人工知能学会研究会資料
SIG-SW&ONT-A301-02

「法造」における オントロジー分散開発
Distributed Ontology Development with “Hozo”

砂川 英一, 古崎 晃司, 來村 徳信, 溝口 理一郎
Eiichi Sunagawa, Kouji Kozaki, Yoshinobu Kitamura, Riichiro Mizoguchi

大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research,
Osaka University

Abstract. This paper describes a method for distributed ontology development and its implementation. In addition to the discussion on general distributed development, we especially focus on how to keep the consistency of sub-ontologies as a whole. Its method is based on the management of the dependency between concepts definitions in the sub-ontologies. Aiming at supporting of distributed ontology development comprehensively, we design and implement a prototype system in “Hozo” which has been developed by the authors as a support system for constructing and using ontology.

1. はじめに

Semantic Web の提唱により，かねてより知識システムの基盤として位置付けられていたオントロジーに，新しい具体的な応用の場が開かれた. 特に Web の分散的性質とも相まって，近年のオントロジー研究では，大規模で半永久に固定可能なオントロジーの研究のみならず，小規模で分散したオントロジーを状況に応じてマージしたり，マッピングしたりする技術の研究が盛んに行われるようになっている．本研究の目的はオントロジーの分散開発を支援することにあり，これも上記の流れに沿ったものであると言える.

本研究では，単一のオントロジーは複数の部分的なオントロジーから構成されると捉え，それら部分となるオントロジーを個別に構築することにより，目的となるオントロジー全体を完成させるという「オントロジー分散開発」の方法を検討している．そして，その構築過程を包括的に支援する計算機環境を整えることが本研究の目標である[Sunagawa 03]. オントロジーの分散構築は，Semantic Web のみならず，一般の知識システム構築においても有効であり，オントロジー構築過程における様々な段階や状況に適用可能である.

連絡先：砂川英一, 大阪大学 産業科学研究所 知識システム分野, 〒567-0047 大阪府茨木市美穂ヶ丘8-1, Tel: 06-6879-8416 , Fax: 06-6879-2123 , e-mail: sunagawa@ei.sanken.osaka-u.ac.jp

以下，まず２章でオントロジーの分散開発が意味するところを説明する．３，４章では，分散構築に見られる諸問題とその解決アプローチを扱う．特に 3 章では，本研究の中心的課題として，分散構築における整合性保持の問題と，依存関係管理による解決の枠組みについて詳しく述べる．５章では，それまで検討した内容に基づいて実装されたシステムを紹介する．6 章では関連研究との差異を述べる．最後に７章で本論文の総括的まとめを行い，今後の課題を検討する.

2. オントロジーの分散開発

一般にオントロジーは複数の部分的なオントロジーに分割することが可能であり，また逆に，複数のオントロジーを統合して単一のオントロジーとみなすことも可能である．この時，部分となるオントロジーは，含まれる概念のレベルや種類によって認識される．その実例として，旧通産省下のヒューマンメディア・プロジェクトで構築されたプラントオントロジー[Kozaki 02]を図１に示す．このオントロジーは，トップレベルオントロジーの下位で大きくタスク概念とドメイン概念のオントロジーに分割できる．さらにドメイン概念のオントロジーは物理属性概念のオントロジーと，対象物・部品・機能概念のオントロジーに詳細化されるデバイスオントロジーに分割できる，図の矢印はその方向でオントロ

