

次世代Web委員会・2007年度活動報告

# オントロジーバージョンニング技術について



---

オントロジーバージョン管理ワーキンググループ

乙守 信行 (ジャストシステム株式会社)

清水 昇 (慶應義塾大学)

岡部 雅夫 (東京電力株式会社)



## 発表内容

---

- 1.モチベーション
- 2.オントロジーのバージョンニングとは
- 3.オントロジーバージョンニングのフレームワーク
- 4.構造的な変更と影響について
- 5.支援toolの要件
- 6.今後の予定



# モチベーション

## ■次世代Web委員会■

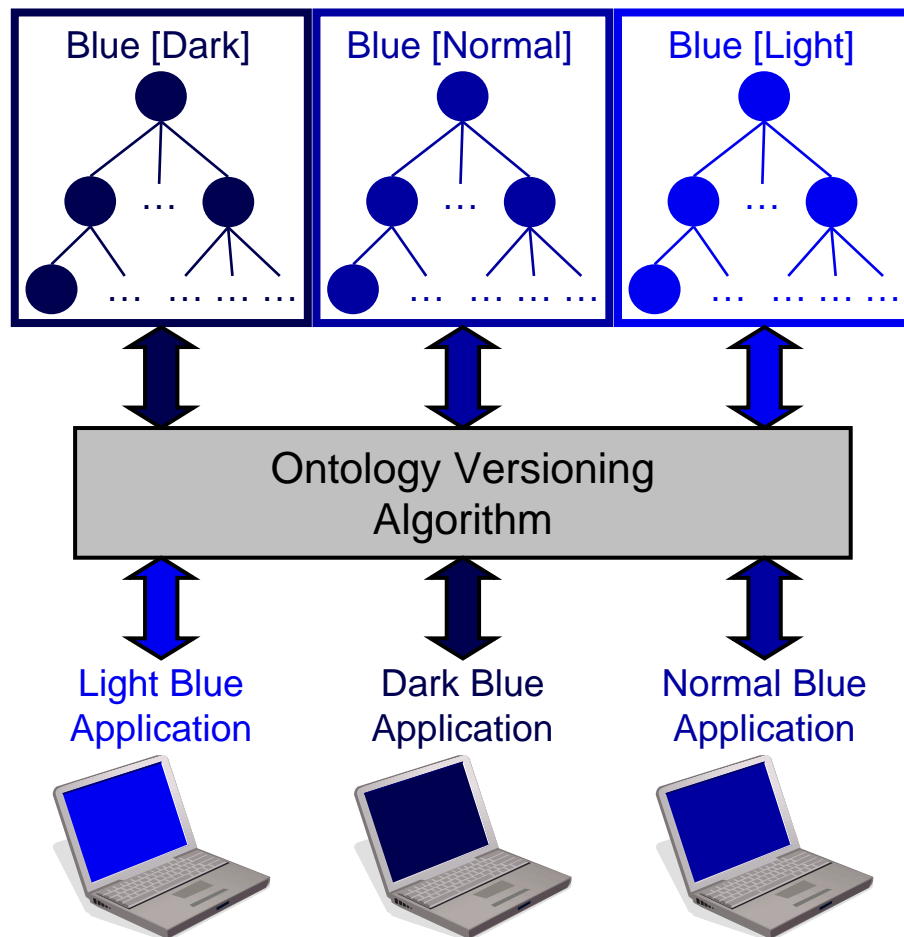
- オントロジー・リポジトリの作成
- オントロジーサイトの立ち上げ
- 情報家電オントロジー/程度表現オントロジーの公開
- ”オントロジバージョン管理“が現実的な問題として出始めた

## ■委員会で出された疑問点■

- URIで公開されているものにするか
- URIで公開しているファイルとURIで利用している語彙（語彙）との関係は
- 新しいオントロジーが古いオントロジーに置き換わるときのバージョン管理対象
- import時における整合性
- 情報家電オントロジー、法令
- 複数のオントロジーがあるのが前提とそうでない場合がある？
- 程度表現オントロジーが0.7 0.8と変わっていった場合はどうするのかというユースケース

