

# フォークオントロジーの提案と構築環境の開発

川島隆徳<sup>1\*</sup> 往住彰文<sup>1</sup>

<sup>1</sup> 東京工業大学大学院社会理工学部価値システム専攻

**Abstract** オントロジーの知識を持った専門家だけが構築できる伝統的なオントロジーに対して、専門的知識無しで構築できるオントロジーは、オントロジー構築の敷居を下げるばかりでなく、新しい機能や用途を持つ可能性がある。フォークソノミーの手法を応用したオントロジーとしてフォークオントロジーを定義し、その概念的な枠組みの特徴づけをおこなった。1)多義性を許し、2) 大規模性を前提とするフォークオントロジーの、大人数による協調構築をサポートする GUI エディタとして、Folk Ontology Workspace(FOWS)を設計し実装した。

## Folk Ontology Proposal: Conceptual Framework and Development Tool

Takanori Kawashima<sup>1\*</sup> and Akifumi Tokosumi<sup>1</sup>

<sup>1</sup>Department of Value and Decision Science, Tokyo Institute of Technology

**Abstract** A major obstacle for ontology construction is that it requires high levels of both domain expertise and ontology skills to maintain internal consistency. In this paper, we propose a conceptual framework for a folk ontology which people can construct with only the appropriate domain knowledge. In order to realize this collaborative ontology construction, we have developed the Folk Ontology Workspace (FOWS), which is an ontology editor for distributed environments implemented in Java.

**Keywords:** folksonomy, ontology, collaborative ontology, ontology editor

## 1 はじめに

オントロジーはコンピュータが複雑な情報を扱うための知識表現として、近年様々な分野での利用が検討されている[1]。特にセマンティックウェブのツール群の一つとして OWL 言語が策定されるなど、Web 上の資源を管理検索するための基礎技術として盛んに研究がなされている。

オントロジーの構築に関する研究もまた盛んであり、様々な方法論[2]と構築のためのツール[3]が存在する。しかし、多くの方法論や構築ツールは複雑であり、オントロジー工学に関する知識無しに使いこなすことは難しい。このため、多くのユーザにとつ

てはオントロジー構築のための敷居が高くなり、たとえオントロジーが記述する分野の知識があったとしても、オントロジー構築に参加することは困難である。

このような人的リソースに関する問題を解消する最も直接的な方法としてオントロジーの自動生成が考えられるが、現段階の自動生成研究では完全に新しいオントロジーを生み出すのではなく、既存の構造化された知識の再編成が主となっている[4]。

一方、2004年ごろからフォークソノミーという新しい知識構造構築の手法が目ざされ始めた。情報システムのユーザ自身に分類のためのタグ付けをしてもらうというこの手法は瞬く間に広がり、ブックマーク、写真、ブログ記事など様々な既存のコンテンツに対する分類が行われるようになった。当然、このフォークソノミーを他の情報システムに利用しようという研究[5][6]も登場してきた。しかし、フォー

\*連絡先：東京工業大学大学院 社会理工学研究科  
価値システム専攻 往住研究室  
〒152-8552 東京都目黒区大岡山 2-12-1  
E-mail: kawashima.t.aa@m.titech.ac.jp

クソノミーによって生み出されたタグは、それを生み出したシステム内では十分に有用であるものの、外部ではそのデータ構造の単純さゆえオントロジーほど有効ではない。

そこで、フォークソノミーのような手法を用いてオントロジーを構築するという研究[7][8]が近年新しい構築方法として注目されている。フォークソノミーのような手法とはつまり、複数人でオントロジーを構築するという手法である。そのようなアイデアはオントロジー研究の初期から存在したが[9]、昨今の研究は単一のオントロジーを複数人で、しかもあまり厳密では無い形式で構築するという点が異なっていると言える。