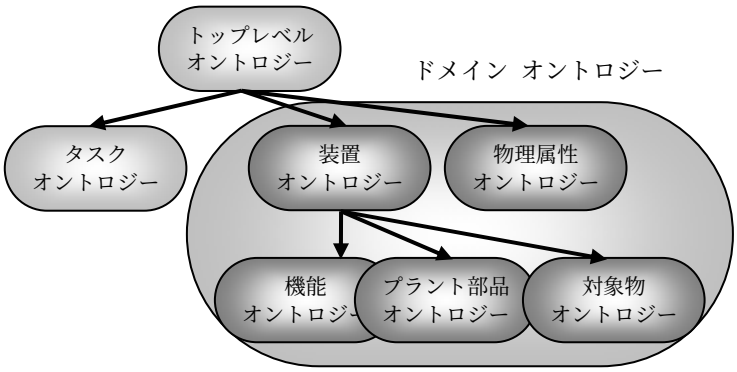

図１．プラントオントロジーの全体像

ジーに含まれる概念が詳細化されていることを示しており，後述の継承関係にあたる．

このような部分的なオントロジーは，その開発前から認識することが可能である．例えばオントロジーを構築する際，既存のトップレベルオントロジーで定義されている概念を利用し，その下に専門分野固有の概念を補足的に付け加えていく事は少なくない．このとき，構築されるオントロジー全体は，既存のオントロジーを再利用した部分と，新たに概念を追加した部分という２つのオントロジーの統合と見なすことができる．また，新たに追加される各部分のオントロジーも，さらに分割することが可能である．

このようにオントロジーの“部分”を意識し，それぞれを個別のオントロジーとして扱うことのメリットには，①概念体系の全体像を把握しやすくする ②協調構築の布石となる ③オントロジーの再利用性向上に貢献する，などがある．具体的には，次のようなオントロジーの分散開発を行うことが可能になる．

- ✧ 構築前に概念を大まかに分類し，それぞれを別のオントロジーとして複数の構築者による分担作業で構築する
- ✧ 構築途中で上と同様のオントロジーの分割を行う
- ✧ 構築するオントロジーの一部として，他のオントロジーから切り出したものを再利用する

続く３，４章では，こうしたオントロジー分散開発を行うに当たって問題となる点や，その解決アプローチを述べる．

3. 依存関係管理による整合性保持

分散構築を行うに当たって本研究が特に焦点を当てているのは，オントロジーの整合性に関する問題である．分散して存在する各オントロジーは，共通の概念体系を構成している“部分”であり，互いに影響を及ぼしあっている．よって，それらを個別に構築する際には，オントロジー間で不整合が生じやすくなると考えられる．これを解決するアプローチとして，我々はオントロジー間に存在する依存関係に着目し，依存関係の整合性を管理することによって，オントロジー全体の整合性を維持する方法を検討した[砂川 02].

3.1 オントロジー間の依存関係

一般に，オントロジーで概念を定義するときには，他の概念の定義内容を必要とする場合がほとんどである．上位概念からの定義の継承，部分概念に対するクラス制約などがそれに当たる．この時，それらの概念間には依存関係があるとみなすことができる．

異なるオントロジーに含まれる概念同士でこのような依存関係を持つとき，それをもとにオントロジー間の依存関係を捉えることが可能であり，本研究ではオントロジー間の依存関係として次の２種を定義した．

- ***継承関係***: 一方のオントロジーで定義される概念が他方に含まれる概念の定義を継承しているとき, この二つのオントロジー間における関係を継承関係と定義する．この際, 上位概念側のオントロジーを“上位オントロジー”，下位概念側のオントロジーを“下位オントロジー”と定義する．
- ***参照関係***: 一方のオントロジーで定義される概念の部分概念や属性概念に対するクラス制約として, 別のオントロジーに含まれる概念が使われている場合, この二つのオントロジー間には“参照関係”が成り立つとし，他のオントロジーの概念を使用している側を“参照オントロジー”，クラス制約として使用される概念を含む側を“被参照オントロジー”と定義する．

3.2 整合性保持のための依存関係管理

あるオントロジーの概念定義を別のオントロジーで使用した場合，その元になっている概念の定義が変化した際に，２つのオントロジーの関係に整合性がとれなくなり，構築し

ようとしているオントロジー全体に矛盾が生じる可能性がある

よって，他のオントロジーに影響を及ぼすような概念の定義に何らかの変更を加える際には，その依存関係が適切に保たれるように支援する必要がある．これには，変化するオントロジーの側でその変化の仕方に制限を加える方法と，変化の影響を受ける側で整合性を保つために追従的な変化を行う方法の２つが考えられる．本論文では後者の手法を主として扱う．

