

# 「法造」によるオントロジーの分散共同構築支援

## Supporting for Distributed and Collaborative Construction of Ontologies Using Hozo

古崎 晃司<sup>1)</sup>, 砂川 英一<sup>1)</sup>, 太田 衛<sup>2)</sup>, 來村 徳信<sup>1)</sup>, 溝口 理一郎<sup>1)</sup>  
Kouji Kozaki<sup>1)</sup>, Eiichi Sunagawa<sup>1)</sup>, Mamoru Ohta, Yoshinobu Kitamura<sup>1)</sup>, Riichiro Mizoguchi<sup>1)</sup>

<sup>1)</sup> 大阪大学産業科学研究研究所, <sup>2)</sup> 株式会社エネゲート  
<sup>1)</sup> I.S.I.R., Osaka-University, <sup>2)</sup> ENEGATE co., ltd

**要旨:** オントロジーの大規模化に伴う構築のコストを軽減する為には, 一つのオントロジーを複数の構築者で協力して開発する分散共同構築が必要とされる. 筆者らが開発を進めているオントロジー構築・利用環境「法造」では, 1)一つのオントロジーを複数に分割し, 相互の依存関係を保持しつつ個別かつ並列的に開発する, 2)一つのオントロジーを複数の構築者で共有し, アクセス管理の元で共同構築を行う, という2種類の方法によるオントロジーの分散共同構築支援機能を提供する. 本発表では, それらの支援機能について述べる.

### 1. はじめに

近年, オントロジーの利用は, 単一のオントロジーを固定的に用いる流れに加え, 複数の開発者で分担・協力しながら開発した大規模オントロジーを, 適宜修正しながら運用する流れが活発になりつつある. 特に Semantic Web での利用を想定したオントロジーの研究では, Web 本来の分散的性質を反映し, 分散したオントロジーを状況に応じてマージング/マッピングする技術や, 流動的にオントロジーを変更する技術が注目されている. こうした変化の理由の1つとして, オントロジー工学が, 基礎研究から応用研究へと活動の幅を広げたことがあげられる. 実利用に耐え得るオントロジーを開発しようとする, 大規模オントロジーへの対応, 構築や運用の過程で明らかになった問題に対する修正や改訂, 整合性保持などのメンテナンス, 既存オントロジーの再利用など, 一般のソフトウェア開発と似た課題に直面するようになる. このようなオントロジー開発を取り巻く状況の変化に際して, オントロジーの開発環境も, それに対応するような新たな枠組みを提供していくことが求められる.

本論文では, 筆者らが開発を進めてきたオントロジー構築利用環境「法造」における, オントロジーの分散共同構築を支援する機能について述べる. 「法造」では, 1)一つのオントロジーを複数に分割し, 相互の依存関係を保持しつつ個別かつ並列的に開発する, 2)一つのオントロジーを複数の構築者で共有し, アクセス管理の元で共同構築を行う, という2種類の方法によるオントロジーの分散共同構築支援機能を提供する. これらの支援機能は, 理論的側面よりも, 実際のオントロジー構築の場で利用できる実用性を重視して

開発を進めており, 既にプロトタイプシステムが完成し試用を開始している.

以下, 2章では本研究で対象としているオントロジーの分散共同開発について述べる. 続く3章および4章では, 「法造」による分散共同開発支援の流れと各過程における支援機能について述べる. そして5章では関連研究との比較を行い, 6章で本論文の総括を行う.