このような協調構築オントロジーの研究はまだ始まって年数が浅い。その方法論として確立されたものではなく、有効な方法を暗中模索している段階にある。Liuら [7]は、協調構築オントロジーのためのエディタを開発し、その上で複数人のオントロジー工学の知識がない参加者に協同でオントロジーを構築してもらい、そのような人達によってオントロジーが拡張されるかを評価している。Braunら [8]は、ドメインオントロジーを複数人で構築するための *Ontology Maturing* という方法論の元で協調構築用のツールを作っている。また、Völkelらの *Semantic Wiki*[10]では、オントロジーを扱える *Wikipedia* によって協調構築を実現しようとしている。

これらの研究に共通する問題点として、編集のための敷居を下げることにしっかりとした考察がなされていないということがある。もちろん、実際にどれだけのユーザを取り込むかは研究の最終目的によって異なるが、ユーザに大きな負担がかかるような難解なデータ構造や使いづらいエディタでは結局人を集めることができず、理想のみで実現しないということになってしまう可能性がある。

そこで本稿では、そのような点にも注目した新しい協調構築オントロジーの提案と、それを構築するためのエディタを紹介する。

以下、2章で協調構築オントロジー「フォークオントロジー」の提案と考察を行い、3章でそのための開発環境である *FOWS*(*Folk Ontology Workspace*)の紹介を行う。4章では今後の展望を紹介し、5章で本稿の総括を行う。

## 2 フォークオントロジー

本章では一般の人々（フォーク）によって記述される新しいオントロジーであるフォークオントロジーを提案し、その性質について考察する。

### 2.1 フォークオントロジーとは

フォークオントロジーは一般の人々によって構築されるオントロジーである。ここで言う一般の人々とは、オントロジー構築の方法論に関する知識を持たないが、構築中のオントロジーが扱う領域に関しては知識を持っており、また最低限エディタを使うことができる人々のことである。フォークオントロジーは、その人々の持つ主観的な知識や判断に従って編集される。当然、見方の違いや知識の量によって編集される内容に差が出るため、その差を内包できるオントロジーになる。

既存の協調構築オントロジー[7][8]とフォークオントロジーは本質的な問題意識や方向性は同一である。しかし大きな違いとして、フォークオントロジーは潜在的編集者のためにオントロジー構築の敷居を下げる点に注目しているということが挙げられる。

フォークオントロジーはいわゆるライトウェイトオントロジーという種類に分類されるものである。ライトウェイトオントロジーの定義は古崎ら[1]によれば、「情報論的な利用効率を重視したオントロジー」であり、*is-a* 関係以外の複雑な公理等を持たないオントロジーのことである。オントロジー工学の知識を持たない人たちが作る以上複雑な公理等を記述することは難しく、フォークオントロジーはライトウェイトオントロジーになる。

フォークソノミーとフォークオントロジーは、一般の人々が作るという点が同じであるため似たような特徴を多く持つ。しかし、*Flicker*<sup>1</sup>や *Del.icio.us*<sup>2</sup>などのフォークソノミーサイトは、写真やブックマークに対してタグをつけるだけであり、概念の記述がなされているわけではない。従って、写真やブックマークの分類以外に簡単に応用することは難しいし、もともとそれ自身で十分役に立つものになっている。一方フォークオントロジーはその編集作業そのものが何かの役に立つわけではなく、何らかの知識を扱うアプリケーションなどに利用されて初めて役に立つ。

タグに対するメタタグが記述できるようなフォークソノミーは存在するが、フォークオントロジーはさらにそれを推し進めて、タグとそれを表象とする概念に対してタグ付けを行うものと考えられることもできる。

### 2.2 フォークオントロジーの特長

フォークオントロジーには5つの大きな特長が考

<sup>1</sup> <http://www.flickr.com/>

<sup>2</sup> <http://del.icio.us/>

えられる。

### 1) 大規模なデータ量を持つ

大人数で作業するという性質上、少人数でオントロジーを作る場合に比べて編集される概念の数が圧倒的に増える。これによって、個人で作るオントロジーに比べて包括的な領域を扱うことができ、その応用性が広がることが考えられる。