あるオントロジーが変化した際，その変化の影響を受ける側（つまり，依存している側）のオントロジーで整合性を保つために取ることのできる対応には以下の５通り考えられる．

1. 変化を受け入れ，それに合わせる形でオントロジーに追従的な変化を加える．追従方法は，変化の種類に依存する．
2. 変化を受け入れ，整合性に問題がない場合, 依存している側での変更は行わない．
3. 変化を拒否し，それを打ち消すようにオントロジーに追従的な変化を加える．追従方法は，変化の種類に依存する．
4. 変化を拒否し，前バージョンのオントロジーに依存し続ける．オントロジー間の依存関係は保持し，さらなる変化によっては再び最新のバージョンと依存関係を結ぶ余地を残す．
5. 変化を拒否し，依存していた概念を取り込むことによって, 依存関係を切り離す．その概念との関係は切り離されるので，４．と異なり依存関係を更新することは想定しておらず，以後の変化の影響は全く無視する．

４と５はあらゆる変化に対して共通に適応できるが，1～３の適応可能性，及び，その具体的な追従方法は変化の種類に依存する．そこで変化の種類を分類するために，まずは影響を与える概念の変化についてパターンを考える．これは依存関係の種類で大別され，さらに概念そのものに対する操作のレベル（概念レベル）と概念の定義内容に関する操作レベル（定義レベル）に分かれる．例えば，概念レベルの操作には「概念の消去」「下位概念の追加」があり，定義の変化には「スロット（部分概念や属性概念）の消去」などがある．

結果として，本研究では変化の種類を 17 パターンに分類した．そして，それぞれのパターンについて上記１～５の適応の可否やその詳細を考えたところ, 合計で 68 通りの追従方法があるという結果が得られた．

表１は，その一部を示したものであり，数字はその方法の数を示している．

表1．概念変化のパターンと，それに対する追従方法の数

変化の種類	追従方法				
	1	2	3	4	5
継承関係					
概念レベルの操作					
概念の消去	2	-	1	1	1
下位概念の追加	2	1	-	1	1
定義レベルの操作					
ラベルの変化	1	1	-	1	1
スロットの消去	1	-	1	1	1
・・・					
参照関係					
概念レベルの操作					
概念の消去	1	-	1	1	1
下位概念の追加	1	1	-	1	1
定義レベルの操作					
ラベルの変化	1	1	-	1	1
・・・					

3.3 変化に対する追従の例

ここでは，プラントオントロジー（図１）を用いて，オントロジーの変化と，それに対する整合性保持のための追従例を示す．

今，装置オントロジーのオーサが，概念“デバイス”の定義から*入力物*スロットを消去した場合を考える．これによって影響を受けるのは，この概念で継承関係を結んでいたプラント部品オントロジーである（図２）．概念単位で考察すると，プラント部品オントロジー内の概念“駆動系部品”や“熱交換部品”などが継承していた定義を失う，という問題が生じる．

この場合，整合性を保つためにプラント部品オントロジーのオーサへ提供される選択肢は，表１“継承関係－スロットの消去”に沿った次の４通りである．（表１）．

1. スロットの消去を受け入れ，影響を受けるプラント部品オントロジー内の全ての概念で，*入力物*スロットを消去する．
3. スロットの消去を拒否し，*入力物*スロットを，“駆動系部品”などで再び定義しなおす．

