共存できる共生コンピューティングであると考えている。本稿では共生コンピューティングへの一つのアプローチとして、人工物であるエージェントと人間利用者が協調して、Web コンテンツを作り上げる Semantic Wiki を取り上げる。Semantic Wiki のプロトタイプとして開発した KawaWiki と、コンテンツの整合性維持の機能について議論する。

キーワード Semantic Wiki, Semantic Web, エージェント, 整合性維持

Symbiotic Computing and Semantic Wiki

Yasuhiko KITAMURA[†], Kensaku KAWAMOTO^{††}, and Takuma KAWASAKI[†]

[†] School of Science and Technology, Kwansei Gakuin University 2-1 Gakuen, Sanda 669-1337, JAPAN

^{††} Murata Manufacturing 2288 Ohshinohara, Yasu 520-2393, JAPAN

E-mail: †{ykitamura,taku925}@kwansei.ac.jp, ††k_kawamoto@murata.co.jp

Abstract Information and communication technologies deeply penetrate into our society. While they make our life convenient, they have raised a number of problems owing to their troubles. We believe the concept of “symbiotic computing,” where artifacts and human users coexist collaboratively, can ease the problems. We discuss Semantic Wiki in which human users and information agents collaborate to create Web contents and introduce KawaWiki, which is a prototype of Semantic Wiki, and its functions to maintain the consistency of the Wiki contents.

Key words Semantic Wiki, Semantic Web, Agent, Consistency Maintenance

1. はじめに

現代社会において情報通信技術は、われわれの生活から切り離すことができないくらい、重要な存在になっている。社会のすみずみまで情報通信ネットワークが張り巡らされ、様々な情報交換や取引がその上で行われており、いったんネットワークに不具合が生じると大きな社会問題が生じる。すでに過去の話題となってしまったが一時「2000年問題」が一つの社会現象を引き起こした。当時のコンピュータソフトウェアには下二ケタで年を表現していたものが多くあり、2000年になるとコンピュータが1900年と誤解して、誤動作し、社会は大混乱に陥ると危惧された。事前に様々な対策が施されたので、実際にはそれほど大きな混乱には至らなかったが、現代社会と情報技術がどれほど密接に関連し合っているかを示す事例の一つであるといえる。

情報システムのトラブルにより大きな経済的損失を引き起こした事例としては2005年12月の「ジェイコム株大量誤発注事件」があげられる。ある証券会社の担当者が株の取引において、コンピュータ端末に「61万円1株売り」とすべきところを「1円61万株売り」と入力してしまった。誤りに気づいた担当者が1分25秒後に3回にわたって取り消し操作をしたが、東京証券取引所のシステムは認識をしなかった。このような通常考えられないような数の売り注文に対して株価は急落し、大量の買い注文を出す投資家や、株価の急落にあわてて所有する株を放出する個人投資家が現れるなど、株式市場は大混乱に陥った。この結果、事件の発端となった証券会社は400億円を超える損失をこうむることになった。この事件は担当者の人為ミスが発端であるが、その後のフェールセーフ機能がほとんど働かなかったことが大きな損失へとつながった。現在のネットワークシステムでは一つのミスがネットワークを介して伝搬し、巨

大な損失を生み出す可能性を含んでいる。

このような問題に対処する一つの手法はインタフェースの改善である。オペレータが誤ったデータ入力を行わないように、水際で食い止めるという方法である。しかし、この手法は十分ではない。先述の「ジェイコム事件」においても、システムは担当者の誤発注に対して、内容が異常であるとの警告を発していたが、警告はたびたび表示されることがあり、担当者はそれを無視してしまった。したがって、誤ったデータがネットワークに流出してからの対策も必要になる。