### 2. オントロジーの分散開発

大規模なオントロジーを構築する際には, 最終目標とするオントロジーを複数の部分オントロジーに分割し, 各部分オントロジーの構築作業を別々に進める場合がある. 時として, それらの構築作業は別々の構築者による並行作業で行われる. そのような状況下においては, ある部分オントロジーが他の部分オントロジーで定義された概念をインポートし, その概念を利用して別の概念を定義することがある. 図1は, あるオントロジーBが別のオントロジーAで定義された概念をインポートしている様子を模式的に示したものである. このとき, オントロジーBはオントロ

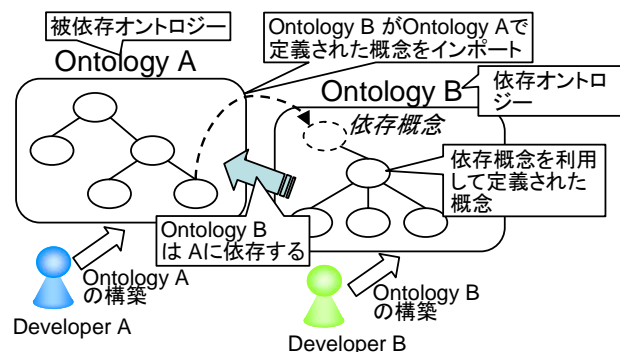


図1 オントロジーの分散開発

ジーA の定義を利用していることから、オントロジーA に依存することを意味する。このように他のオントロジーに依存しているオントロジー（この例ではオントロジーB）を**依存オントロジー**、依存オントロジーがインポートした概念を**依存概念**、依存概念を定義しているオントロジー（ここでは A）を**被依存オントロジー**と呼ぶ。そして、このようにオントロジーを複数のオントロジーに分割し、依存概念を適時利用しながらオントロジー全体を構築する開発方法を、本研究では**オントロジーの分散開発**と呼ぶ。

前述のようにオントロジーの分散開発においては、構築者は最終目標となるオントロジーを複数の部分オントロジーに分割し、それぞれの部分オントロジーの構築を別々の構築者が担当し、共同作業でオントロジーの構築を進める。その際、可能であれば、他の構築者が別のオントロジーで定義した概念を依存概念としてインポートし、各自が担当するオントロジーの構築に再利用する。しかし、構築途中のオントロジーを被依存オントロジーとした場合、その被依存オントロジーの更新に伴い、依存概念を介したオントロジーの整合性が崩れる可能性がある。例えば、依存オントロジーで依存概念の下位概念を定義していたとき、被依存オントロジーの更新の伴いその依存概念の元の定義が消去されたとすると、その影響で依存オントロジーに不整合が生じる。

このような問題を避けるため、オントロジーの分散開発を支援するシステムには、依存／被依存オントロジー間の関係を適切に管理し、各オントロジーの構築者の共同作業を支援する機能が必要とされる。このような背景の下、筆者らは被依存オントロジーが変更された際に、依存概念を介して、依存オントロジーにどのように影響を与えるかの考察し、それぞれの影響に対してオントロジーの整合性を保持するための対処方法を検討し 71 通りの適用可能な対応方法としてまとめた [1,2,3]。本論文では、それらの考察結果に基づき開発した、オントロジー分散構築支援機能について述べる。