### 2) 現場の知識を反映する

Mathes[11]によれば、フォークソノミーの大きな強みの一つに、それがユーザの語彙によって作られていることがあるという。情報システムを考えたとき、そこで使われる語彙はシステムのデザイナーや分類構造の設計者の語彙となる可能性もあるが、それがユーザにとって使いやすいとは限らない。結局、ユーザが自分達自身で語彙を決定できるのが最も理想的な形であると考えられる。

フォークオントロジーを利用したシステムにおいても同じような強みが発揮される。ユーザがシステムに使われているオントロジーをチューニングすることができ、より使いやすく進化させていくことができる。

### 3) 知識の多義性を確保できる

多くのユーザによって編集されるフォークオントロジーにおいて、「 $X=A$  and only  $A$ 」と定義することには問題がある。このように一通りの定義しかできない場合、見方の違う複数の人が同じ概念を編集しようとしたときに編集合戦が起こり、最後に編集した人（あるいは最初に編集した人）の意見のみが反映されることになってしまうからである。従ってフォークオントロジーは一つ概念に対する複数の見方を内包できるようなデータ構造を持たなければならない。

現実の知識は多義性を持ち、その知識が用いられている文脈や背景によって様々な意味を持ち得る。上述したようなデータ構造を持つことによって、そのような知識の性質の一部をオントロジーに含ませることができると考えられる。

とは言え、複雑な構造で多義性を表現しようとするればそれはユーザにとって分かり難いものになってしまう。従って、一回の編集単位でのデータ構造は簡単であるが、それが集まることによって多義性が表現されるという形式を採用している。

### 4) 作り手が存在する

既存のオントロジー設計手法においては、設計者は作ろうとしているオントロジーの領域知識だけで

はなく、オントロジー工学に関する知識も有している必要があった。従って、オントロジー構築の敷居が高く、潜在的なオントロジーの作り手を遠ざけてきた。結果、オントロジーが必要とされており、それを利用する技術が多くあるのにもかかわらず、オントロジーの作り手が少ないという状況になってしまった。

フォークオントロジーにおいては、この敷居を下げることによって領域知識を持つ大勢の人がオントロジー構築に参加できるようになる。さらに、容易に導入できる開発環境があれば、有志によるオントロジー構築プロジェクトなどが立ち上げやすくなり、オントロジーの絶対量が増えるということも考えられる。

### 5) 集合知の側面

複数人で共同して一つのオントロジーを構築していく際に集団的知性が発揮され、よりよいオントロジーが構築される可能性がある。それだけではなく、オントロジーが知能の要素である知識の表現方法だということを考えると、フォークオントロジーそのものが集合知の側面を具現化したものになる可能性もある。

## 2.3 フォークオントロジーの欠点

フォークオントロジーにはその利点と表裏一体となるような二つの問題点が考えられる。

### 1) データが単純で曖昧

これは知識の多義性を確保できることの裏返しとなる欠点である。多義的に知識が表現できてしまうため、フォークオントロジーを利用して推論を行う場合などに概念をしっかりと定義できないという問題が生じる可能性がある。従って、論理演算処理や自動的な意味処理には既存のフォーマルなオントロジーの方が適している可能性がある。また、悪意ある編集者による編集や、非常識なもの見方などがオントロジーに含まれてしまう可能性もある。

### 2) 少人数の独断的知識になる可能性

Heymannら[6]によれば、Del.icio.usの全タグの56%は使用頻度上位10%のユーザによってつけられていたという。フォークオントロジーにおいても同様の現象が起こる可能性はあり得る。ただし、従来のオントロジーも少人数によって構築されていることが多いことから、これはそれ程大きな問題にはならない可能性もある。