このようなネットワークシステムにおける安全性を保障する一つのアプローチとしてエージェントと人間が共存する「共生コンピューティング」[1] が考えられる。現在のインターネットをベースとするシステムでは多くの人間が相互作用を行い、情報交換や取引を行っている。そこにエージェントを導入し、ネットワーク上に分散して存在するエージェントがデータを監視し、問題がある場合は警告を発して人間に知らせるというアプローチである。問題の発見には単一のエージェントだけでは不可能な場合もある。その際には複数のエージェントが協力して問題を発見することも考えられる。本稿は「共生コンピューティング」に対する一つのアプローチとして、人間とエージェントが協調して、整合性のある Wiki コンテンツを構築する Semantic Wiki について議論する。本稿では以下、2 章においてエージェントと人間が協調可能な Wiki とその利点について述べる。3 章ではそのような Wiki の実例として、われわれが開発している KawaWiki について述べる。4 章では KawaWiki が提供する整合性検証機能について述べる。5 章で Semantic Wiki に関する関連研究を述べ、6 章でまとめと今後の課題について述べる。

2. エージェントと人間が協調する Wiki

Wiki とは Ward Cunningham 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 [2] で、ネットワーク上の不特定多数の人間同士がブラウザを介して協調しながら編集活動を行うことができる。Wikipedia^(注1)に代表されるように、多数の利用者が協力して、特定のトピックについての新たな Wiki ページを作成したり、他の利用者が作成した Wiki ページの一部を編集したり、誤りを修正したりすることでコンテンツが構築されていく。その一方で、Wiki コンテンツには構文的、意味的な誤りが訂正されないまま放置されるという問題も生じている。エージェントが Wiki 上のコンテンツを理解して、その編集活動に参加することができれば、Wiki コンテンツの整合性の検証、不整合性の警告、さらには自動修正といったことが可能になるであろう。

具体的な事例として研究室活動を支援する Wiki を想定しよう。例えば、メンバーの個人的予定と、ゼミや打ち合わせなど

の研究室の予定が衝突しているかエージェントは検証を行うことができる。もし衝突していれば、個人的な予定を変更するようにメンバーに警告を行う。また逆に、各メンバーの予定を参照して、次のゼミをいつ設定すればよいかをエージェント側から提案することも考えられる。

また場合によっては、エージェントが Wiki コンテンツを自動的に修正することも可能である。例えば、研究室 Wiki においてプロジェクトの Wiki ページとメンバーの Wiki ページがあるとする。メンバーはいずれかのプロジェクトに所属し、そのプロジェクトはメンバーページに記述され、プロジェクトページには所属するメンバーのリストが記述されているとする。このような研究室 Wiki において、あるメンバーが所属するプロジェクトを変更し、そのことをメンバーページに記述した場合、この内容は新旧のプロジェクトページにも反映させる必要がある。従来の Wiki ではこのような更新は手作業で行う必要があり、それを怠ると Wiki コンテンツに矛盾が生じるようになる。そこでエージェントが編集活動に参加することによって、あるメンバーが自分の所属するプロジェクトを変更するとエージェントが自動的にそれを検知し、プロジェクトページのメンバー一覧を自動的に更新する。

3. KawaWiki

人間とエージェントが Wiki 上で協調するためには、Wiki コンテンツをエージェントが理解する必要がある。従来の Wiki では主に自然言語によってコンテンツが記述されているため、エージェントがそれを理解することは困難である。そこでわれわれは、Semantic Web 情報を付加することで、エージェントが Wiki コンテンツの一部を理解可能にし、人間の利用者と相互作用を行うことが可能な Semantic Wiki システム KawaWiki を開発した [5] ~ [7]。

3.1 Semantic Tag

Semantic Web は、Web の開発者でもある Tim Berners-Lee によって提唱されている次世代 Web システムの標準として期待されている枠組みで [3], [4], RDF や OWL といった記述言語が提案されている。Web 情報に意味タグを付け加えることで、エージェント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。

KawaWiki では Wiki コンテンツに Semantic Tag を挿入することで、容易に RDF/XML データを出力できる。図 1 に示すように、従来の Wiki と同様のテキストフィールドインタフェースから、通常のテキストに加えて、[[属性名:値]] という Semantic Tag を挿入できる。編集が終わると、図 2 に示す、エージェントのための RDF/XML ページと、図 3 に示す、人間のための Wiki ページが生成される。