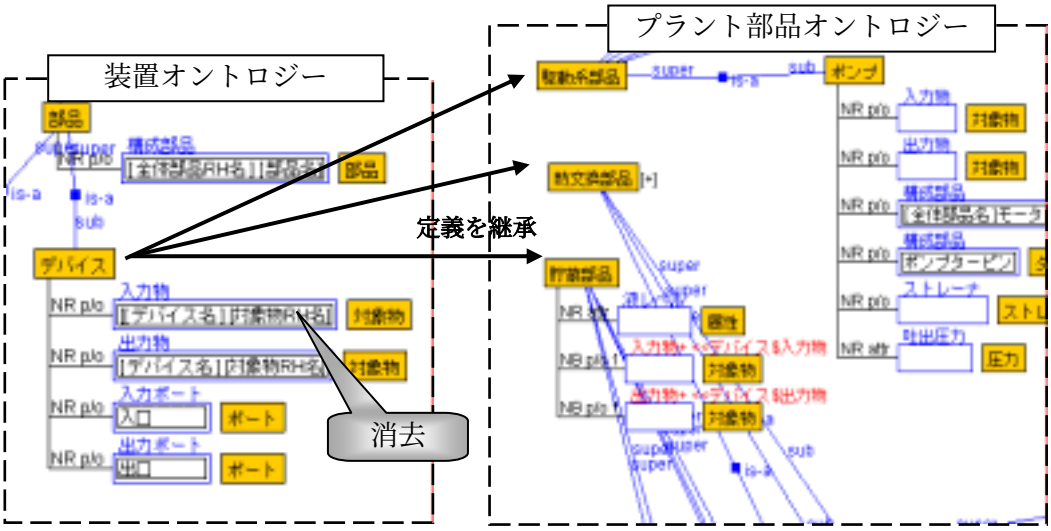

図２．継承関係における変化の例

4. スロットの消去を拒否し，システムが保持している変更前の装置オントロジーに依存し続ける．
5. スロットの消去を拒否し，*入力物*スロットを消去される前の“デバイス”概念を，プラント部品オントロジーに取り込む．

4. その他の問題点とその解決

本章では，依存関係に基づいた整合性保持問題以外に，オントロジーを構築するにあたって必要な機能を，構築される各オントロジーの管理，協調構築に向けてのオーサ管理，という２点から述べる．

4.1 オントロジー管理

バージョン管理

前述の依存関係管理を実現するため，システムには必要に応じて古いバージョンのオントロジーが保持されている．よって，システムが管理している依存関係情報に，それらのオントロジーのバージョン情報を加える事で，バージョンを含めたオントロジー管理は容易に行える．

しかしながら，例えば複数の部分的なオントロジーを統合して一つのオントロジーへと変化させる際などは，バージョン管理が複雑になる．このような問題の解決は今後の課題となっている．

各オントロジーの更新と変化の影響の伝播

本研究では，“個々のオントロジーはローカルに編集され，共有場に公開された各オントロジーを更新することによって分散構築が進む”という方針をとる．各オントロジーは，編集開始時やオーサの要求した時に，公開された他のオントロジーの最新バージョンにアクセスする．これによって他のオントロジーからの変化が伝わり，オーサは整合性を保持するために必要な編集を加えることが可能になる．

オントロジーの再利用

本研究におけるオントロジーの再利用とは，ある分散構築過程で部分として存在するオントロジーを，別のオントロジー構築で用いることを指す．

オントロジー構築には開発者間の合意の形成が欠かせないため，分散構築においても，開発者のグループは目標を共有し，相互のコミュニケーションの下で各オントロジーを管理，また構築するべきである．よって，オントロジーを再利用するためのアプローチの一つは，再利用する側の開発者と元来の開発者との合意形成を支援し，全員を開発グループのメンバーとして分散構築を行うことである．しかし，このような形態の構築を支援する分野は主としてコミュニケーション支援の範疇に入るものであり，本研究の直接の対象ではない．別のアプローチは，ある段階のオントロジーを用い，それ以後は，再利用する側の開発者で独自の編集を加える方法である．本研究では，主としてこちらの形態の再利用支援を考えている．この場合，独自に編集されたオントロジーは，元来のオントロジーの派生としてではなく，別に構築されたオントロジーとして扱われ，管理される．そして，依