以上の二つの問題はフォークオントロジーの性質に深く根ざしているが、これらの問題への対策を二

つ提案する。

一つは、ユーザ間でお互いの編集を評価できるシステムにすることである。もし推論等を行う上で概念をしっかり定義したい場合は、最も評価が高い編集結果を採用すればよい。また、受け入れがたい編集結果には悪い評価がつくため、そのような編集結果はオントロジーを利用するアプリケーション側で適当なボーダーラインを決めてフィルターをかけることができる。さらに、評価が行われることによって忒意性が下がる可能性もある。評価に関しては、オントロジーを利用するアプリケーション側からもフィードバックがあるなど、編集よりもう一段階敷居が低いものにして、さらに多くの評価を集めるような仕組みにする必要があるだろう。

もう一つは、全ての編集結果にその編集を行ったユーザの情報を記録し、ユーザ単位でオントロジーの再構築を行えるようにすることである。例えば、ある悪意ある編集のみを行うユーザがいた場合、その人の行った編集全てを取り除いたオントロジーを生成できる、といった具合である。ただし、これはオントロジーそのものを変えてしまうわけではなく、オントロジーのサブセットを作るという形で実行されるべきだろう。このようなシステムが搭載されていれば少人数の編集率が高い集団の編集結果を除いた、忒意性の低いオントロジーを作り出すこともできる。また、ユーザ自身に対する評価や、ユーザの専門などの個人情報があれば、それに基づいて様々な面を示したオントロジーを切り取ることができる。もっとも、切り取ったオントロジーの規模が小さくなってしまい、使い物にならなくなってしまう可能性はある。

## 2.4 実現のための要件

以上のような特徴をもつフォークオントロジーを実際に構築するためには、以下のような要件を満たさねばならない。

### 1) フォークオントロジーのエディタ

インターネット上で百科事典を作る試みがwiki上で成功したように、フォークオントロジーが成立するためにもそのための構築環境が不可欠である。もちろん、既存のツールであるwikiを使ってフォークオントロジーを作ることは可能であるが、大量のタグを打ち込むなどユーザ側に大きな負担をかける作業になる可能性が高い。敷居を下げるためには、ユーザのインプット作業を最低限に抑えるようなGUIベースのエディタが望ましい。また、ユーザのモチベーションを高めるような機能も必要で、例えば自分が編集したオントロジーが美しく表示される機能

等があったほうがよい。

また、エディタが軽快に動作し、インストールの手間などが少ないほうが望ましい。

### 2) 適切なデータ構造

その性質上、フォークオントロジーはあまり複雑なデータ構造を持つべきではない。複雑で厳密なデータ構造はユーザに理解されず、敷居を高くする原因の一つだからである。

最も現実的な案は「分類」を主眼に据えたデータ構造であろう。ものを分類し整理整頓するという作業は日常的感覚で行える。正確な分類というのは得てして難しいが、曖昧な分類ということであれば誰にでも高度な知識無しに可能だからである。

ユーザはエディタ越しにオントロジーを編集することになる。従って、エディタ側がデータ構造をうまく隠蔽することで、ユーザが感じるよりも高度なデータ構造を持つことは可能である。例えば、ユーザが行うある一動作がオントロジーの複数の箇所に影響を与えるが、エディタにはその最も表面的な結果しか反映されない、というようなシステムが考えられる。

### 3) 人を集めるための戦略

どれだけ敷居を下げて、編集してくれるユーザがそもそもいないのではオントロジーが構築されることはない。ただ単にフォークオントロジーのエディタをインターネット上に公開するだけでもある程度人が集まる可能性はあるが、能動的に人を集め、そしてなるべく離さない戦略が必要である。

まず、どのようなフォークオントロジーを作るのが問題である。インターネット上の不特定多数の手によってオントロジーを構築してもらう場合、映画や音楽などの趣味領域のオントロジーが向いていると予想される。学術的に高度なものや、企業内部で使うことを目的とするオントロジーなどは向いていない。逆にそのようなオントロジーを構築したい場合は、研究者の間や、社員の間でエディタを共有すればよい。

