特集論文 「エージェント」

オントロジー構築サービスONTOMOの開発

固有名詞抽出によるインスタンス・プロパティ自動推薦エージェントの評価

Development of Ontology Building Service ONTOMO

Evaluation of Ontology Recommendation Agent Using Proper Noun Extraction

川村 隆浩 Takahiro Kawamura
電気通信大学大学院 情報システム学研究科
Graduate School of Information Systems, University of Electro-Communications
kawamura@ohsuga.is.uec.ac.jp

沈 偉 I Shin
（同 上）
shinichi@ohsuga.is.uec.ac.jp

中川 博之 Hiroyuki Nakagawa
（同 上）
nakagawa@ohsuga.is.uec.ac.jp

田原 康之 Yasuyuki Tahara
（同 上）
tahara@ohsuga.is.uec.ac.jp

大須賀 昭彦 Akihiko Ohsuga
（同 上）
akihiko@ohsuga.is.uec.ac.jp

keywords:
ontology, proper noun extraction, web service, agent

Summary

Ontology-enabled services are rapidly increasing in the Web. However, those are sort of “lighter” ontologies compared with ontologies used in design and diagnosis. In this paper, we propose ONTOMO that enables internet users to take part in building those ontologies. ONTOMO is designed not for ontology experts, but for general users of the light-weight ontology. So we focused on easy-use, no installation, and cooperative work environment. Also, it has an agent function which recommends instances and properties belong to ontology classes to boost the users’ input. In the recommendation agent, we built our own proper noun extraction mechanism based on bootstrapping. Furthermore, ONTOMO provides a ontology-based blog search as a sample application to motivate the users’ ontology building. After the ONTOMO overview, we present the instance and property recommendation agent with experimental evaluation.

1. はじめに

近年，Web上でのオントロジー利用事例が増えている．多くの場合，Web上のオントロジーは，特定分野の物事を説明するためのデータ (メタデータ) で使用される用語を定義するものであり，従来の設計や診断に用いられてきたオントロジーに対して構造が単純であることが多い．Faltings[Faltings 07] らによれば，これらのいわゆるライトウェイトオントロジーでは “is-a” 関係が全関係数の80-90%を占めると言われている．しかし，一般のインターネットサービスの利用者・開発者がオントロジーを構築するには，これまでに提案・開発されたオントロジー構築ツールは専門知識を持つ研究者向けのものが多く，簡単に使えるようなツールは数少ない．

そこで我々は，インターネットサービスの利用者または開発者に対して，自らの手で比較的簡単にオントロジーを構築可能な環境を提供することを目的に，オントロジー構築サービスONTOMOを開発した．本システムの主なターゲットユーザは，オントロジーとは何であるか？その考え方や整理のスキーマ，専門用語などを知らないことを前提としている．または，オントロジーに関する知識はあるが，(当該ドメイン知識の多寡に関わらず) オントロジーを構築した経験がない，またはほとんどないことを前提としている．本システムでは，こうしたユーザがオントロジー研究者向けツールに対して抱くであろう３つの障害を取り除くことを試みている．これらは，著者の周囲のオントロジーを専門としていないシステム開発者，プログラマーなどからヒアリングによって得られたものである．

(1) 専門用語が多く，機能も豊富であるために，必要なツールの準備や使い方が分からない．
(2) オントロジーのスキーマで物事を整理して考えたことがないため，クラス，インスタンス，プロパティとして何を入力すればよいかが分からない．

(3) 大量の用語登録が困難である (この点は程度の差はあっても今回のターゲットユーザに限らず，言えるであろう).

これらの問題に対処するため，ONTOMO では以下のようなアプローチを取っている．

A. 導入準備の容易化
専用ツールのインストールを不要とし，ブラウザからアクセスできるようにする．

B. 使い勝手の向上 (ユーザインターフェースの工夫)
専門用語を避けて平易な言葉を用い，機能をライトウェイトオントロジーに必須な閲覧編集機能の一部に絞る．また，ブラウザベースでも視認性や操作性が落ちないようにする．

C. 用語登録の支援
数が多いインスタンスやプロパティは，ユーザが単独で網羅的に思い付くことは困難である．そこで，他のユーザによる入力履歴も参照し，それらに基づいて登録候補となるインスタンスやプロパティ用語を推測し，ユーザに推薦する．

D. モチベーションの維持
オントロジーの構築によって，どのようなサービスが構築されるかサンプルアプリケーションを示し，ユーザにインセンティブを与える．また，ユーザのサービスで利用可能なようにオントロジーにアクセスするための Web サービス API を公開する．

E. 共同作業の支援
仲間同士で開発できるように，初めから複数人による作業を可能とする．

上記，3 つの問題は単独の対策でなかなか解決するものではなく，5 つのアプローチとは 1 対 n で対応している．図 1 に，我々の考える問題とアプローチの対応関係を示す．また，各問題点への効果の測定に関しても，ユーザの感性的，心理的な要素によるところが大きく，必ずしも定量的に検証できるとは言えない．そこで本論では，ONTOMO の全体概要について述べた後，アプローチの 1 つである C. 用語登録の支援に焦点をおき，インスタンス・プロパティ自動推薦エージェントの実装と評価について詳述したい．これは，3. 用語登録作業が大変である，および 2. 入力すべき用語が分からない，という問題に対応するものであるが，推薦された用語の精度は，これらの問題へのユーザ支援度合いを測る 1 つのメトリクスと見なせると考えたためである．

本論の構成は以下の通りである．まず，2 章で ONTOMO の概要を述べた後，3 章でインスタンス自動推薦エージェントの機能を説明し，評価実験について述べる．また，4 章ではプロパティ自動推薦の機能と評価実験について述べ，5 章で考察とする．最後に，6 章で関連研究と比較し，7 章で今後の課題を述べる．