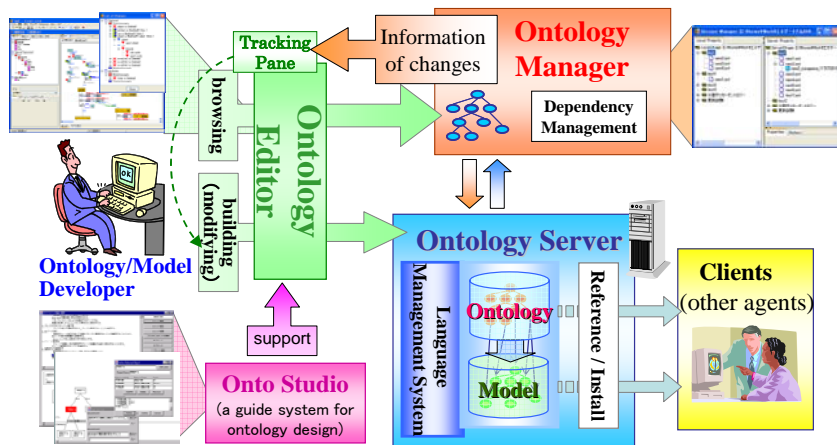


図2 オントロジー構築利用環境「法造」

### 3. 「法造」によるオントロジーの分散共同構築支援

#### 3.1 オントロジー構築利用環境「法造」

オントロジー構築利用環境「法造」は、筆者らが開発を進めているオントロジー (= “法”) を構築する (= “造”) ための計算機環境で、オントロジーをグラフィカルに表示・編集する **Ontology Editor**、ガイドラインに従いオントロジーを構築支援する概念工房 (**OntoStudio**)、構築したオントロジーを管理する **Ontology Server**、複数のオントロジー間の依存関係を管理する **Ontology Manager** の 4 つのシステムから構成される (図 2)。法造が用いるオントロジー記述言語は XML ベースのフレーム型言語であるが、RDF(S),OWL 形式でのエクスポート・インポート機能<sup>1</sup>を持つ。

法造の特徴は、1) ロール概念を中心としたオントロジー工学の基礎理論に基づいて設計されている [4,5,6]、2) 独自のグラフィカル表現を用いたオントロジー表示・編集機能、そして本稿で述べる 3) オントロジー間の依存関係管理に基づく分散開発支援機能にある。なお、「法造」の **Ontology Editor** は Web サイト<sup>2</sup>上においてフリーソフトウェアとして公開しており、国内外を含めて約 1000 名<sup>3</sup>がユーザ登録をしている。

<sup>1</sup> エクスポートされる OWL のフォーマットについては [6] を参照。また OWL のインポート機能に関しては、現状では subClassOf 関係など OWL 形式の一部の要素に限定されている。

<sup>2</sup> <http://www.hozo.jp/>

<sup>3</sup> 2007 年 1 月現在

### 3.2 オントロジー分散共同構築の流れ

ここでは、本研究が対象とする2種類のオントロジー分散共同構築の形態について、構築作業の概要を述べる。

#### 3.2.1 複数オントロジーの分散構築

図3に互いに依存関係を持つ複数のオントロジーを複数の構築者で分担し、並列的に開発する枠組みを示す。システムはサーバ上の「オントロジー共有スペース(Shared Space)」とクライアントシステム上の「個人編集スペース(Personal Space)」を構築者に提供する。オントロジー共有スペースでは構築されたオントロジーが全構築者で共有・管理される。各構築者はサーバ上の共有スペースに保存されたオントロジーを直接編集することはできず、各自の個人編集スペースに必要なオントロジーをダウンロードしてクライアント上で編集を行う。そして、編集したオントロジーをサーバにアップロードすることで、他の構築者と共有する。その際、必要に応じて共有スペース上のオントロジーのバージョン管理とアクセス管理が行われる。

なお、共有スペースの複数のオントロジー全体が構築目標とするオントロジーとなり、各構築者はそれぞれが担当するオントロジーを他のオントロジーで定義された概念を適時インポートしながら構築する。すなわち、それぞれのオントロジーは、構築者毎に各自が構築を担当するオントロジーと参照するオントロジーに分かれる。

以下に、このような構築形態を実施する際の流れを示す。

- 1) **Ontology Manager** を用いて共有スペース上のオントロジーの最新情報を取得し、必要なオントロジーを個人編集スペースにダウンロードする。その際に必要に応じてこれから編集するオントロジーを他の構築者が更新(アップロード)できないようにロックする。
- 2) 前回の編集時以降に更新されたオントロジーを確認し、**Ontology Manager** の **Tracking Pane** を利用して、変更箇所が自分の編集するオントロジーに影響する範囲を調べる。
- 3) 変更箇所の影響が自分の編集するオントロジーに不整合をもたらす際には、不整合を解消し整合性を保持する為の修正を行う。その際にシステムは、適用可能な対応方法の案を提示する。