存関係管理の下，再利用が実現される．

4.2 協調構築とオーサ管理

複数のオーサが並行して同じオントロジーを構築すると，整合性を保持するのが極めて難しくなる．そのため本研究では，部分となる各オントロジーを構築するのは基本的には一人のオーサであると想定し，オーサ毎にオントロジーへのアクセスを管理している．しかし，並行作業を避ける形であれば複数人による構築が実現できるよう，本システムでは各オントロジーでメインのオーサを決定し，その許可によって他のオーサもそのオントロジーを編集可能とする．

また，オントロジーの再利用（4.1）の箇所でも触れたように，分散構築一般に見られる問題として，コミュニケーションの計りにくさが残されている．特にオントロジー構築はオーサ間の話し合いによって合意が形成されるため，コミュニケーション支援が必要なのは明白である．具体的な支援の仕様決定は本研究の目指すところではないが，例えば筆者所属の研究室ではネットワークを介した合意形成支援を行うツールの研究も進められている．

5. 「法造」における分散開発

本章では，これまで議論した内容を実装したシステムについて述べる．まず，本研究室で開発されているオントロジー構築支援環境「法造」の概要を示し，分散構築がそこでどのように実現されるかの概要を述べる．続いて，それを実現させる２つのツールについて説明する．

5.1 法造

「法造」[古崎 02b]は，オントロジー（＝“法”）を構築する（＝“造”）為の計算機環境で，「オントロジーエディタ」，「概念工房」（オントロジー構築ガイドシステム），「オントロジーサーバー」，そして「オントロジーマネージャ」の４システムから構成される（図３）．

オントロジーエディタは，基礎理論に基づいたオントロジーをグラフィカルに記述する環境を提供する．意味定義は，定義の継承のみならず，部分概念の担うロール概念や，関係概念についても記述可能であり，これらはシステムが動的に管理する． 概念工房[石川 02] は， オントロジー構築方法 AFM（Activity-First Method）に基づき，専門家等による自然言語のドキュメントを基にし，そこから語彙を抽出し，ガイドラインに沿ったオントロジー構築の過程を支援する．オントロジーサーバーは，上記の２つのシステムで構築されたオントロジーやモデルを管理する．またオントロジーで定義された公理に基づく整合性の検証機能も備えられている． オントロジーマネージャは，前節で述べた考察に基づき設計・拡張されたシステムで，部分