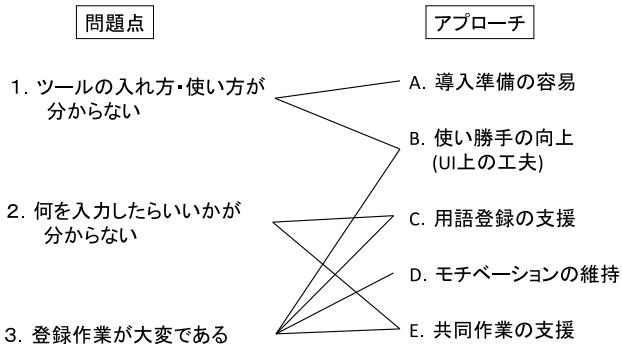

図 1 問題点とアプローチの対応

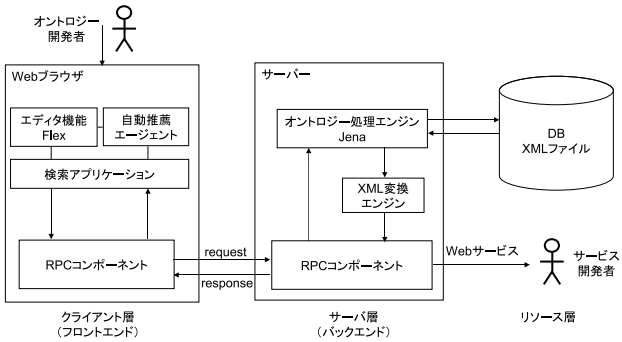

図 2 ONTOMO の構成

2. ONTOMO の概要

ONTOMO の機能は，フロントエンド (Flash Web サイト) とバックエンド (Web サービス) で構成されており (図 2)，フロントエンドにはエディタ機能，自動推薦エージェント，検索アプリケーションの 3 つがある．一方，バックエンドの Web サービスでは構築されたオントロジーを参照する API を提供しており，他のシステムからのオントロジー利用を可能としている．フロントエンド-バックエンド間は非同期通信を行なう Flash を用いたことで，高い応答性と操作性を有している．本章では，エディタ機能と検索アプリケーション，および実装について概要を述べる．

2·1 エディタ機能

ここではライトウェイトオントロジーを対象とし，クラス，インスタンス，プロパティといった主要構成要素の閲覧機能と，新規追加，編集および削除といった基本機能を提供している．

オントロジーの閲覧には，3 つの方法がある．1 つ目はオントロジーの階層構造を直感的に把握するためのグラフ表示である．ここでは Flash と Spring Graph を利用し，オントロジーの階層構造を可視化している．図 4 は，クラス，サブクラスおよびインスタンスからなるオントロジーの階層構造をバネモデルによるグラフ自動配置ライブラリ Spring Graph[Spring Graph] で可視化した様子を

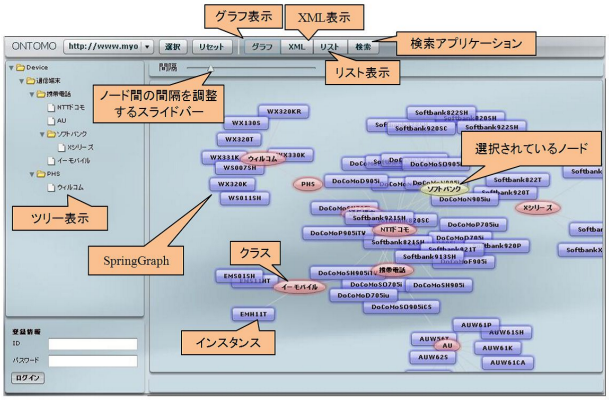

図 3 ONTOMO のインターフェース

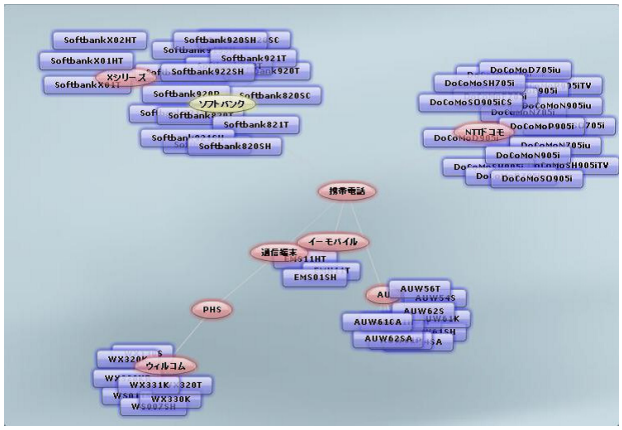

図 4 グラフ表示

示す．操作性と可視性を重視し，インスタンスがクラスに近づきすぎたり離れすぎたりしないように，インスタンスがクラスとどれぐらいの距離を置くかを調整できる．

また，クラスの配置を変える場合においても，Spring Graph は有用である．クラスを移動したら，そのクラスのインスタンスが自動的にクラスに追従し，事前に設定されたパラメーターの値によって，クラスから一定の距離をおいたところに移動する．

更に，クラスの周辺にインスタンスがたくさん配置されている場合，クラスだけに集中し，インスタンスを一時的に非表示にしたい場合がある．そこで，使いやすさと見やすさを考慮したインスタンスの表示・非表示の機能を導入し，クラス（楕円）をダブルクリックするとインスタンスを表示したり，非表示にしたりすることができるようになっている．

また，Spring Graph を利用したグラフ画面は見やすく，オントロジーの階層構造を把握しやすいといった利点がある一方，どの程度のクラスやインスタンスが定義されたかを一覧したり，クラスの定義といった詳細情報を知るには不適である．そのため，リスト表示と XML 表示も可能とした．リスト表示には，クラスの一覧，プロパティ一覧，インスタンス一覧の 3 つがあり，オントロジーの構成要素別に閲覧できる (図 5)．

図 5 リスト表示

また，XML 表示では，以下のようにオントロジーの中身である OWL[OWL] データを XML 形式で表示する．この 3 つの表示を使い合わせて，網羅的にオントロジーの構造と定義を把握できることを狙ったものである．

```
<rdf:RDF xmlns:mobilephone=
  http://www.myontology.co.jp/mobilephone/#
    xmlns:rdf="http://www.w3.org/1999/02/22-
      rdf-syntax-ns#"
    xmlns:xsd=http://www.w3.org/2001/XMLSchema#
    xmlns:rdfs=http://www.w3.org/2000/01/
      rdf-schema#
    xmlns:owl=http://www.w3.org/2002/07/owl#
    xmlns:daml="http://www.daml.org/2001/03/
      daml+oil#">
    <owl:Ontology>
      <rdfs:comment>
        MobilePhone OWL Ontology
      </rdfs:comment>
    </owl:Ontology>
    <owl:Class rdf:about="
      http://www.myontology.co.jp/
      mobilephone/#ソフトバンク">
      <rdfs:subClassOf>
      <owl:Class rdf:about="http://www.
        myontology.co.jp/mobilephone/
        #携帯電話"/>
      </rdfs:subClassOf>
    </owl:Class>
  </rdf:RDF>
```

オントロジーの編集は，グラフ表示上での右クリックで現れるポップアップメニューから行う (Web ブラウザの関係上，一般の Web アプリケーションは右クリック機能を備えていないが，ここでは Flash を利用することで右クリック可能としている)．対象となるクラスとインスタンスを右クリックするとポップアップメニューが表示され，新規定義，編集，削除などを選択することができる．