Semantic Tag と RDF/XML ページの関係は図 4 に示すような KawaWiki テンプレートに記述されている。これにより、

(注1): <http://www.wikipedia.org/>



図 1 Semantic Tag の挿入

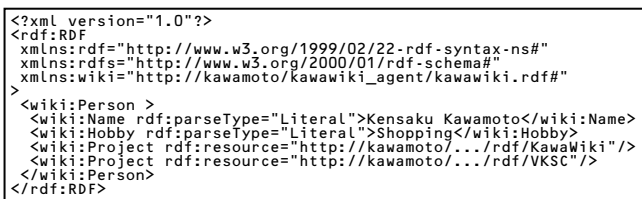


図 2 生成された RDF/XML

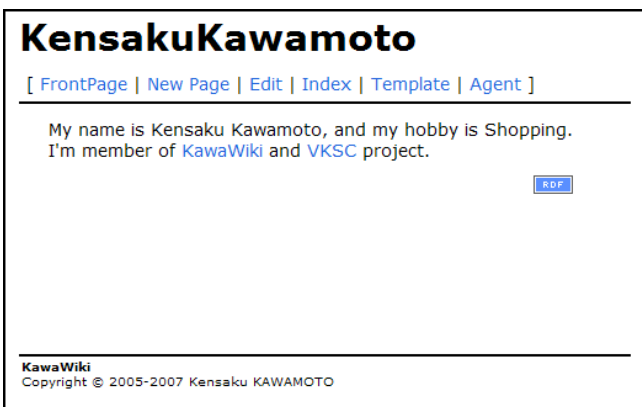


図 3 生成された Wiki ページ

“Kensaku Kawamoto” が人名 (Name) であり, “Shopping” が趣味 (Hobby) であり, “KawaWiki” や “VKSC” がプロジェクト名 (Project) であることをエージェントに理解させることができる。

KawaWiki テンプレートは XML 宣言 (1 行目), 名前空間定義 (2~6 行目), RDF 本文 (7~13 行目) から構成される。RDF/XML 文書中の属性の定義は, <%属性名\\値の取り得る範囲%>という形式で, 属性名とその値の取り得る範囲を記述する。

図 4 は, 個人プロフィールの RDF データを生成するための KawaWiki テンプレートの例である。8 行目に氏名を示す Name, 9 行目に趣味を示す Hobby, 11 行目に自分の所属するプロジェクトを示す Project の 3 つの属性が宣言されている。

この KawaWiki テンプレートでは, 自分の所属するプロジェクトを複数持つことができるような記述になっている。10~12

行目で, 属性 Project は&list(...); という繰り返し記述を示すタグに括られており, これは属性 Project が複数の値をもつことができることを示している。

3.2 KawaWiki クエリ

KawaWiki ではエージェントに RDF/XML で記述された属性にアクセスする手段として KawaWiki クエリを提供する。KawaWiki クエリは, Semantic Web クエリ言語である SPARQL を利用している。

図 5 では KawaWiki クエリとその結果が取得される様子を示している。SPARQL クエリは SELECT 句や WHERE 句などを持ち, SQL に似た構造をしている。このクエリは Person テンプレートから生成された Wiki ページで, wiki:Project が “KawaWiki” となっているページの wiki:Name の値を?name 変数に代入せよという意味になり, KawaWiki プロジェクトに所属しているメンバーの一覧が取得できる。