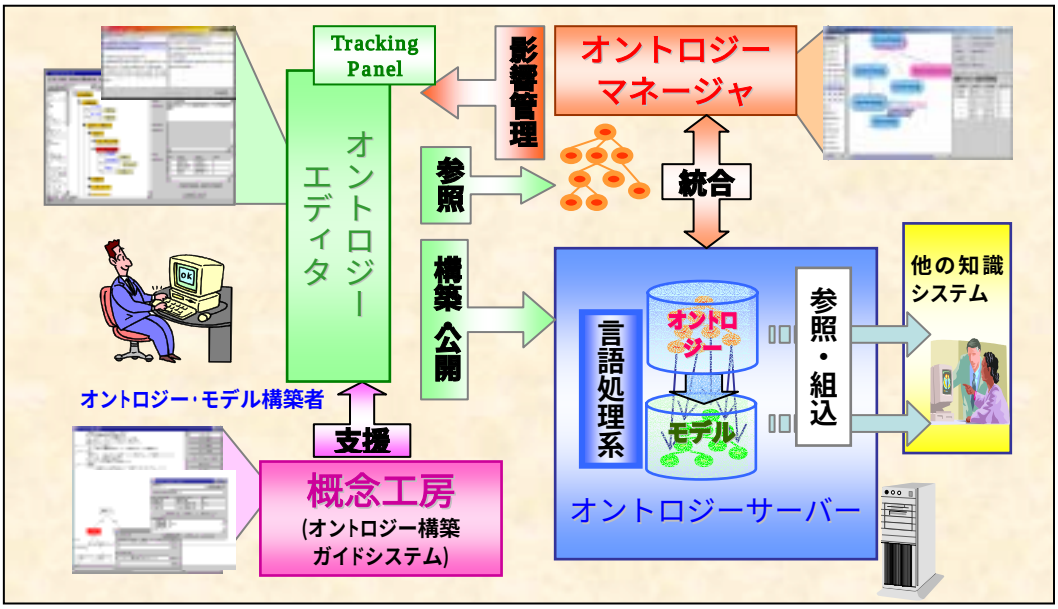

図３．「法造」の全体像

として存在する各オントロジーを仮想的に統合し，構築するオントロジーの全体像をオーサに提供する．これは本研究の中心的存在であり，5.3 節で詳細を説明する．

5.2 分散構築の実現

分散構築の流れ

分散構築は各オントロジーのオーサが以下のステップを繰り返す事で進められる：

1. 法造にログインする．
2. オントロジーマネージャを起動し，構築するオントロジーの全体像や相互の依存関係を把握する．
3. 編集するオントロジーをオントロジーマネージャから選択し，オントロジーエディタで開く．この時，オントロジーマネージャは依存している他のオントロジーにも自動的にアクセスし，依存関係の整合性や更新情報がチェックされる．
4. 依存関係のあるオントロジーが更新されていれば（あるいは必要があれば），整合性保持のための追従変化を行う．
5. オントロジーを編集する．なお，依存関係のチェックは編集中いつでも可能である．
6. 編集が終了したら，オントロジーをオントロジーサーバー上に保存し，必要に応じて他のオーサがアクセスできるように公開する．

データ構造とその利用

システムは，各オントロジー内に保持されている以下の情報に基づいて依存関係を管理する．

- オントロジーの名前，バージョン，オーサ，最終更新日時
- 依存している他のオントロジーの名前とバージョン
- 依存している概念定義のコピー

他のオントロジーと依存関係を結び，他のオントロジーの概念を利用する際，システムは依存する側のオントロジー内にその概念の定義をコピーしたデータを作成する．このデータは，依存関係を結んだ後に，その概念に変化があったかどうかを検出するために用いられる．オリジナルの概念とコピーのデータとを比較し，その差分から概念に加えられた変化を検出することができる．変化を検出した後は，表１の変化のパターンと照らしあわせ，必要に応じて合性保持のための支援を行う．

分散構築実現のためのオントロジー操作

オントロジーマネージャによって，オーサには以下の４つの操作が提供される：

①部分としての新規オントロジーの作成
②オントロジーの分割
③オントロジーの統合
④オントロジーの再利用．

これらの操作は主として，オントロジーの構成とその間の依存関係を先に規定しておき，その後で各オントロジーの詳細を構築する場合に用いられる操作である．
一方で，オントロジーの全体像が定まる前に各オントロジーの構築を開始し，構築途中必要に応じて依存関係を結ぶ状況も考えられる．この場合は，オントロジーエディタによって

⑤他のオントロジーの概念を検索し，依存関係を結ぶ

という操作が提供される．

5.3 オントロジーマネージャ

オントロジーマネージャ（図４）は，４つのパネルから構成される．

・Ontology List:： オントロジーサーバーに公開されているオントロジーの一覧を（選択的に）表示する．ユーザーがここから選択したオントロジーに関する必要な情報は，他の３つの画面に表示される．
・Ontology Viewer： 構築される各オントロジーと，そのオントロジーが持つ依存関係