図 6 オントロジー検索

図 7 ブログ検索

また，構築したオントロジーはデータベースに保存され，OWL ファイルとして出力できる他，Web サービス API で外部からアクセスすることができる．

2·2 検索アプリケーション

オントロジー検索では，プロパティを検索項目としてオントロジーを横断検索することが可能である (図 6)．ONTOMO には，携帯電話，デジタルカメラ，メディアプレイヤーの 3 つのオントロジーが予め入力されている．そこで，共通プロパティを条件として検索した場合，3 つのオントロジーを渡って条件に満たしたものを探し出す．また，複数の検索項目を組み合わせて検索することもできる．例えば，再生時間とワンセグを検索項目として検索した場合，携帯電話とメディアプレイヤーが検索対象となる．

ブログ検索は，オントロジー検索で検索されたインスタンスに関して，Google blog 検索を行い，関連する最新ブログ情報を表示する (図 7)．インスタンスに関連する最新の情報を提供することで，情報の更新に役立つと考えられる．

2·3 ONTOMO の実装

ONTOMO は実装上，クライアント，サーバ，リソースの 3 層から構成される (図 2)．クライアント層は，非同期通信を実現するためのフレームワーク Flex[FLEX] を使用し，柔軟なユーザインターフェースを提供している．サーバ層には，Jena[JENA] で実装されたオントロジー処理エンジンがあり，主にオントロジー処理とデータベース処理の 2 つを行っている．リソース層は，データベース MySQL を用いて XML 形式でオントロジーデータを保存・管理する．クライアント–サーバ間の非同期通信は Flex Remote Object，Flex Data Service 等の Flex コンポーネントにより実現している．ユーザが Flash ベースの Web 画面上で操作を行うと，クライアントがそのリクエストをサーバに送り，サーバ上の Flex コンポーネントがリクエストに応じた関連 API を呼び出し，処理を行う．その結果は XML の形でクライアントに返され，XML 変換エンジンを通して画面へ反映される．また，サーバ層ではオントロジー処理エンジンの結果を XML 変換エンジンを通じて，ツリー用 XML とグラフ用 XML に変換し，クライアントに渡す．クライアント層では，受け取った XML を XML 変換エンジンを通じて Flash に適した形式に変換し，画面上に表示している．

3. インスタンス自動推薦エージェントと実験・評価

実際にインスタンスを追加する作業では，何らか手掛かりがなければ，網羅的にインスタンスを増やすことは困難である．また，家電製品，CD/DVD，パッケージソフトのように日々，増加していくものをいち早くオントロジー化するためには，手作業では限界がある．そこで，本研究では複数ユーザの利用履歴をデータベースに記録し，それらに基づいてインスタンスの候補を推薦するエージェントを開発した．自動推薦エージェントは，ユーザにより直接追加されたいくつかのインスタンスをシードとして，固有名詞抽出を用いて同じ集合に属する他のインスタンスを推測し，ユーザに推薦する．最終的には，ユーザが推薦された候補から，自分の判断で正しいと思う候補をインスタンスとして追加する．例えば，ユーザが “日産”，“ホンダ” という順に 2 つのインスタンスを追加した後，新たに 3 つ目のインスタンス “トヨタ” を入力すると，利用履歴に基づいて最新の 3 インスタンス { 日産，ホンダ，トヨタ } をシードとして固有名詞抽出を用い，既登録のインスタンスや関係がない語を除外したうえで候補となる語をユーザに推薦する (図 8)．

3·1 ブートストラッピングに基づく固有名詞抽出手法

インスタンス自動推薦エージェントには固有名詞抽出手法が使われている．ここで，固有名詞抽出とは，入力された単語と同じ集合に属すると推測される他の用語を

図 8 インスタンスの自動推薦

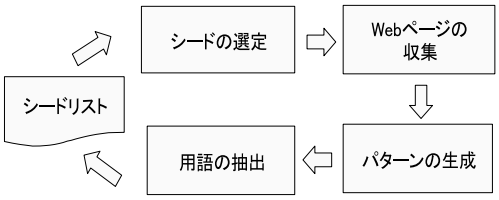

図 9 ブートストラッピング処理の概要

抽出するものである．インターネット上には，既に固有名詞抽出サービスがいくつか公開されており，有名なものとして GoogleSets[Google Sets] や SEAL[SEAL] がある．しかし，我々の調査では GoogleSets や SEAL は最新の情報への対応が遅いという難点がある．また，仕様も公開されておらず，データの安定性と信頼性に問題がある．そこで，我々はブートストラッピング手法 [Brin 98] を用いて，独自の固有名詞抽出を開発した．ブートストラッピングとは，少量のシード用語に基づいて文書からパターンを生成し，このパターンを使って再び文書からほかの単語を抽出し，抽出した単語を使ってさらにパターンを生成するという手法である．この処理を繰り返すことで少量のシード用語から大量の用語を抽出できる．以下，4 ステップに分けてこれを説明する (図 9).

(1) シードの選定

上記の処理が実行されるためには，いくつかのシードが必要である．シードの良し悪しによって，抽出される単語が大きく変わる．

インスタンスを抽出する場合は，事前に少なくとも 3 つのインスタンスを含むシードリストを用意する．一回目は任意の 3 つのインスタンスを選び，ブートストラッピング処理を行う．そして，抽出した新しいインスタンスをシードリストに追加する．二回目は前回使われた古いインスタンス 2 つと新しい抽出したインスタンス 1 つをシードとする．これは，新しく抽出したインスタンスが必ずしも正しいとは限らないためである．そこで，精度の悪化を防ぎながら，新しい要素を入れるため，二回目以降は古いインスタンス 2 つと新しいインスタンス 1 つをシードとする．

(2) Web ページの収集

検索クエリを生成し，Google や Yahoo などの検索エンジンを利用して関連する上位 100 の Web ページを収集する．インスタンスを抽出する場合，クエリは 3 つのインスタンスの組み合わせとなる．例えば，{ 日産, ホンダ, トヨタ },{GM, クライスラー, フォード } などである．

(3) パターンの生成

収集した Web ページから 3 つのシードをともに囲む HTML タグを探す．例えば { 日産，ホンダ，トヨタ } をシードとする場合, 下記のような HTML 文書を抽出したとする．

```
<td>日産</td>
<td>ホンダ</td>
<td>トヨタ</td>
<td>マツダ</td>
<td>スズキ</td>
```

$< td >$ タグは 3 つのシードをともに囲んでいるため，$< td >$ 用語 $< /td >$ をパターンとする．このパターンを利用し，ほかの HTML 文書を適用することで “マツダ”, “スズキ” を抽出できる．

一方，下記のように,“日産”，“ホンダ” を囲む HTML タグが存在するが，“日産”，“ホンダ”，“トヨタ” を囲む HTML タグが存在しない場合は，パターンが生成されず，候補用語は抽出されない．

```
<td>日産</td>
<td>ホンダ</td>
<tr>トヨタ</tr>
<td>マツダ</td>
<td>スズキ</td>
```

3 つのシードを共に囲む HTML タグを探す理由は，精度を高めるためである．条件が厳しければ厳しいほど，精度の向上につながる．経験上，シードが 2 つの場合，関係がない用語を大量に抽出してしまい，精度が大幅に低下する．