オントロジーのバージョン変更時の影響、オントロジーの運用,バージョン変更を  
どう表現するのかの課題自体の整理から始めよう

# オントロジーバージョンニング

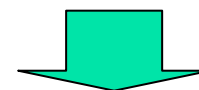


目的:

複数バージョンのオントロジを扱う。  
正しいバージョンのオントロジ、正しいデータ  
としてアクセスできるオントロジを提供する

入力: 異なったバージョンのオントロジ

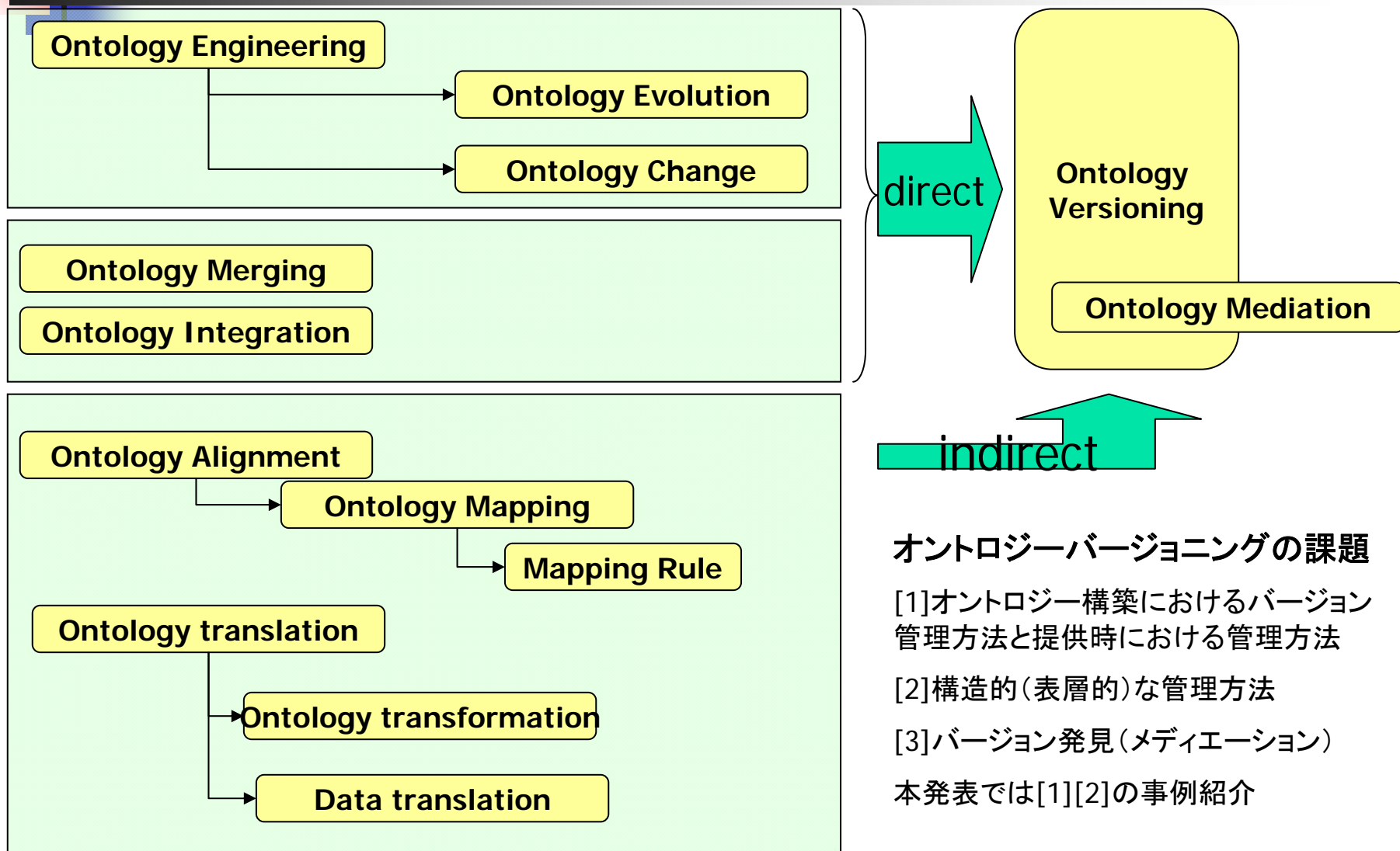
出力: 対象となるオントロジ



研究テーマとしては興味  
深く重要であるが、

ここまで急には行かな  
い??