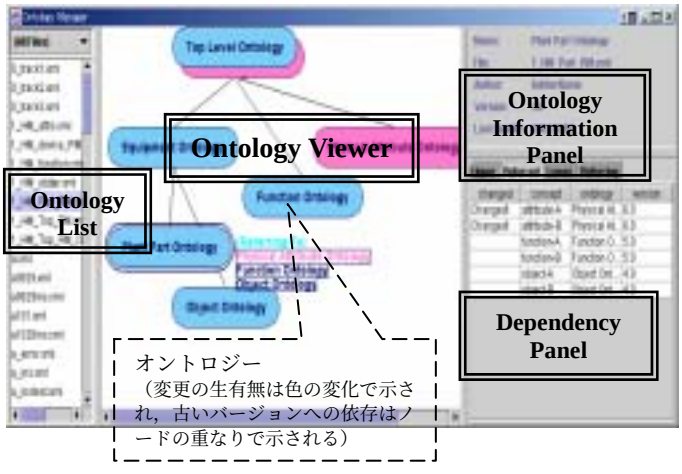

図４．オントロジーマネージャ

のうち継承関係がノードとリンクで表示される．

・Ontology Information Panel： 選択されたオントロジーの名前, バージョン, オーサ, 最終更新日時が表示される．

・Dependency Panel： 選択されたオントロジーと依存関係を持つオントロジーの一覧が，その関係の種類毎に表示される．分類は上位・下位・参照・被参照の４種で，これらは Tab で表示を切替える事が可能である．一覧には，オントロジー名，依存関係を担う概念，オントロジーのバージョン，変更の有無が表示される．

これらのパネルによってユーザーに情報を提供しつつ，オントロジーマネージャは「部分としての各オーサが構築するオントロジーと，開発目標となる統合されたオントロジーとの橋渡し」として，以下３つの機能を果たす．

1.依存関係管理

オントロジーマネージャは，個別に構築される各オントロジーを仮想的に統合し，その全体像を Ontology Viewer でオーサに提示する．オーサはこのパネルによって，構築するオントロジーの全体像を容易に把握することが可能である．また，依存関係の詳細は Dependency Panel によって知ることができる．

2.バージョン管理

オントロジーマネージャは，オントロジー同士を関連付ける際に，そのバージョン情報も扱う．選択されたオントロジーが古いバージョンのオントロジーに依存している場合，その情報は Ontology Viewer 上でグラフィカルに表示される．

公開されたバージョンのオントロジーはオントロジーサーバーに保存されており，オーサはオントロジーマネージャを介してサーバー内を検索して必要なオントロジーにアクセスすることが可能である．

3.オーサとアクセス管理

オントロジーサーバーには，分散構築によって開発されるオントロジーについて，それを構成する各オントロジーとそのオーサについての情報が保持されている．システムはこの情報に沿って法造にログインしたオーサを識別し，オーサが分散構築に加わっているオントロジーや，その中でオーサが編集可能な部分を管理する．

5.4 Tracking Panel

Tracking Panel は，依存しているオントロジーに変化があった際，整合性保持のための追従変化で用いられるオントロジーエディタ内のツールである（図３）．

あるオントロジーに変更が加えられたことは，Ontology Manager の Ontology Viewer や Dependency Panel でオーサに示される．その具体的な変化の内容は，変化の影響を受けたオントロジーのオーサが，オントロジーエディタでそのオントロジーの編集を開始したときに Tracking Panel 上で表示される．Tracking Panel には，加えられた変更のリストが表示され，それぞれに対応するための追従変化の方法が選択肢として提示される．オーサはその中から適切な追従方法を選ぶだけで，追従変化がシステムから半自動的になされ，影響を受けたオントロジーの整合性が保たれる．

6. 関連研究

オントロジー構築支援は様々なコミュニティで研究されており，本研究以外にも協調構築支援を謳ったツールが存在する．例えば Onto Edit[Sure 02]は，オントロジーに複数のオーサがアクセスすることを許可し，それによって協調構築を実現しようとしている．しかし，整合性を保持する機構は，アクセス制御によって同時編集を阻止するという単純なものである．これに対し，本研究はオントロジーを分割することによって複数のオーサが並行して構築することを可能にしており，かつ，それによって生じる整合性の問題を解決する支援も包括的に行っている．