また、最初はエディタを使用してくれるユーザでも、飽きてくればすぐに使用をやめてしまうだろう。フォークソノミーと違って、フォークオントロジーの構築にはユーザ自身に大きなメリットがあるわけではないからだ。そこで、自分がやった仕事の量が一目で分かったり、他の編集者とチャットすることが可能であったり、ボランティアワークになるべく楽しく、苦が無いものにする必要があるだろう。これは、エディタの機能によって実現されるべき事柄でもある。Wikipediaが有志によって成り立っている

ことを考えると、適切な環境さえあればフォークオントロジーも同様に成り立つ可能性は高い。

## 2.5 フォークオントロジーの具体例

フォークオントロジーの具体例として、この節ではテレビゲームのオントロジーを考えてみたい。

テレビゲームがフォークオントロジーに向いている理由が3つあげられる。まず、インターネット上にテレビゲームの知識を持った人が数多くいて、さらにゲームの批評や評価といったことに興味がある人も多いこと。次にゲームには領域固有の言葉が多く存在し、また、それらの中には厳密に分類することが難しい言葉や、文脈依存の多義性を持った言葉が存在すること。最後に、テレビゲームは「ジャンル」と呼ばれるカテゴリーによって緩く分類されているものの、各ゲームソフトは様々な特徴を持っており、構築されたオントロジーによってより詳細なジャンル分けや検索などが可能になる題材であることである。

エディタ、データ構造、編集者に関する問題についてはここでは深く考えないこととして、どのようなオントロジーが出来上がるかを考えてみたい。まず、ジャンルに関するオントロジーが考えられる。「ロールプレイングゲーム」というジャンルのサブジャンルとして「アクションロールプレイングゲーム」や「シミュレーションロールプレイングゲーム」などが分類されるだろう。そのようにしてできたジャンルに、今度は具体的なゲームソフトが分類される。ゲームソフトには様々な面があるため、人によってそれがどのジャンルに分類されるかは異なるだろう。そして今度は、各ゲームソフトに対する評価が様々なタグによって記述される。また、テレビゲームにおける専門用語も、その意味は記述されなくても、どのジャンルに属する用語であるか、あるいはゲームのハードウェアやゲーム業界に属する用語であるかなどが分類される。

このようにして作られたフォークオントロジー(図1)は、「特定のタグを持ち、あるジャンルかそのサブジャンルに分類される」ようなゲームソフトを検索したり、あるテキストの内容がどのようなジャンルのテレビゲームについて語っているか、あるいはテレビゲームとの関連が低いかを判別したりすることに容易に利用可能であると考えられる。

## 3 構築環境の実装

本章では2章で述べたフォークオントロジーを構築するためのエディタであるFOWS(Folk Ontology Workspace)とそのデータ構造について述べる。

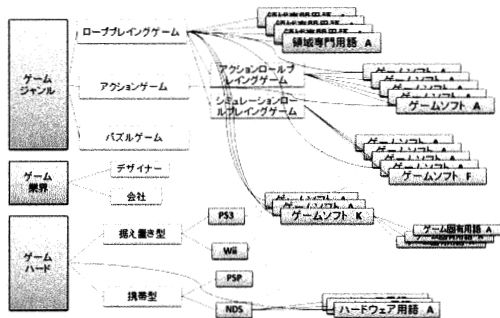


図1 ゲームオントロジーの一例

## 3.1 FOWSの特徴

FOWSはフォークオントロジーに対する考察を元にして設計されたオントロジーエディタであり、Javaで実装されている。FOWSは以下の特徴を持つ。

### 1) リッチクライアント型アプリケーション

FOWSはJavaで実装され、Java Web Startという技術によってユーザのコンピュータに配信される。これは、Javaの実行環境さえインストールされていればWebブラウザ上でリンクをクリックするだけでJavaアプリケーションが立ち上がるという技術であり、面倒なインストール作業などを伴わない。これにより、手軽さを維持しつつもAJAXなどとは異なりブラウザに依存することなくどの環境でも動く。また複雑なGUIを実現可能である。