# オントロジーバージョンニングの課題



## オントロジーバージョンニングの課題

[1]オントロジー構築におけるバージョン管理方法と提供時における管理方法

[2]構造的(表層的)な管理方法


[3]バージョン発見(メディエーション)

本発表では[1][2]の事例紹介

## 運用面

# オントロジーバージョンニングフレームワーク

オントロジー定義 (Identification)	オントロジーを構成する要素についてはあらかじめ明確にその意図を定義する。 これによって何を変更するのかということが明確に記述できることになる。
変更記述 (Change specification)	オントロジー定義に従って可能性がある変更は記述されなければいけない。 なぜならオントロジーの表現が異なっていれば、提供するコンポーネントも異なり、変更の記述も異なることになるからである。
変更の明確化 (Transparent evolution)	ある特定の変更によって何を行うべきかを明確にしておかなければいけない。 変更記述は、オントロジー構成要素の異なるバージョンの違いや関係の判定に利用されることになる。
タスクの認識 (task awareness)	互換性には多くの要因(例えば、インスタンスデータの保存、質問結果の保存、結果の保存など)があるので、適切な変換をバージョンの間に提供するために具体的なタスクを意識していなければならない。
変更履歴へのサポート (support for untraced change)	1つのバージョンから新しいバージョンまでの変更のステップを表す変化履歴がないケースがしばしばある。そのような場合、バージョン管理のフレームワークでは2つのバージョンが互換性があるかどうか決定するためにメカニズムを提供するべきである。
推論の整合性 (Consistent reasoning)	オントロジーを使った推論に影響がないことを保証するのを目的とする。これを実現する方法としては、少なくとも特定のセットの質問の答えが異なったバージョンでも同一結果を返すことで保証できる。
同期とデータ変換サポート (synchronization and data translation support)	分散環境では必要である。 これまでの機能は、ローカルにオントロジーをリモートにアップデートするものであるが、本機能は影響があるデータソースを新しいバージョンのオントロジーと整合性があるように自動的変換する。



# 構造的(表層的) オントロジーの変更箇所(OWL)

## 階層性

- クラスまたはプロパティの追加
- クラスまたはプロパティの削除
- クラスまたはプロパティのマージ
- ひとつのクラスを分割する

## プロパティ

- プロパティ名を換える
- プロパティのドメインとレンジを変える
- サブプロパティ関係を代える
- ラベルとコメントを代える

## クラス

- クラス名の変更
- クラスのラベル、コメント、数制約の変更
- 親クラスの変更
- 親クラスの削除
- 子クラスの追加
- 子クラスの削除
- クラスへのプロパティの追加
- クラスからのプロパティの削除

## その他

- プロパティ定義の変更
- 同等性または非同等性の変更
- 数制約の変更
- ユニオン/インターラクションの変更



# オントロジーの変更操作と インスタンスデータへの影響(1)

操作	影響	説明
新しいクラスCの作成	+	データは失われない
クラスを削除する	-	クラスC自体が特別でなければ上位クラスのインスタンスとなる だけすむ場合もある
スロットSの作成	+	データは失われない
クラスCにスロットSを加える	+	データは失われない
クラスCからスロットSを削除する	-	クラスCのインスタンスのスロットSの値は失われる
スーパークラスCとサブクラスC間に あたらな上位下位関係を加える	+	サブクラスは上位クラスから新しいスロットを継承する。多くの 場合は、新しいスロットを加えることと同じ操作となる
スーパークラスCとサブクラスC間に あたらな上位下位関係を削除する	-	サブクラスは上位クラスからの継承ができなくなる。 サブクラスCのインスタンスにおけるスロット値は失われる。
インスタンスIを新しいクラスにま とめあげる	+	データは失われない
クラスCをインスタンスとして整理す る	-	クラスCのインスタンスの型定義が失われる
C1とC2を排他として宣言する	-	C1とC2双方に属していたインスタンスの妥当性が失われる