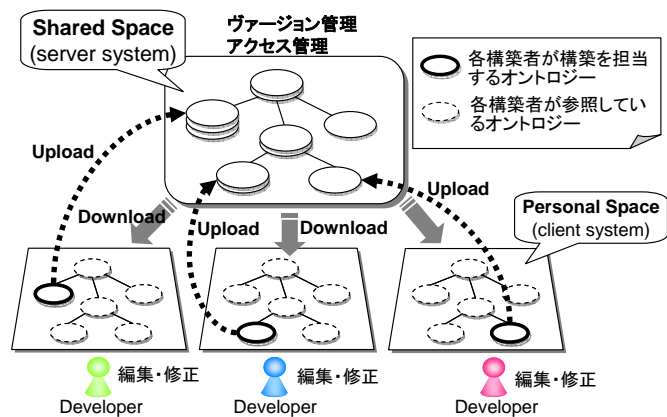


図3. 複数オントロジーの分散開発

- 4) 整合性保持のために修正を終えた後、オントロジーの編集作業を行う。編集時に、必要に応じて共有している他のオントロジーで定義されている概念をインポートすることが出来る。その際、オントロジー間の依存関係は **Ontology Manager** によって管理される。
- 5) 編集作業の終了後、編集したオントロジーを共有スペースにアップロードし、他の構築者と共有する。その際、他の構築者に更新を許可する場合は、オントロジーのロックを解除する。

上記の過程を複数の構築者で並列的に繰り返すことによって、最終的な構築目標とするオントロジー全体が、サーバの共有スペース上に構築される。

#### 3.2.2 単一オントロジーの共同構築

単一のオントロジーを複数の構築者で共有し、共同作業で構築を行う。この場合、適切なアクセス管理の下、同時に編集・更新を行わないよう注意する必要がある。以下にこのような構築形態を行う際の流れを示す。

- 1) サーバ上の共有スペースで構築対象とするオントロジーを複数の構築者で共有する。オントロジーのバージョンは、**Ontology Server** によって管理される。
- 2) オントロジーを編集するには、他の構築者が同時に更新することを避けるためオントロジーをロックした後、個人編集スペースにダウンロードする。
- 3) 以前の編集以降に、他の構築者によりオントロジーが更新されていた場合は、**Tracking Pane** を用いて古いバージョンとの差分を

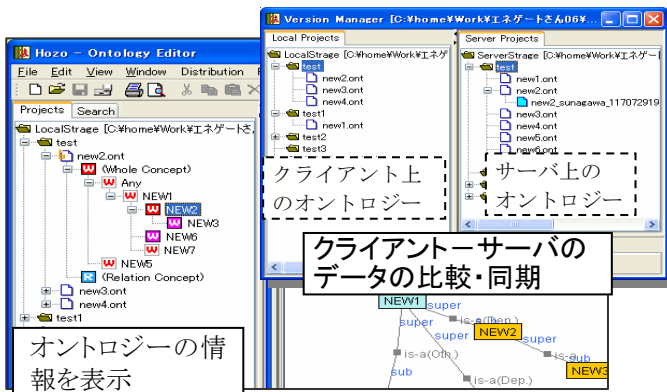


図4 Ontology Manger

比較することにより、更新箇所の影響範囲を確認する。必要ならば、更新箇所を元に戻す。

- 4) 変更箇所の確認後、オントロジーの編集を行う。編集が終わったら、オントロジーを共有スペースにアップロードし、ロックを解除して他の構築者が更新可能にする。

## 4. 「法造」の分散構築支援機能

ここでは、前章で述べたオントロジーの分散共同構築を支援するために「法造」が提供する機能について述べる。

### 4.1 オントロジー間の依存関係管理

#### 4.2.1 Ontology Manager

Ontology manager(図4)はOntology Server上の共有スペースとクライアントシステム上の個人編集スペースをつなぐインタフェースと、以下のような機能を提供する。