また，KAON（Karlsruhe Ontology and Semantic Web Infrastructure）では，オーサの編集操作に対して整合性を保持するための追従を支援するツールの開発が進められている[Stojanovic 02]．そのツールも，オントロジーに変更が加えられた際，その種類に沿って整合性保持のための追従変化を支援する枠組みを持っている．これは本研究における整合性保持の方法と極めて類似したものである．しかし，KAON が対象としているのは単一のオントロジー内での概念体系の整合性であるのに対し，本研究が焦点を当てているのは分散構築下にある複数のオントロジー間における整合性である点が異なっている．

7. おわりに

本稿ではオントロジーの分散構築を行うに当たって必要な点を検討し，その実現方法と支援システムについて述べた．現在，基本的な機能を実装した試作段階でのシステムの開発が完了している．しかしながら必要な点が全て論じられたとは言えず，まだ発展させることのできる下記のような課題が残されている．

・影響を与える側からの整合性保持

本稿で扱ったのは影響を受ける側からの整合性保持であるが，その逆に，影響を与える側からも整合性を保持する事が可能である．これは，オーサの操作に何らかの制限をかけたり，あるいは代替案を示したりすることによって実現される．しかし，これらはオーサの活動を制限するものであるため，具体的な仕様は慎重に検討する必要がある．

・複数のバージョンを含む依存関係の取り扱い

例えば，構築したオントロジーを利用してインスタンスを生成する段階などでは，複数のオントロジーを統合して単一の概念体系とみなすことが必要になる．ここで，もし複数のバージョンのオントロジーが存在しているなら，“同一の概念が複数の個所で定義されている”などの問題が生じてしまう．よって統合時には，概念レベルでバージョンの調整を行う必要が生じる．

・依存関係以外のオントロジー間の関係を利用した構築支援

概念定義の依存関係以外にも，概念体系の構造に基づく関係（例えばロールホルダとロールの関係）などが存在し，それらは構築に利用可能だと思われる．

・オントロジー分割の方法論

オントロジーは概念の種類やレベルで分割することが可能であるが，どのような形で境界線を引くことが望ましいかの議論は本稿で扱っていない．この点はドメインや利用などに依存するところが大きいものの，適切な支援がなされれば，分散構築の初期段階においてオーサにとって非常に有効であると思われる．

今後も，さらなる支援機能の検討や改良を引き続き行っていき，より包括的なオントロジー分散構築環境の整備を目指す．

参考文献

[Sunagawa 03] E. Sunagawa, K. Kozaki, T. Kitamura, and R. Mizoguchi: Management of Dependency between Two or More Ontologies in an Environment for Distributed Development, Proc. of the International Workshop on Semantic Web Foundations and Application Technologies (SWFAT2003), pp.35-41, Nara, Japan, March 12, 2003

[砂川 02] 砂川，古崎，來村，溝口：分散開発環境における複数オントロジー間の依存関係管理，人工知能学会研究会資料(SIG-KBS-A202)，pp.53-58，2002

[Kozaki 02] K. Kozaki, Y. Kitamura, M. Ikeda, and R. Mizoguchi: Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of Role” and “Relationship”, Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002), pp.213-218, Sigüenza, Spain, October 1-4, 2002

[古崎 02] 古崎，來村，池田，溝口：「ロール」および「関係」に関する基礎的考察に基づくオントロジー記述環境の開発，人工知能学会誌，vol.17，No.3，pp．196-208，2002

[石川 02] 石川, 久保, 古崎, 來村, 溝口：タスク・ドメインロールに基づくオントロジー構築ガイドシステムの設計と開発－石油精製プラントを例として－, 人工知能学会論文誌, Vol. 17, No. 5, pp. 585-597，2002

[Sure 02] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke: OntoEdit: Collaborative Ontology Development for the Semantic Web, Proc. of the First International Semantic Web Conference (ISWC2002), Sardinia, Italy, June 9-12, 2002

[Stojanovic 02] L. Stojanovic, A. Maedche, B. Motik, N. Stojanovic: User-driven Ontology Evolution Management, Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002), pp.213-218, Sigüenza, Spain, October 1-4, 2002