# オントロジーの変更操作と インスタンスデータへの影響(2)

操作	影響	説明
スロットSを遷移または対称と定義する	-	遷移対称でないスロットSのスロット値との不整合が発生する
サブクラスCから上位クラスCへスロットを移動させる	+	サブクラスCは依然スロットSを継承する
上位クラスからサブクラスCへスロットを移動させる	-	上位クラスからはスロットCがなくなり、上位クラスCのインスタンスのスロットCの値は失われる。
クラスC1から参照クラスC2へスロットSを移動する	~	スロット値が移動しなければデータは失われない
新しいクラスにスロットの集合を隠蔽する	~	スロットの値が移動しなければデータは失われない
クラスCの上位クラスをより上位の階層へ変更する	-	Cはこれまでの直接に継承していたクラスからのスロットを保持することはできない。クラスCのこれらのインスタンスにあるスロットは失われる
クラスCの上位クラスをより上位の階層へ変更する	+	Cには新しいクラスが継承されることになるはずである。データは失われない。
スロットSの制約を緩める	+	スロット値はすべて整合性を保ったままである。
スロットSの制約を強める	-	強い制約に反するスロット値の妥当性がなくなる
クラスをマージする	~	スロットの値に移動がなければデータは失われない
クラスを分割する	~	スロット値を移動しなければデータは失われない