(4) 用語の抽出

続いて，生成したパターンを使って，そのパターンに当てはまる他の用語を抽出する．これは，Web ページは作成者によって記述形式が様々であるが，同じページ内ならば，記述形式が同じ場合が多いからである．本手法は同じページ内から 1 つのパターンを生成し，それをそのページの他の部分に適用し，用語を抽出する．ステップ 3 の例では，$< td >$ 用語 $< /td >$ というパターンを使って “マツダ”, “スズキ” などの用語を抽出できる．そして，抽出した用語をシードリストに追加する．

ただし，この手法で用語を抽出する際に，関係のない用語を抽出してしまう可能性がある．それを防ぐために，同じページから生成したパターンの数に閾値を設けている．この閾値を N とすると，実験の結果 $N >= 2$ の場

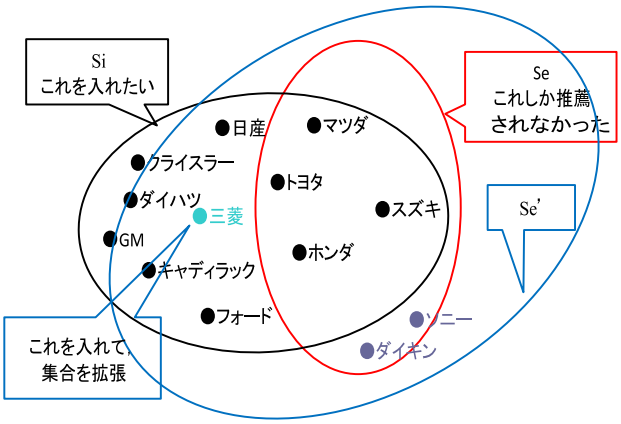

図 10 $|Se \cap Si| \ll |Si|$

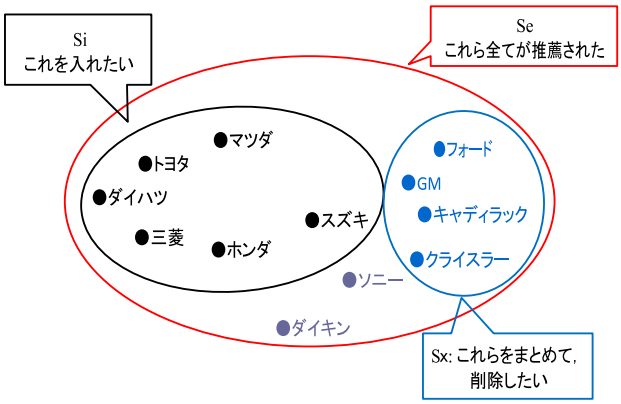

図 11 $|Se \cap Si| \ll |Se|$

合に最も効率的に用語を抽出できた．そこで，$N >= 2$ の場合のみ，生成したパターンを有効とする．そうでない場合は，そのパターン，およびそのパターンによって抽出した用語をすべて破棄する．これにより，関係のない用語を削除し，精度を高めることができる．

3·2 固有名詞抽出を用いたインスタンスの自動推薦

固有名詞抽出を利用して抽出した用語集合は，必ずしもユーザが意図する集合と一致するとは限らず，そのままユーザに推薦すると，ユーザに膨大なインスタンスを判別し，適切なものを選別するといった負荷をかける可能性がある．そこで，本論では固有名詞抽出 (集合拡張) にオントロジーの特徴である階層構造を組み合わせて抽出精度を高めるための仕組みを追加した．

まず，ユーザが意図する集合を Si，シードから固有名詞抽出を用いて推測された集合を Se とすると，Se が Si とほぼ一致することが理想的であり，その場合は Se をすべて登録すればよい．一方で，一致しない場合には次のような対策が考えられる．

§1 再現率の向上

Si の全要素のうち少数しか Se に含まれない場合 (図 10)：Se が網羅する正解インスタンスが少ないため，再現率が低くなる．そこで，Se に現れない Si の要素をシードに追加し，Se を拡張し，Se' を得ることによって，再現率を高める (実験 1).

§2 適合率の向上

関係のない語が多く，Se の全要素のうち少数しか Si の要素がない場合 (図 11)：Se が網羅するインスタンスが多いが，関係のない語も多く含まれており，適合率が低下する．そこで，次の適合化フィルタを適用し，関係がない集合 Sx を除外することにより，再現率を保ちながら適合率を高める (実験 2).

適合化フィルタとは，適合率を高めるために，固有名詞抽出から得られたインスタンス集合から関係がないと推測される語を除外する処理である．

(1) シードとして選択されたインスタンスと同クラスに属する他の既存インスタンスと比較し，重複している要素を除外する．
(2) シードとして選択されたインスタンスと異なるクラスに属するいくつかのインスタンスを新たなシードとして再度，固有名詞抽出を行い，得られた集合と比較し，重複している要素を除外する．

図 12 のような自動車オントロジーがあった場合，ユーザが [日本車] クラスの下に日本車のインスタンスを追加する場合を想定する．ユーザが意図する [日本車] 集合を Si，インスタンス { 日産，ホンダ，トヨタ } をシードとして，固有名詞抽出を用いて推測された集合を Se とすると，“ダイキン”，“ソニー”，“GM”，“フォード” など関係のない語が Se に多く含まれており，適合率が低下している (図 11)．そこで，適合化フィルタは，まず { 日産，ホンダ，トヨタ } と同クラスに属する他の既存インスタンスと比べ，重複しているものを除外する．次に，他のクラスに属するいくつかのインスタンスをシードとして固有名詞抽出を行う．ここでは，[米国車] クラスのインスタンス {GM，フォード，クライスラー } をシードとして，固有名詞抽出を用いて得られた集合を Sx とする．Sx には他の米国車が含まれている可能性が高いため，これらを Se と比較し，重複しているもの，すなわち Se で米国車とされているインスタンスを除外する．但し，集合 Sx に日本車が含まれた場合は，日本車が削除されることになり，再現率が落ちる可能性もある．

3·3 評 価 実 験

前述の通り，インスタンスの自動推薦は新製品など変化の激しいインスタンスの取得を意識したものであるが，実験においては比較的，静的なデータを用いたほうが正確に再現率と適合率を測ることができると考えた．そこで，2007 年世界新車販売ランキング [RANK] に基づき，実験 1 では 50 位までの自動車メーカー (中国メーカーと