サーバサイドはMySQLで実装されており、オントロジーの全てのデータはそこに格納されている。クライアントは必要に応じてデータを検索、取得し、また編集されたデータをサーバに送る。

### 2) GUIベースの簡易な操作

FOWSは使いやすいユーザーインターフェイスを目標として設計されており、主にマウスによる操作で簡単にオントロジーが編集できる。もちろんキーボードから入力する必要がある部分もあるが、その部分を最小限にできるように工夫している。

FOWSには現在3つのモードがある。第一の分類モードでは、ユーザは名前がついた小さな四角を、名前のついた小さなフレームにドラッグアンドドロップする。これによって、四角によって表される概念は、フレームによって表される概念の一部として分類される(図2)。

第二の記述モードでは、分類された概念の特徴を示すスロットを埋めていくというモードで、これも、スロットを選び、そこに入れたい値を選択、あるいは



図3 記述モード

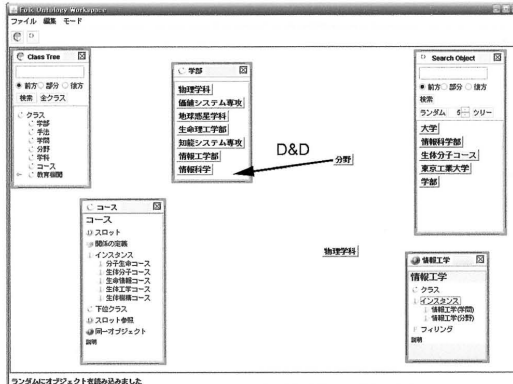


図2 分類モード

は入力すればよいという簡単な操作になっている (図3)。

第三のモードは設計モードといって、分類のための枠組みであるクラスを設計するモードである。このモードはオントロジーの根幹的な部分を操作することになるため、アクセス制限をかけてある。ここでの操作は多少複雑なものになるが、それでも最低一項目、多くても四項目ほど設定するだけでクラスを設計できるようになっている。

### 3) ユーザ単位の行動ログ

ユーザが行う全ての編集行為は「アクション」としてサーバ側にその編集を行ったユーザと日時を記録する。また、編集を行ったユーザ以外はその内容を消すことはできない。これによって、全ての編集を許可しつつも悪質なユーザの編集は後で一斉に消去するなど、オントロジーの質を向上させることができる。

## 3.2 FOWS のデータ構造

FOWS が扱うデータ構造も、フォークオントロジーに適したデータ構造になっている。この構造は RDF や OWL などの既存のスキーマに準ずるものではないが、それらの形式に変換することは可能である。

構造としては、Minsky の提唱した Frame[12] を基にした構造をしている。データ構造の概観を図4に示す。

基本的には、表1に示した5つのデータ型を持つ。FOWS のデータ構造では、全てはオブジェクトを拡張したものとして扱われる。オブジェクト一つが概念一つを表すという思想の元で作られており、各オブジェクトが (そしてオブジェクトのみが) 名前を

持つ。このオブジェクトを設計モードでクラスにすることが出来る。また、オブジェクトは他のクラスによって分類されることでインスタンスになる。一つのオブジェクトは複数のクラスのインスタンスになることが可能であり、同時にクラスとなることも可能である。

オブジェクトをクラスにするときに、そのクラスにスロットを設定する。スロットはそのクラスに分類されるものが持っている属性を示し、例えば「紅茶」クラスなら「発祥地」スロットを持っている。そして、インスタンスはそのスロットの具体的な値、フリリングを持つことになる。例えば「アールグレイ」インスタンスの「発祥地」スロットは「イギリス」(あるいは「中国」)フリリングを持つだろう。