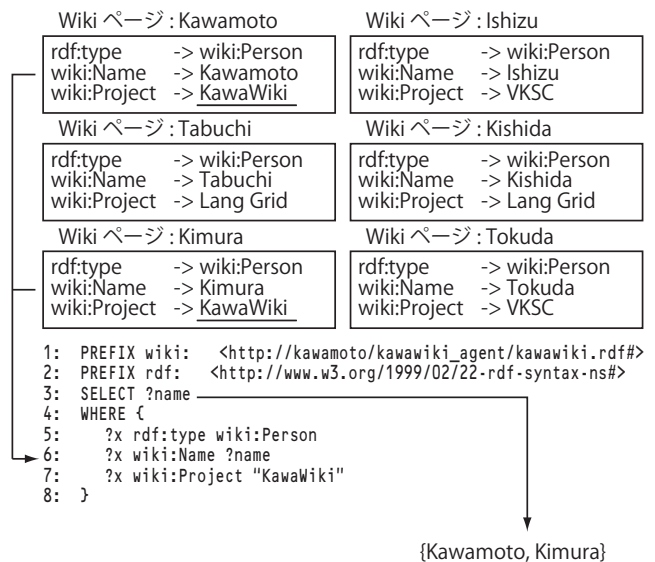


図 5 KawaWiki クエリとその結果

3.3 エージェント記述言語

エージェント記述言語は PHP をベースとしており, PHP で表現できる機能に併せて, KawaWiki クエリの呼び出しや Wiki への書き込みを行う, 表 1 に示した KawaWiki メソッドが用意されている。

メソッド名	説明
kawawiki_query	KawaWiki クエリを実行する
kawawiki_read	指定の Wiki ページを読み込む
kawawiki_write	Wiki ページを作成する
kawawiki_delete	指定の Wiki ページを削除する

表 1 KawaWiki メソッド一覧

エージェントはひとつのクラスとして定義され, execute メソッドを必ず持っている必要がある。図 6 にエージェント記述言語によるエージェントの記述例を示す。このプログラムの詳細は後述する。

```

01: <?xml version="1.0"?>
02: <rdf:RDF
03:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
04:   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
05:   xmlns:wiki="http://kawamoto/kawawiki_agent/kawawiki.rdf#"
06: >
07:   <wiki:Person>
08:     <wiki:Name rdf:parseType="Literal"><%Name\\/^([a-zA-Z0-9 ]+$/%></wiki:Name>
09:     <wiki:Hobby rdf:parseType="Literal"><%Hobby\\/^.+$/%></wiki:Hobby>
10:     &list(
11:       <wiki:Project rdf:resource="<%Project\\project%"/>
12:     );
13:   </wiki:Person>
14: </rdf:RDF>

```

図 4 KawaWiki テンプレートの例

```

01: <?php
02: class eventrulef
03: {
04:   function execute(){
05:     $query = <<<EOD
06:     PREFIX wiki:<http://kawamoto/.../kawawiki.rdf#>
07:     PREFIX rdf:<http://www.w3.org/.../22-rdf-syntax-ns#>
08:     SELECT
09:       ?summary_official,
10:       ?summary_private,
11:       ?time_private
12:     WHERE {
13:       ?x rdf:type wiki:event
14:       ?x wiki:Time ?time_official
15:       ?x wiki:Summary ?summary_official
16:       ?x wiki:Type ?official
17:       FILTER regex(?official, "seminar|meeting").
18:       ?y rdf:type wiki:event
19:       ?y wiki:Time ?time_private
20:       ?y wiki:Summary ?summary_private
21:       ?y wiki:Type ?private
22:       FILTER regex(?private, "private").
23:       FILTER regex(?time_private, ?time_official).
24:     }
25:   EOD;
26:   $result = kawawiki_query($query);
27:   $output = Array();
28:   if(!empty($result)){
29:     foreach($result as $value){
30:       $output[] = $value["?time_private"]->label . " の "
31:       . $value["?summary_private"]->label . " が "
32:       . $value["?summary_official"]->label
33:       . " と衝突しています." ;
34:     }
35:   }
36:   return $output;
37: }
38: }
39: ?>

```

図 6 エージェントの記述例

3.4 KawaWiki の実装

KawaWiki の利用者インタフェースは図 7 に示す構成になっている。左部分は Wiki ページの表示を行う Wiki エリアであり、右部分はエージェントの操作やメッセージを表示するエージェントエリアである。

Wiki エリアではユーザが Wiki ページを閲覧したり、新たに作成したり、編集したりできる。また、KawaWiki テンプレートの作成や編集、エージェントの管理も行うことができる。

エージェントエリアではエージェントが利用者に対して通知や警告を行う。利用者はエージェントからの通知や警告を見て、Wiki 上の矛盾を修正する。また、利用者はエージェントエリアでエージェントを起動することができ、エージェントからのメッセージを受け取ったり、エージェントに対して指示できる。