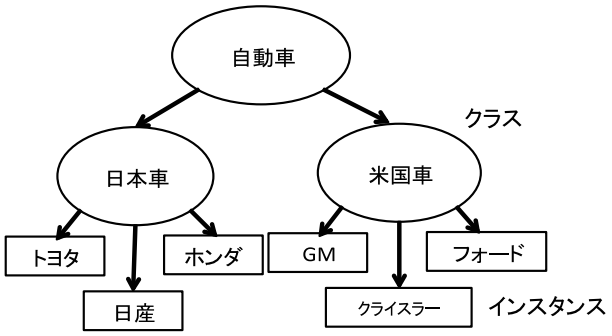

図 12 自動車オントロジーの例

日本	トヨタ, ホンダ, 日産, マツダ, 富士重工業, 三菱, スズキ, ダイハツ, 日野, スバル, いすゞ, 三菱ふそう, 日産ディーゼル, レクサス, アキュラ, インフィニティ,
アメリカ	GM, フォード, クライスラー, シボレー, ハマー, キャディラック, ビュイック, ポンティアック, サターン, GMC, リンカーン, マーキュリー, ダッジ, ジープ, Navistar
ドイツ	ポルシェ, オペル, アウディ, BMW, フォルクスワーゲン, ダイムラー, シュコダ, ベントレー, ブガッティ, セアト, マイバッハ, ベンツ, スマート, AMG, ミニ, ロールス・ロイス, ランボルギーニ
イタリア	フェラーリ, アルファ・ロメオ, フィアット, マセラティ, ランチア, アバルト, イヴェコ
イギリス	ジャガー, ロータス, ランドローバー, ローバー, アストンマーチン, PACCAR
フランス	ルノー, プジョー, シトロエン, PSA
スウェーデン	ボルボ, サーブ, スカニア
韓国	ヒュンダイ, 大宇, 起亜,
インド	タタ, マヒンドラ
ロシア	AVTOVAZ, GAZ, UAZ

図 13 インスタンスの正解集合

OTHERS を除く[*1])31 社とそれらの派生ブランドを含めた計 76 を正解集合とした (図 13). また，実験 2 では日本のメーカーとブランド計 16 を正解集合とした．尚,「ニッサン」と「日産」,「キャディラック」と「キャデラック」など Web 上には表記ゆれが多数存在するが，本実験では正解集合の表記以外は不正解とした．また，本来，自動推薦エージェントは推薦されたものをオントロジーに加えるかどうかはユーザに委ねるものであるが，ここでは推薦の精度を確かめるため，推薦されたものは重複していない限り，全て加えていくものとする．以下では，提案した固有名詞抽出と適合化フィルタを用いて，如何に再現率と適合率を高められるかを評価する．

また，評価指標としては，再現率と適合率を用いた．

再現率 = その時点で登録されているインスタンス数/全正解インスタンス数

適合率 = その時点で登録されているインスタンス数/関係のない語を含めて登録されているすべてのインスタンス数

§1 実験 1(固有名詞抽出)

実験 1 は，世界の自動車メーカーとブランド計 76 を Si として，固有名詞抽出を用いることで，操作回数に対して，どれだけ効率的にインスタンスを増やせるかを再現率，適合率を用いて評価した．初回のシードとしては，

*1 中国の自動車メーカーは，07 年ランキングからの移り変わりが激しいこと，表記ゆれ (繁体字，簡体字，アルファベット，カタカナなど) が散見されること，ブランド展開等を十分に調査しきれなかったことなどから，正解集合から外した．

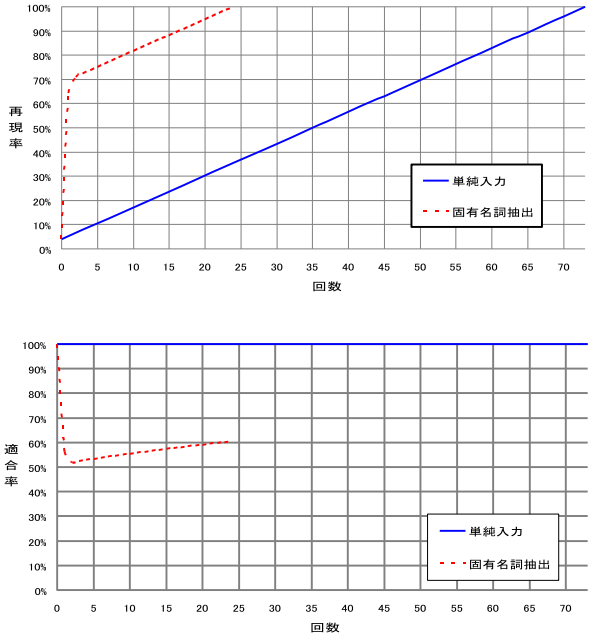

図 14 実験 1 の結果

クラス [自動車] にインスタンス { 日産, ホンダ } を登録した．そして，正解集合から登録順にまだ登録されていないインスタンス “トヨタ” を入力すると，利用履歴に基づいて既に登録された最新の 3 インスタンス { 日産, ホンダ, トヨタ } をシードとして固有名詞抽出を実行し，得られた候補集合 Se から未登録の要素をすべて登録する．これを繰り返し，正解集合の要素がすべて登録された時点で実験を終了する．

評価方法は次の通りである．

再現率 $= (3+\Delta)/76$

適合率 $= (3+\Delta)/(3+\Delta+$関係のない語$)$

（Δ は抽出したインスタンスの数）

実験結果を図 14 に示す．再現率については，単純入力の場合，システムが登録順に正解インスタンスを 1 つずつ登録していくため，入力回数に比例して上昇し，73 回目に 100%に達した．それに対し，固有名詞抽出の場合は大量の候補インスタンスが抽出され，再現率が急上昇し，2 回目には 71.1%に達している．しかし，その後は候補用語が重複し出し，上昇ペースが緩やかになり，24 回目に 100%に達した．これにより，固有名詞抽出を用いることで，ユーザからの少ないインプットでインスタンスを効率的に増やせることが確認できた．一方，適合率については，単純入力の場合は常に 100%であるのに対し，固有名詞抽出の場合は関係のない語が抽出されて低下し，24 回目でも 60.3%に留まった．

§2 実験 2(適合化フィルタ)

実験 2 は，日本の自動車メーカーとブランド計 16 を Si として，適合化フィルタを用いることで，どれだけ適合率の低下を抑えられるかを検証した．適合化フィルタ