- 1) 共有スペース上のオントロジーの最新情報(更新済み、自分がロック中、他ユーザによるロック中、など)の表示
- 2) 共有スペース上のオントロジーへのアクセス管理(オントロジーのロックおよびロック解除)
- 3) 共有スペースおよび個人編集スペース上のオントロジーのバージョン管理
- 4) 他のオントロジーで定義された概念の検索
- 5) 共有スペースと個人編集スペースのデータ同期

#### 4.2.2 依存概念のインポート

2章で述べたように、他のユーザが構築しているオントロジーで再利用可能な概念が定義されていた場合、その概念を依存概念として自分が編集しているオントロジーにインポートすることができる。このインポート操作は Ontology Manager の Import Dialog というウィンドウを

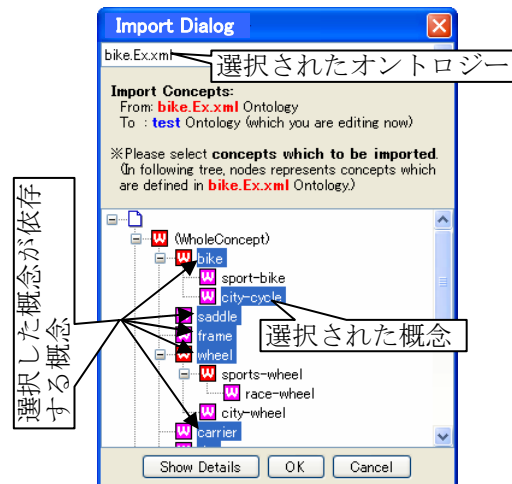


図5 Import Dialog

用いて行う(図5)。Import Dialog は選択したオントロジーで定義された概念の is-a 階層を表示し、ユーザはその階層からインポートする概念を選択する。するとシステムが選択された概念が依存しているすべての概念<sup>4</sup>を検索し、それらの概念と共に選択された概念のインポートを実行する。インポートされた依存概念が Ontology Editor 上では通常の色と異なる色で表示され、オントロジーの一貫性を保つためユーザは依存概念の定義を直接編集<sup>5</sup>することはできない。

### 4.2 概念依存関係管理に基づく整合性保持

#### 4.2.1 被依存オントロジーの変更箇所の確認

Ontology Manager はユーザにオントロジーの更新情報を通知する。あるオントロジーが更新されていたとき、それと依存関係を持つオントロジーの整合性を保持するためには、“どの概念やスロットが変更されたのか? ”、“その変更に伴い影響を受ける概念はどれか?”といったより詳細な情報が必要となる。法造では、そのような情報を Ontology Editor の Tracking Pane と Browsing Pane に表示する。

Tracking Pane は編集中のオントロジー内の依存概念のうち定義元である被依存オントロジー内で変更された箇所の一覧と、ユーザが編集中のオントロジーでその変更が影響する範囲を表示する(図6)。これらの変更箇所はツリー状に表示され、各ノードのアイコンは消去、修正、追加

<sup>4</sup> 例えば、選択した概念の上位概念や、その概念定義で参照している概念。

<sup>5</sup> 依存概念の下位概念を定義するなど、依存概念を用いて他の概念を定義することはできる。



図6 Tracking Pane による変更箇所の表示

といった変更の種類を表す。さらに各ノードにつながる下位ツリーのノードは、その変更が影響を及ぼす概念の一覧を示す。それらのノードをクリックすることで、詳細な定義や変更内容が Browsing Pane に表示される。

Browsing Pane (図7) では、オントロジーがネットワーク構造を用いて表される。依存概念の変更箇所は、Tracking Pane と同じ3種類のアイコンで示される。さらに変更の種類が「修正」である箇所をマウスでクリックすると、修正内容の詳細が表示される。