4. KawaWiki での整合性検証

KawaWiki で提供されている機能を用いて、人間とエージェ

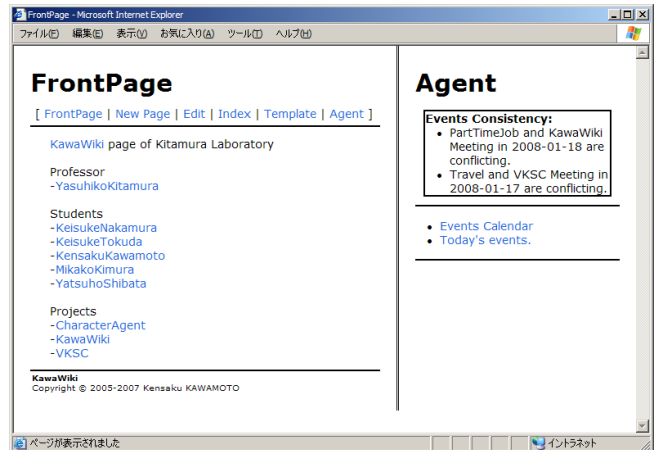


図 7 Wiki エリアとエージェントエリア

ントは KawaWiki 上で協調しながら整合性のとれたコンテンツの生成や修正を行うことができる。

4.1 構文レベルの整合性検証

構文レベルの整合性検証とは整数や文字列といった属性の型や数値の範囲を KawaWiki テンプレートに定義することによって整合性を維持する方法である。例えば、「年齢には 0 以上の整数が入る」、「人名には文字列が入る」などと定義することで整合性を維持する。KawaWiki では構文レベルの整合性検証の手段として、KawaWiki テンプレートが利用されている。テンプレート上に正規表現を使用して属性値の取りうる範囲を定義しておくことで正しい値が記入されたかを検証できる。

例えば、図 4 で示す KawaWiki テンプレートにおいて、氏名 (Name) は英数字を用いるように定義されている。ここで利用者が “Kensaku@Kawamoto” のように、氏名の記述に英数字以外の文字を用いると図 8 に示す警告が表示される。この警告 Incorrect Value は Semantic Tag で囲まれた値に誤りがあったことを示している。これ以外にも記述自体が存在しない場合の警告 (No description) や記述に重複が存在する場合の警告 (Too many descriptions) がある。

4.2 型レベルの整合性検証

型レベルの整合性検証とは、属性の型を定義することによって整合性を維持する方法である。例えば、「教授名」が入る項目

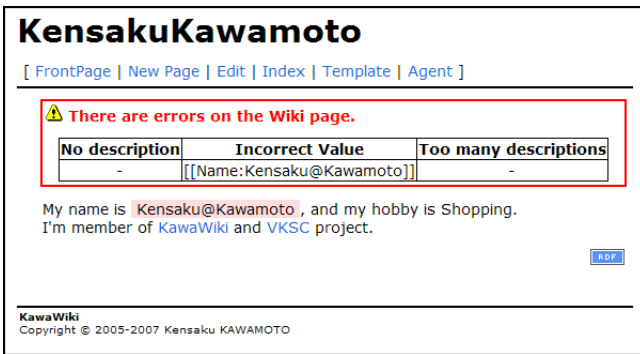


図 8 警告表示

に「生徒名」が入ってしまう「国籍名」が入る項目に「県名」が入ってしまうという不整合を防ぐための方法である。KawaWikiにおける型レベルの整合性検証の手段として KawaWiki テンプレートに変数の取る範囲としてクラス名を指定する。KawaWikiにおいてテンプレートはクラスであり、テンプレートより生成されるページはインスタンスの関係にある。例えば、[person] についてのテンプレートを作成する際、このテンプレートに「所属するプロジェクト」という属性を入れるとする。このときテンプレートに「プロジェクトはクラス [project] の値をとる」と定義することによって、ページ編集時にプロジェクトの項目に入力された値はテンプレート [project] から生成された Wiki ページであるかどうかをチェックし、ページ上の整合性を維持する。例えば、図 4 で示す KawaWiki テンプレートにおいて、11 行目で示されるプロジェクト (Project) は project テンプレートから生成されたインスタンスを値として用いるように定義されている。もし project 以外のインスタンスを代入すると図 8 同様の警告が表示される。