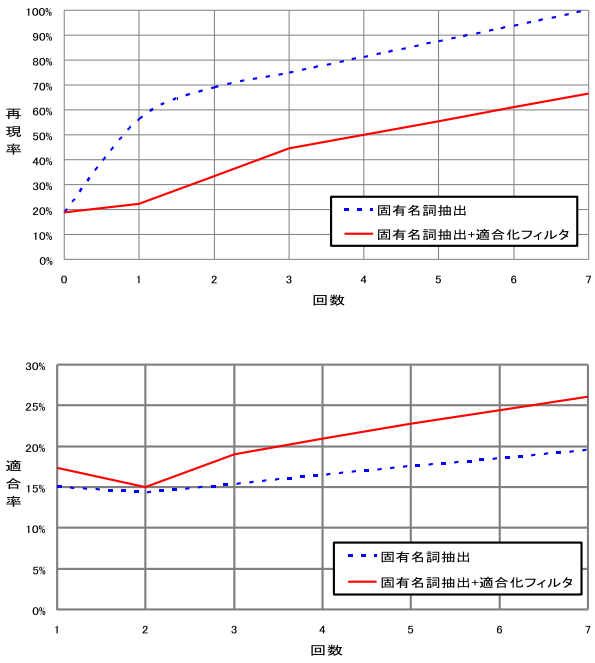

図 15 実験 2 の結果

はオントロジーの階層構造を利用した手法であるため，Si は [自動車] のサブクラス [日本車] に属するものとし，同レベルにサブクラス [米国車] があると仮定する．初回のシードとしては，サブクラス [日本車][米国車] それぞれに { 日産, ホンダ }，{GM，フォード，クライスラー } を登録した．そして，実験 1 と同じ手順 (固有名詞抽出のみ) に加えて，適合化フィルタを用いてクラスが異なると思われる用語を除外してから登録することを繰り返す．

評価方法は次の通りである．

$$\text{再現率} = (3+\Delta)/16$$

$$\text{適合率} = (3+\Delta)/(3+\Delta+\text{関係のない語})$$

（Δ は抽出したインスタンスの数）

実験結果を図 15 に示す[*2]．適合率については，固有名詞抽出が検索結果にアメリカ，ヨーロッパのメーカーやブランドを多く含み，7 回目で 19.5%であるのに対し，適合化フィルタを用いた場合は 26.1%となり，その差は回を追って広がっている．これにより，適合化フィルタに適合率を向上させる効果があることが確認できた．一方，再現率については，固有名詞抽出の場合は 7 回目に 100%に達したのに対して，適合化フィルタを適用した場合は 7 回目で 66.7%に留まった．これは関係のない語を除外するときに，いくつかの日本車も除外してしまったためである．

*2 表記上，適合率に関しては 0 回目を除く

4. プロパティ自動推薦エージェントと評価実験

4·1 固有名詞抽出を用いたプロパティの自動推薦

プロパティもインスタンスと同様，網羅的に追加することが困難であるため，自動推薦エージェントを開発した．プロパティの自動推薦も，ブートストラッピングに基づいた固有名詞抽出手法を利用してプロパティの獲得を行う．但し，プロパティ推薦の場合，Web ページを収集するための最初のクエリとしてそのクラスに属するプロパティ3 つにインスタンスを 1 つ追加したものを用いる．これは，プロパティが属するクラスのインスタンスをシードに加えることで，商品スペック表が書かれているような Web ページを多く収集でき，関連する他プロパティの抽出に役立つと考えたためである．また，固有名詞抽出の際のシードにも，プロパティ3 つにインスタンスを 1 つ加える．例えば，ユーザがカメラクラスに，光学ズーム，デジタルズーム，画素数の 3 つのプロパティを追加すると，自動推薦エージェントは，それら 3 つのプロパティとカメラクラスに属する 1 つのインスタンスを手掛かりに，焦点距離，ISO 感度，シャッタースピードなどの他のプロパティを獲得し，ユーザに推薦する．尚，二回目以降のシードの入れ替えは古いプロパティ2 つ，新しいプロパティ1 つ，および新しいインスタンス 1 つをシードとする．

本来，ブートストラッピングによる固有名詞抽出は，いわば横並びに記載された単語を抽出するものであり，プロパティとインスタンスなどレベルの異なるシードが入っているとうまく抽出できないことも予想される．しかし，商品スペック表などは多くの場合，はじめに商品名があってから仕様 (プロパティ) を載せているため，プロパティ抽出に関してはレベルの異なるシードを入れることによる悪影響は少ないと考えた．

4·2 評 価 実 験

プロパティは商品によって，専用プロパティもあれば，共通のプロパティもある．例えば，液晶サイズはカメラのプロパティであり，パソコンのプロパティでもある．したがって，インスタンスのように明確に分類することが難しく，適合化フィルタを適用すると再現率を大幅に低下させる可能性がある．そこで，評価実験においては，適合化フィルタを用いず，固有名詞抽出だけで実験を行う．また，比較的プロパティの多い商品を実験の対象とし，カメラ，レンズ，テレビ，パソコン，液晶ディスプレイ，ブルーレイプレーヤーの 6 つの商品 (クラス) を対象に，それぞれに属するプロパティを抽出した．それぞれの正解データを図 16 に示す．カメラを例に実験の流れを説明する．事前にカメラというクラスに属するプロパティ { 光学ズーム，デジタルズーム，画素数 } をシードとしてシードリストに登録しておく．また，カメラに属するインスタンスはインスタンス自動推薦により抽出し，

カメラ	有効画素,画素数,記録画素数,レンズ形式,焦点距離,光学ズーム,F値開放絞り値, デジタルズーム,シャッタースピード,撮影感度,最短撮影距離,記録メディア,記録フォーマット,液晶モニター,液晶サイズ,電池タイプ,撮影枚数 ,内蔵メモリ,ファインダー,質量,外形寸法,ISO感度,顔認識,手ぶれ補正,メーカー名,連写撮影,フレッシュ,内蔵ストロボ,動画撮影
レンズ	レンズタイプ,フォーカス,レンズ構成,絞り羽根枚数,焦点距離,最短撮影距離,最大撮影倍率,F値,画角,対応マウント,手ブレ補正,最大径,フィルター,最小口径比,質量,メーカー名,フィルターサイズ
テレビ	画面サイズ,フルハイビジョン対応,画素数,コントラスト,輝度,応答速度,画面分割,LEDバックライト,倍速液晶,液晶パネル 方式,デジタルチューナー,アナログチューナー,HDMI ,D端子入力,コンポーネント入力,PC入力端子,i.Link端子,ネットワーク端子,寸法外形,重量,壁掛け対応,消費電力,待機時消費電力,視野角,録画機能デジタル光音声入力端子,スピーカー,アスペクト比,メーカー名
パソコン	画面サイズ,解像度,モニタ接続,CPU周波数,CPU種類,二次キャッシュ容量,メモリー容量,メモリー最大容量,メモリー種類,空メモリスロット数,全メモリスロット数,PCIスロット数,HDD容量,HDD回転数,光学ドライブ,モデム,LAN,無線LAN,OS,駆動時間,寸法外形,重量,ビデオチップ,ビデオメモリ,HDDインターフェイス,光学ドライブ,フロッピーディスクドライブ
液晶ディスプレイ	液晶モニター,画面サイズ,液晶方式,有効表示領域,表示画素数,画素ピッチ,表示色,視野角,輝 度,コントラスト比,応答速度,入力端子,出力端子,音声入出力,消費電力,重量,表面処理,外形寸法,走査周波数,スピーカー,ビデオ入力,解像度,最大表示色,周波数,アスペクト比,OSDメニュー
ブルーレイプレーヤー	再生対応ディスク,HDMI端子,コンポーネント出力,プログレッシブ対応,D端子出力,S端子出力,DVI出力,i.LINK出力,光デジタル音声出力,同軸デジタル音声出力,デジタル音声出力,外形寸法,重量,消費電力,対応オーディオ,他再生メディア,入力端子,メーカー名,アップスケール