ところで、この「イギリス」フリリングの実体は「発祥地」クラスに分類された「イギリス」インスタンスである。このような二重構造になっている理由は二つあり、一つはこのような構造にすることによって関連性のデータが増えるという理由である。もう一つは、あらかじめ分類されたインスタンスをフリリングに設定する場合には、新たにフリリングの名前をキーボードで打ち込む必要がなく、候補の中から適当なものをマウスでクリックするだけで良いため、作業がシンプルになるという理由である。

FOWS のデータ構造は決してユーザに分り易いものではないが、ユーザは特に構造を気にすることなく作業できるようにエディタのほうに工夫がこらしてある。

## 3.3 実際の作業手順

FOWS における実際の作業手順は以下のようになっている。

### 1) オブジェクトを作る

表1 データ型

データ型	説明
オブジェクト	クラスとインスタンスの実体
クラス	インスタンスを分類する枠組み
インスタンス	クラスに分類されたもの
スロット	クラスによって規定されるインスタンスの属性。実体は他のクラス。
フィリング	各インスタンスが実際に持つスロットの値。実体は他のインスタンス。

オブジェクトを新しく作る場合には、その概念の名前となる文字列を入力すればよい。複数の文字列を改行で区切って入力することで、一挙に複数のオブジェクトを作ることできる。

## 2) クラスを設計する

設計モードでオブジェクトの中から適当なものをクラスにする。この際、スロットを設定することもできる。なお、整数値や文字列などを自由に入力できるスロットを設定することもできる。

## 3) オブジェクトをクラスで分類する

分類モードでクラスにオブジェクトを分類していく。分類は最も根幹的な作業であるため、手軽にできるようにしてある。ランダムにオブジェクトを呼び出して分類するだけなら完全にマウスだけで操作できる。もちろん新しいオブジェクトを作る場合にはキーボードから入力することも出来る。

## 4) 分類されてきたインスタンスを記述する

記述モードで、(3)で生成されたインスタンスのスロットを埋めていく。これは具体的には、クラスを選び、インスタンスを選び、そのスロットを選び、候補からフィリングを選ぶという作業になり、全てマウスだけで出来る。もし適当なものが無い場合には、キーボードで打ち込むと、自動的にその実体となるオブジェクトやインスタンスが生成される。

# 4 今後の展望

本章では本研究の今後の展望について述べる。

## 4.1 FOWS の拡張

FOWS は現状では設計、分類、記述の基本的な機能しか持たないため、より実用的なアプリケーション

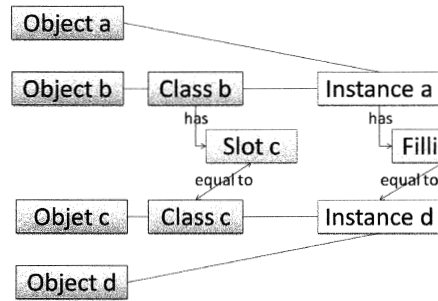


図4 データ構造

ンとして完成度を上げるために以下のような機能を追加する必要があると考えられる。

サーバサイドには実装したもののクライアントサイドには実装していない機能として、ユーザ同士がお互いの編集にコメントをつけ、また数値的に評価し合う機能がある。さらにクラス同士の関連性を、線を引くことで記述できるようなモードも実装する予定である。他にも、同じ領域のデータを編集しているユーザ同士がチャットできる機能や、半機械的な編集支援機能、また編集結果をグラフィカルに表示し、様々な形式で出力する機能、掲示板のスレッドを作るような感覚で新しいドメインのオントロジーワークスペースを立ち上げられるような機能等を考えている。これらの機能は全てユーザのモチベーションを上昇させ、編集者人口を増やすことを狙っている。

## 4.2 オントロジーの構築と検証

今回提案したフォークオントロジーの有用性や、その構築における要件に関する仮説は検証されていない。他のオントロジーの協調構築に関する研究においても、実際に大規模なオントロジーを作ったの検証はなされていない。FOWS を用いて実際にオントロジーを構築し、その規模やデータの有用性などを確認する必要がある。有用性については、構築されたオントロジーを用いて検索エンジンのクエリを作り、検索のF値から確かめることができると考えている。