#### 4.2.2 整合性保持のための編集支援

オントロジー間の依存関係に基づく整合性を保持するため、法造はユーザに適用可能な対応方法を提示する。提示される対応方法は、オントロジー間の依存関係と変更の種類に基づいて考察されたものである[1,2,3]。

最初に「法造」は、“**変更に合意する**”または“**変更を拒否する**”というもっとも基本的な対応方法を提示する。前者は編集中のオントロジー内の依存概念を変更後の新しい定義のものと置き換えることを意味し、後者は変更された依存概念をインポートすることを止め変更前の依存概念と同じ定義内容の概念を新たに編集中のオントロジー内で再定義することに相当する。例えば、“変更に合意する”は、変更の種類が「修正」の場合は“依存概念を修正された新しい概念と置き換える”ことに相当し、変更の種類が「消去」の場合は“依存概念を消去する”ことに相当する。ユーザはこれらの対応方法を変更された概念をクリックした際に表示されるポップアップメニ

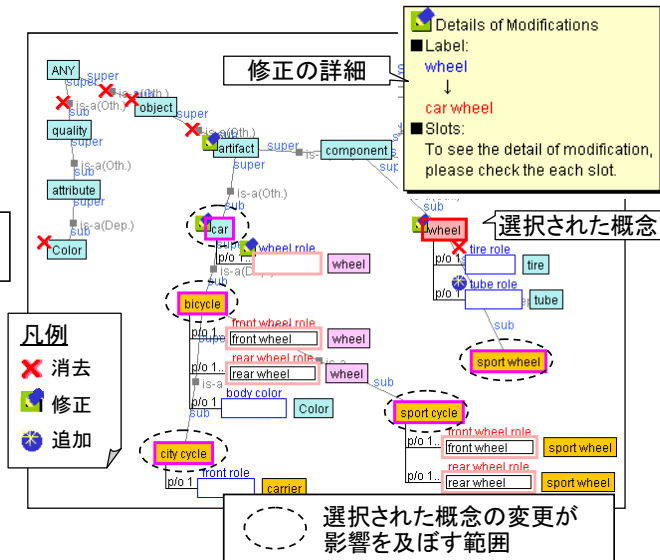


図7 Browsing Pane による変更箇所の表示

ューから選択することで実行する。

変更点に対してこれらの対応方法を適用した後、ユーザの必要に応じてオントロジーの編集を進める。その際には、変更箇所の影響範囲を表示する機能を利用することができる。また、より複雑な対応方法<sup>6</sup>をユーザに提示する Harmonizing Pane も利用することができる。

#### 4.3 ヴァージョン管理に基づく整合性保持

Ontology Server は共有スペース上にアップロードされたすべてのオントロジーの履歴を管理している。3.2.2 項で述べた単一オントロジーの共同構築を行う際には、Ontology Manager のアクセス管理(ロック)機能を用いて複数ユーザが同時に一つのオントロジーを更新することを避けると共に、最新のオントロジーを古いバージョンと比較して、更新内容を確認することが必要とされる。4.2 節で述べた Ontology Manager の変更箇所の確認・整合性保持のための編集支援機能は、このような同一オントロジーの旧バージョンとの差分比較にも利用することができる。

4.2 節で述べた被依存オントロジーの変更箇所の確認は、編集中のオントロジーが依存している旧バージョンの被依存オントロジーが最新版のオントロジーでどう変更されているかを確認している。それに対し、本節で述べている差分比較は、最新版のオントロジーを旧バージョンと

<sup>6</sup> 現在、[1]で議論されている対応方法のうち、一部についてのみ実装している。しかし、“変更に合意する”または“変更を拒否する”という基本的な2つの対応方法で、必要な修正の大部分が行えると思われる。

比較している点異なるが、変更箇所の表示や編集支援の方法は同様である<sup>7</sup>。

## 5. 関連研究

PromptDiff はバージョンの比較による変更箇所の検出と変更内容の同定を行い、変更の承諾／拒絶を選択肢として提供することで、整合性あるオントロジーの改訂に対する支援を行うオントロジーのバージョン管理ツールである [7]。そのアプローチは我々の枠組みと同様であるが、法造の特徴である複数オントロジーの分散構築 (3.2.1 項参照) はサポートされていない。