図 16 プロパティの正解集合

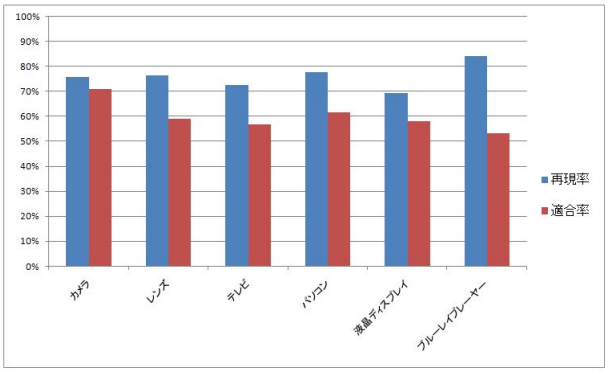

図 17 プロパティ推薦結果

登録する．そして，{ カメラインスタンス 1, 光学ズーム, デジタルズーム，画素数 } をシードとして，固有名詞抽出を実行し，候補用語を獲得した．

実験の結果を図 17 に示す．再現率は 70-80%に達しており，クラスに属するプロパティをほぼ獲得できることが確認できた．一方，適合率については全体的に 50-60%であった．本来，商品スペックが載っているメーカーなどの Web ページからプロパティを抽出するのは理想であるが，現在は抽出元ページの選別は行っておらず，商品のプロパティの意味を解説するページなどから関係のない用語を多く抽出してしまったことで適合率が低下したと考えられる．

5. 考　　察

1 章で挙げた C. 用語登録の支援に関しては，以上述べてきた通り，インスタンスとプロパティの自動推薦エージェントを利用することで，ユーザの負担を減らすことができるようになったと考えている．インスタンス自動推薦では，固有名詞抽出のみでは適合率が不十分であったため，適合化フィルタを導入した．結果的に，再現率を低下させてしまったが，適合率を向上させることができた．適合率を重視した理由は，オントロジーに詳しくないターゲットユーザ層を考慮すると，不適切なものが多く混じった候補群から正しいものを選別する作業は困難度が高いと考えたためである (ドメイン知識の多寡にもよるが)．一方で，再現率は多少低くても，ユーザは自動推薦–選別の過程を繰り返せばよいだけであり，(適合率が高ければ) 単純な作業であると言える．また，プロパティの自動推薦においては適合化フィルタが使えないため，更なる適合率の向上に向けて検索ヒット件数を参照するなど抽出元ページを選別する処理の導入を検討している．

但し，今回の自動推薦エージェントで用語を推薦できるオントロジーは，主に製品カテゴリなど既存の括りがオントロジーのクラスに相当する場合に限定される．これは，Web 上からブートストラッピングを用いて収集する手法の限界であり，いわば既に誰がカテゴライズしたものでなければ，1 つの分類として集めることができない．また，3 つのインスタンスだけではクラスが特定できないもの (3 つのインスタンスが他のクラスにも出てくるもの) はうまく括りだすことができない．例えば，{ 東京，品川，新宿 } だけでは，東京の都市クラスなのか，山手線の駅名クラスなのか，自動的に判断することは難しい．ただ，プロパティ抽出の際に行ったように，代表する概念 (この場合，“都市名”，“駅名”) をページ検索時に陽に指定することで，ある程度，クラスを特定することはできるだろう．今後，ブートストラッピングによるパターン抽出処理と併せて改良を検討していきたい．

また，他にアプローチとして挙げた A. 導入準備の容易化，B. 使い勝手の向上に関しては，2 章で述べたようにブラウザベースでインストールを不要としつつも，Flex を用いて右クリックのようなデスクトップアプリケーションの操作性を実現した．また，Spring Graph を用いてオントロジー全体の可視性とクラス間，クラス・インスタンス間の関係把握の両立を実現している．更に，D. モチベーションの維持に関しては，サンプルアプリケーションの提供やオントロジー利用 API の公開によってユーザを動機づけることを試みている．E. 共同作業の支援に関しては，DB を排他制御することで複数人によるオントロジー構築の同時作業を実現している．尚，本来の目的とは異なるが，ブラウザベースとすることでアプリケーション本体がサーバに置かれることになり，システムのアップデートやメンテナンスがより簡単に行えるようにもなった．また，Flex によるバックグラウンドでの非同期通信の仕組みを利用することで，頻繁に更新されるオントロジーを構築する際，Ajax のように Web ページを切り替えずに動的に最新情報を更新することが可能になった．但し，これらの点に関しては，一般的にもマニュアルやヘルプを工夫する，目新しい UI(奥行きを持たせ 3 次元化する，など) を付ける，ユーザの情報登録にポイントを付与する等々，さまざまなアプローチが存在する．今後，種々のアプローチを比較し，効果の検証方法を検

討していきたい．

6. 関 連 研 究

本章では，関連するオントロジー構築ツールと固有名詞抽出手法との比較について述べる．まず，代表的なオントロジー構築ツールとして Protégé と法造，KiWi を挙げる．

Protégé[Protégé] は，最もよく使われているオントロジーエディタの 1 つである．10 年以上に渡って研究者を中心に広く利用されており，推論処理などを高度な機能を備えている点に特徴がある．また，プラグイン仕様が公開されており，現在，公式サイトでは 72 のプラグインが登録され，RDB からのデータインポート，テキストマイニングによる語彙獲得などを追加することができる．法造 [法造] は，ロール概念を明示的に扱えるなどの特徴を持つツールである．また，分散開発環境を備えており，編集中のオントロジー間の差分を調べて，データの整合性を保つことができる．但し，これらはオントロジー研究者向けのヘビーウェイトオントロジー構築ツールであり，今回提案の ONTOMO とはターゲットユーザと解決したい問題点が異なっている．ただ，オントロジー可視化用プラグインや，分散データの整合性を保つ仕組みなどは今後，よく参考にしていきたい．

一方で，KiWi(Knowledge in A Wiki)[KiWi] は，ライトウェイトオントロジーの構築に焦点を当て，従来の Wiki にセマンティック Web の技術を加えることで拡張したものである (セマンティック Wiki とも呼ばれる)．特徴は，Wiki と同等のユーザインタフェースを通してコンテンツやメタデータを編集できるため，ユーザはオントロジーを意識することなく，意味データを登録できる点にある．これは，我々とは方法が異なるが，同様のターゲットユーザに向けて導入準備を容易にし，使い勝手を向上させている点，また共同作業を支援している点で共通している．今後，Wiki をベースとした履歴管理と差分機能は ONTOMO にも取り入れていきたい．