また、各ユーザの編集回数や、ユーザ登録の際の簡単なアンケートなどから、潜在的なオントロジーの編集者の性質や、どのような開発環境が要求されるかなどの人的側面も調査する必要がある。

## 4.3 オントロジーの利用

フォークオントロジーは従来のオントロジーとは性質が異なるため、従来のオントロジー応用研究と

は直接的には結びつかない可能性がある。

フォークオントロジーの緩いデータ構造を考慮すれば、論理演算等よりも曖昧検索などユーザとインタラクトする局面で利用価値が高いと考えられる。どのような領域に応用が可能であるかを検証するのも今後の重要な課題である。またオントロジーのデータを誰でも自由に使えるようにサーバにアクセスするためのAPIを開発することも考えている。

## 5 まとめ

本研究では、協調構築オントロジーの一つとしてフォークオントロジーの提案と考察を行い、それを構築するための開発環境である FOWS を設計し、開発した。

フォークオントロジーは高度な構造をもった通常のオントロジーに比べて利用しにくいなど様々なデメリットも考えられるが、Wikipedia のように活動が盛んになれば必ずや将来の Web を支える大規模知識資源になるだろう。

## 謝辞

本研究の一部は 21 世紀 COE プログラム「大規模知識資源の体系化と活用基盤構築」の助成による。

## 参考文献

- [1] 古崎晃司, 來村徳信, 溝口理一郎: Web2.0 時代のオントロジー利用雑感—ライトウエイトからヘビーウエイトまで—, 第 14 回セマンティックウェブとオントロジー研究会 (SIG-SWO-A602-06) (2006)
- [2] Fernandez-Lopez M.: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies. *OntoWeb deliverable I.4*, Universidad Politecnica de Madrid,(2002)
- [3] 古崎 晃司, 溝口 理一郎: オントロジー構築ツールの現状, *Journal of Japanese Society for Artificial Intelligence*, Vol.20, No.6(20051101) pp. 707-714(2005)
- [4] Hsun-hui Huang, Yau-hwang Kuo: Towards auto-construction of domain ontology: an auto-constructed domain conceptual lexicon and its application to extractive summarization, *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*, Hong Kong(2007)
- [5] Y Yanbe, A Jatowt, S Nakamura, K Tanaka: Can social bookmarking enhance search in the web?, *International Conference on Digital Libraries: Proceedings(2007)*
- [6] Paul Heymann: Can Social Bookmarks Improve Web

Search?, *Conference on Web Search and Data Mining (WSDM2008)*(2008)

- [7] Jiahui Liu, Daniel Gruen: Between Ontology and Folksonomy: A Study of Collaborative and Implicit Ontology Evolution, *2008 International Conference on Intelligent User Interfaces (IUI 2008)* (2008)
- [8] Simone Braun, Andreas Schmidt, Valentin Zacharias: Ontology Maturing with Lightweight Collaborative Ontology Editing Tools, *4th conference professional knowledge Experiences and Visions*, Potsdam, Germany (2007)
- [9] Adam Farquhar Richard Fikes James Rice: The Ontolingua Server: a Tool for Collaborative Ontology Construction, *International journal of human-computer studies* [1071-5819] 1997, vol46, No.6, pp.707-727(1997)
- [10] Völkel, M., et al.: Semantic Wikipedia, *Proceedings of the 15th, International Conference on World Wide Web*, Edinburgh, Scotland (2006)
- [11] A. Mathes: Folksonomies - cooperative classification and communication through shared metadata, *Computer Mediated Communication, LIS590CMC (Doctoral Seminar)*, Graduate School of Library and Information Science, University of Illinois Urbana-Champaign (2004)
- [12] M. Minsky: *A Framework for Representing Knowledge*, *The Psychology of Computer Vision*, McGraw-Hill (1975)