この整合性検証は継承にも対応している。クラスとインスタンス、クラスとサブクラスの関係は kawawiki.rdf という名称の RDFS ファイルに記述されている。例えば、図 9 では、[professor] は [person] のサブクラスであることが記述されている。したがって、[person] のインスタンスをとる属性に [professor] のインスタンスである “YasuhikoKitamura” を代入することは許される。

```
<rdfs:Class rdf:ID="person"/>

<rdfs:Class rdf:ID="professor">
  <rdfs:subClassOf rdf:resource="#person"/>
</rdfs:Class>

<wiki:professor rdf:about="http://localhost/kawawiki/index.php/rdf/YasuhikoKitamura#YasuhikoKitamura"/>
```

図 9 kawawiki.rdf

4.3 関係レベルの整合性検証

関係レベルの整合性検証は、属性間で矛盾がないようにするための方法である。例えば、「KawaWiki」という [project] に参加している」と記述している「Kensaku Kawamoto」という [person] ページがあるとする。「KawaWiki」の [project] ページではメンバー欄には「Kensaku Kawamoto」が入っていることで整合性を維持していることになる。関係レベルの整合性検証として、KawaWiki では関係レベルの整合性を検証するために KawaWiki クエリを利用している。

図 5 に示す KawaWiki クエリでは KawaWiki プロジェクトに所属するメンバーの一覧が取得でき、その内容を KawaWiki プロジェクトのメンバー変数に反映させることにより整合性の維持を行うことができる。

Wiki コンテンツの矛盾においては、直接属性値を更新することで整合性を維持できる場合もあるが、内容によっては整合性の維持は利用者に委ねなければならない場合もある。この場合は、エージェントが利用者に対して警告を表示する。例えば、研究室内のスケジュール管理を行う場合、メンバーの個人的な予定と公的な予定が衝突している場合が考えられる。この場合には、どちらの予定を変更すべきかは利用者に委ねなければならない。KawaWiki ではエージェント記述言語により、整合性の検証と警告の表示を行うことができる。

図 6 に、個人的な予定 (private) と公式的な予定 (seminar, meeting) が同じ日に衝突しているような個人的な予定を取得し、警告を示す例を示す。12 行目～16 行目で予定のタイプ (wiki:Type) が seminar か meeting である予定をすべて取得し、その公式的な予定の日時 (wiki:Time) と内容 (wiki:Summary) を変数 (?time_official) と (?summary_official) として取得している。同様にして、17 行目～21 行目では個人的な予定の日時と内容を変数 (?time_private) と (?summary_private) として取得している。22 行目では衝突している予定を絞り込むために、個人的な予定の日時と公式的な予定が同じである予定のみを抽出している。27～36 行目において、エージェントはその結果をユーザに対して警告表示している。エージェントからの警告表示は図 7 に示されている。

5. 関連研究

現在、いくつかの Semantic Wiki が開発されており、主な目的によって 3 つのタイプの Semantic Wiki を挙げることができる。

a) RSS が自動生成される Wiki

Wiki ページを作成したり編集したりするだけでユーザが意識することなく自動的にメタデータを示す RDF である RSS が生成される。このタイプの Wiki システムは多く存在し、その

代表的なものに PukiWiki^(注2), YukiWiki [8] などが挙げられる。これらの Wiki システムによって生成される RDF は、タイトル, 更新日時, リンクなどのような Wiki ページに関するメタデータに限定されている。

b) Semantic Web 情報やオントロジを生成するための Wiki
あらかじめ独自に定義された RDF/XML 構文記述用のタグや構文を使用して記述することで, RDF/XML 形式の情報が生成されたり, オントロジが構築される。このタイプの Wiki は, PeriPeri^(注3), Rhizome Wiki [9], COW [10] などが挙げられる。これらの Wiki は Semantic Web 形式での情報発信を主な目的としており, Wiki を利用して協調的に Semantic Web 情報やオントロジの構築を可能にする。