いずれの研究もインスタンスやプロパティを自動推薦する機能は有しておらず，この点でも ONTOMO と差異化できると考えられるが，本機能に関して代表的な構造化文書に対する固有名詞抽出手法である DIPRE と Snowball との比較を述べる．

DIPRE[Brin 98] は Web ページを対象としたブートストラッピングの先駆け的な研究である．まず，著者名と書籍名の組のリストをいくつか用意し，それらを検索エンジンで検索する．検索結果に対して，“URL，prefix 文字列，書籍名，middle 文字列，著者名，suffix 文字列” のような正規表現を見付けると，今度はその正規表現にマッチした著者名と書籍名の組を追加する．Snowball[Agichtein 00] は DIPRE の拡張であり，DIPRE が文字列のペアを抽出するのに対して，Snowball は固有表現に関する抽出パターンを導入し，これを抽出することを試みている．これに対し我々は，DIPRE で提案されたブートストラッピングをベースに，特定の集合に属する用語 (インスタンス，プロパティ) の抽出を試みた．また，抽出した結果を分析してパターン数に閾値を設定する他，オントロジーのクラス構造を参照し，関係が低いと思われる用語を除去する適合化フィルタを導入した点に特徴がある．今後は，前述した Google Sets や SEAL とうまく条件を合わせて性能比較などを検討していきたい．

7. お わ り に

本論では，オントロジーに詳しくないユーザが比較的簡単にオントロジー構築可能な環境の提供を目的に ONTOMO の開発について述べた．特に，従来のオントロジー研究者向けツールの問題点の内，用語登録の支援に焦点をおき，ONTOMO の概要を述べた後，ブートストラッピングに基づく固有名詞抽出手法を用いたインスタンス・プロパティ自動推薦エージェントについて詳述した．我々は評価実験の結果から，本エージェントによって一定程度，ユーザの負担を減らすことができるようになったと考えている．

今後の課題としては，Wiki のような履歴管理と差分機能を導入したいと考えている．変更履歴を管理することにより，過去データをロールバックできるようになる．また，差分機能によって複数ユーザが共同でオントロジーを構築する際に発生したデータの競合を検出し，整合性を保つための解決方法をユーザに提示，差分同期を行うことができるだろう．また，オントロジーの自動構築についても検討していきたい．Wikipedia のようにある程度，情報の信頼性が高いサイトは，オントロジーの情報源として適していると考えられる．既に，Wikipedia の一部情報を取得してオントロジーを構築する研究が行われているが，今後は如何に多様な情報源から精度よく情報を抽出するかが課題となるだろう．今後も，インターネットサービスの利用者または開発者が容易に使えるツールを目指し，広くユーザの意見を反映させて行きたい．

◇ 参 考 文 献 ◇

[Agichtein 00] E. Agichtein, L. Gravano: “Snowball: Extracting relations from large plain-text collections”, Proc. of 5th ACM/IEEE-CS Joint Conf. on Digital Libraries (JCDL 2000), pp.85-94, 2000.

[Brin 98] S. Brin: “Extracting patterns and relations from the world wide web”, WebDB Workshop at 6th Int. Conf. on Extended Database Technology, pp.172-183, 1998.

[Faltings 07] B. Faltings, V. Schickel-Zuber: “Oss: A semantic similarity function based on hierarchical ontologies”, Proc. of 20th Int. Joint Conf. on Artifical Intelligence (IJCAI 2007), pp.551-556, 2007.

[FLEX] Adobe Flex, http://www.adobe.com/jp/products/flex/.

[Google Sets] Google Sets, http://labs.google.com/sets.

[法造] 法造～オントロジー構築・利用の研究サイト～, http://www.hozo.jp/hozo/.

[JENA] Jena - A Semantic Web Framework for Java, http://jena.sourceforge.net/.

[KiWi]　KiWi - Knowledge In A Wiki, http://www.kiwi-project.eu/.
[OWL]　OWL Web Ontology Language, http://www.w3.org/TR/owl-features/.
[Protégé]　The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/.
[RANK]　WORLD RANKING OF MANUFACTURERS YEAR 2007, http://oica.net/wp-content/uploads/world-ranking-2007.pdf.
[SEAL]　Boo!Wa! - A List Extractor for Many Languages (former Seal), http://boowa.com/.
[Spring Graph]　Spring Graph, http://mark-shepherd.com/blog/springgraph-flex-component/.

〔担当委員：山下 倫央〕

2009 年 12 月 28 日　受理

著　者　紹　介

川村　隆浩（正会員）

1992 年早稲田大学理工学部電気工学科卒業．1994 年同大大学院理工学研究科電気工学専攻 修士課程了．同年，(株) 東芝入社．現在，同社研究開発センター 主任研究員．工学博士．2001–2002 年米国カーネギ・メロン大学 ロボット工学研究所 客員研究員．2003 年より電気通信大学大学院情報システム学研究科 客員准教授．2007 年より大阪大学大学院 工学研究科 非常勤講師．主としてマルチエージェントシステム，セマンティック Web の研究・開発に従事．情報処理学会会員．

沈　　偉

2010 年電気通信大学院情報システム学研究科社会知能情報学専攻 修士課程了．同年，NTT データ株式会社入社．

中川　博之

1997 年大阪大学基礎工学部情報工学科卒．同年，鹿島建設 (株) 入社．2007 年東京大学大学院情報理工学系研究科修士課程了，2008 年同大学院博士課程中退．同年より電気通信大学 助教．エージェントおよび自己適応システム開発手法の研究に従事．情報処理学会，電子情報通信学会，IEEE CS 各会員．

田原　康之

1991 年東京大学大学院理学系研究科数学専攻 修士課程了．同年，(株) 東芝入社．1993–1996 年情報処理振興事業協会出向．1996–1997 年英国 City 大学 客員研究員．1997–1998 年英国 Imperial College 客員研究員．2003 年国立情報学研究所入所．2008 年より電気通信大学 准教授．博士 (情報科学)(早稲田大学)．エージェント技術，およびソフトウェア工学などの研究に従事．情報処理学会，日本ソフトウェア科学会各会員．

大須賀　昭彦（正会員）

1981 年上智大学理工学部数学科卒．同年，(株) 東芝入社．同社研究開発センター，ソフトウェア技術センターなどに所属．1985–1989 年 (財) 新世代コンピュータ技術開発機構 (ICOT) 出向．2007 年より電気通信大学大学院情報システム学研究科 教授．工学博士 (早稲田大学)．主としてソフトウェアのためのフォーマルメソッド，エージェント技術の研究に従事．1986 年度情報処理学会論文賞受賞．情報処理学会，電子情報通信学会，日本ソフトウェア科学会，IEEE CS 各会員．