DILIGENT [8] , ONKI [9] は法造と同様にサーバ上の共有スペースを用いたオントロジーの分散構築をサポートしている。しかし、整合性保持のための対応方法を提示するといった支援機能は持たず、ユーザ間の協調作業を支援する機能を中心としている。

KAON は本研究と同様に、あるオントロジーで生じた変更が別のオントロジーに影響して不整合を引き起こす事に注目し、その対処法として *evolution strategies* を適用したオントロジー開発を提案している [10]。しかし、KAON の *evolution strategies* は、変更されたオントロジーを包含するオントロジーに適用されるものであるため、その変更を拒否するという対応がサポートされていない。この違いは、オントロジー間の依存関係をとらえる観点の相違に起因していると思われる。

## 6. まとめ

本論文では、法造のオントロジー分散共同開発支援機能について述べた。オントロジー間の整合性の保持は、分散構築を行う際には特に重要な課題である。本システムはオントロジー間の概念依存関係管理に基づき、その問題の解決に貢献する。また同じ枠組みを、単一オントロジーの共同構築支援にも適用することができる。現在、プロトタイプシステムの試用を進めており、大規模なオントロジー構築における本格的な利用も予定されている。

今後の課題としては、(1)本システムの枠組みに基づくオントロジーとインスタンスモデル間の整合性保持、(2)OWL 形式のオントロジーへの本枠組みの適用、(3)OWL import を用いた他のサ

ーバ上のオントロジーの参照を扱う機能、などを検討している。

## 参考文献

- [1] 砂川英一, 古崎晃司, 來村徳信, 溝口理一郎: 法造」における オントロジー分散開発, 第3回人工知能学会セマンティックウェブとオントロジー研究会, SIG-SW&ONT-A301-02, 2003
- [2] E. Sunagawa, et al.: An Environment for Distributed Ontology Development Based on Dependency Management, Proc. of ISWC2003, pp. 453-468, 2003.
- [3] 砂川英一: 分散開発フレームワークとロール概念組織化理論に基づくオントロジー構築支援環境に関する研究, 博士学位論文, 2007
- [4] Kozaki K., et al.: Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship", Proc. of EKAW2002, pp.213-218, Siguenza, Spain, 2002
- [5] Sunagawa, E., et al. :Organizing Role-concepts in Ontology Development Environment: Hozo, Proc. of 2005 AAAI Fall Symposium on Roles, an interdisciplinary perspective, 2005
- [6] Kozaki K., et al.: Fundamental Consideration of Role Concepts for Ontology Evaluation, Proc. of EON2006 Edinburgh, United Kingdom, May 22, 2006
- [7] Noy, N., et al.: Tracking Changes during Ontology Evolution, Proc. of ISWC2004, Hiroshima, Japan, pp.259-273, 2004
- [8] Tempich, C., et al.: An Argumentation Ontology for DIstributed, Loosely-controlled and evolvinG Engineering processes of onTologies (DILIGENT), Proc. of ESWC2005, Greece, pp. 241-256, 2005
- [9] Valo, A., Hyvonen, E. and Komurainen, V.: A Tool for Collaborative Ontology Development for the Semantic Web, in: Proc. of DC 2005, Madrid, Spain, 2005.
- [10] Stojanovic, L., Maedche, A., Motik, B. and Stojanovic, N., User-driven Ontology Evolution Management, Proc. of EKAW 2002, Madrid, Spain, pp. 285-300, 2002

<sup>7</sup> 旧バージョンの概念が最新版で削除されている場合の表示方法 (半透明のノードと「消去」を示すアイコンで表される) など, 比較する方向の違いにより表示され得る内容が若干異なる。