c) Semantic Web 情報の付加を支援する Wiki
従来の Wiki の記述に専用のタグを追加することで, Wiki ページ上の情報の属性を記述したり, Wiki ページ間のリンク関係を記述することができる。このタイプの Wiki は, Semantic MediaWiki [11] が挙げられ, Semantic Web を利用した情報検索を主な目的としている。例えば, Tokyo に関する記述中に [[capital of::Japan]] というタグを記述することによって, Tokyo という Wiki ページから Japan にリンクを行った際に, リンク先との関係が首都であるということを定義することができる。また, [[population:=12,000,000]] というタグにより, 12,000,000 という数字が人口を表しているということを定義することができる。これらの情報はクエリを用いて他のページから参照することができる。例えば,
{#{ask: [[capital of::Japan]] | ?population}}
と Wiki ページ上に記述すれば, Japan の首都の population の値を取得せよという意味のクエリになり, 結果として 12,000,000 が Wiki ページ上に表示される。

従来の Semantic Wiki は Semantic Web 形式での情報発信や Semantic Web を利用した情報検索を主な目的としている。本研究ではエージェントを Semantic Wiki の中に組み込み, 人間利用者と協力して Wiki 上での編集活動に参加することを目的としている。

6. まとめと今後の課題

本論文では, エージェントと人間の協調を実現する新たな Wiki システム KawaWiki と Wiki コンテンツの整合性を維持する仕組みについて述べた。エージェントと人間の共通情報表現として Semantic Web の枠組みを利用し, エージェントが情報を収集するための Semantic Web クエリ言語として SPARQL と人間が Semantic Web 情報を生成するための KawaWiki テンプレートを導入している。また, PHP をベースとして, エー

ジェントの動作を記述するためにエージェント記述言語を導入した。またこれらの仕組みを利用した Wiki コンテンツの整合性維持の手法について述べた。

今後の課題としては, 本手法の有効性を示すことにある。現在, 研究室の活動管理を行う研究室 Wiki を開発中であり, その実際の運用を通して, その有効性を示してゆく。

文 献

- [1] 藤田茂, 打矢隆弘, 今野将, 北形元, 原英樹, 菅沼拓夫, 木下哲男, 菅原研次, 白鳥則郎, 共生コンピューティング (1) - 概念とモデル -, InterSociety2005, 2005.
- [2] Bo Leuf and Ward Cunningham. *Wiki Way*. ソフトバンクパブリッシング, 2002.
- [3] 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.
- [4] 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.
- [5] Kensaku Kawamoto, Yasuhiko Kitamura and Yuri Tijerino. KawaWiki: A Template-Based SemanticWiki Where End and Expert Users Collaborate, *5th International Semantic Web Conference*, Poster paper, 2006.
- [6] Kensaku Kawamoto, Yasuhiko Kitamura, and Yuri Tijerino. KawaWiki: A Semantic Wiki Based on RDF Templates, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology - Workshops*, 425-432, 2006.
- [7] Kensaku Kawamoto, Motohiro Mase, Yasuhiko Kitamura, and Yuri Tijerino. Semantic Wiki Where Human and Agents Collaborate, *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology - Workshops (to be published)*, 2008.
- [8] 結城浩, 結城浩の Wiki 入門, インプレス, 2004.
- [9] Adam Souzis, Building a Semantic Wiki, *IEEE Intelligent Systems*, Vol. 20, No. 5, pp. 87-91, September/October 2005.
- [10] Jochen Fischer, Zeno Gantner, Steffen Rendle, Manuel Stritt, and Lars Schmidt-Thieme. Ideas and Improvements for Semantic Wikis, *Proceedings of ESWC 2006*, pp. 650-663, 2006.
- [11] Max Völkel, Markus Krötzsch, Denny Vrandečić, Heiko Haller, and Rudi Studer. Semantic Wikipedia, *Proceedings of the 15th international conference on World Wide Web (WWW 2006)*, pp. 585-594, May 23-26 2006.

(注2): PukiWiki Developers Team, PukiWiki, <http://pukiwiki.sourceforge.jp/>

(注3): KritTer, PeriPeri, <http://www.usemod.com/cgi-bin/mb.pl?PeriPeri>