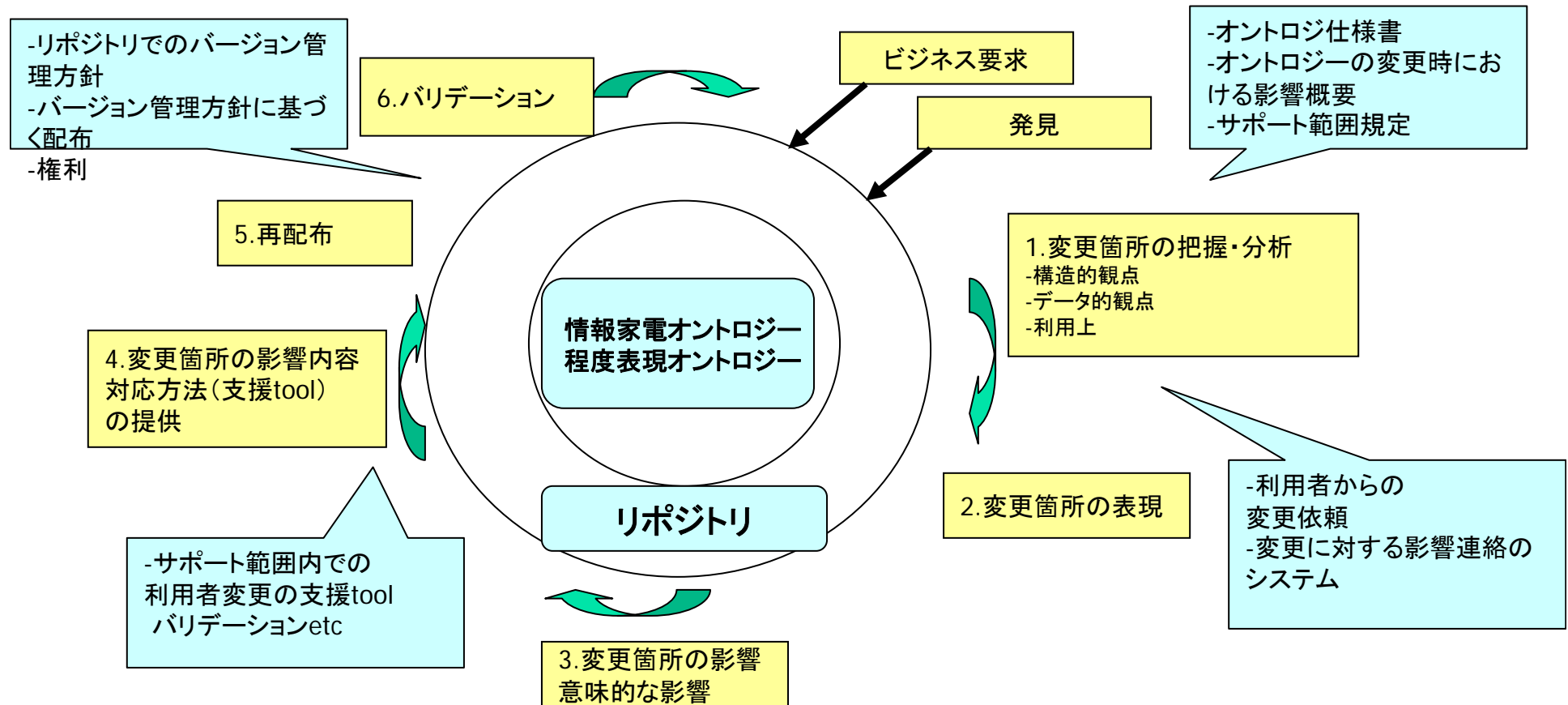
# 支援toolの要件

要件	概要
ツールの仕様として規定	オントロジーの特殊性が存在するので、オントロジーを変更操作については事前に決めておかなければならない
ユーザからの要望からの規定	ベースとなるオントロジーを整合性を保つというユーザー要求があるはずで、これに対して対応できるようにしておかなければいけない。
運用の透明性	変更がある場合には事前にユーザーに連絡がなければいけない
リバーシブル	変更結果において問題が発生した場合には元に戻せる手段が必要である。
変更履歴の保存	上記と関係して問題が発生した場合には、変更履歴から問題の発見ができなければいけない。
オントロジーの継続的な変更への支援	差分の自動抽出など、継続的な変更に対応できる機能が必要である。
ユーザビリティ	オントロジーが変更された際にユーザがわかりやすいような機能（不整合箇所のアラートや含意結果の差分など）が必要である

# 今後の予定

## 程度表現オントロジー、情報家電オントロジーとのリンク

■次世代Web委員会 情報家電オントロジ・程度表現オントロジ・リポジトリ管理を実践的な事例として検討を進める





# 備考:オントロジー技術概要(1/2)

Matching (マッチング)	異なったオントロジーのエンティティ間関係や対応を検出するプロセス。
Alignment (アライメント)	ふたつまたはそれ以上の(細胞の配列アライメントのように)のオントロジーの対応の集合。アライメントはマッチングプロセスの出力結果である
Correspondence (対応)	ある特定のマッチングアルゴリズムまたは異なるオントロジーのエンティティ間のインスタンスにおいて保持している(していると判定できる)関係。これらエンティティは、クラス、インスタンス、プロパティまたは式によって異なる。他ではマッピングの意味で使われることがあるが、本書ではこの意味で利用する。
Mapping (マッピング)	アライメントの基本であり、実際の処理。マッピングは一つのオントロジーを少なくともひとつのオントロジーにマップする。一般的な関係に対して数学的な定義や写像の概念を含む。数学的な定義では原則としてマッピング対象はその像と等価であることが要求される。すなわち、関係と等価関係を指す。マッピングは、同一方向へのマッピングルールの集合として捉えることができる。すなわちひとつのオントロジーから他のオントロジーに、元のオントロジーのエンティティが他のオントロジーの一回以上写像されることになる。
Mapping Rule (マッピングルール)	1つのオントロジーのエンティティを他のオントロジーのエンティティへ対応させること。
Ontology merging (オントロジーのマージ)	二つのオントロジーからひとつのオントロジーを作成することであり、大体はもとのオントロジーへ上書きすることになる。初期のオントロジーは変更がない。マージされたオントロジーは初期のオントロジーの知識を含むべきである。例えば、マージしたオントロジーを使った結果は、それぞれのオントロジーを使った結果になる。この概念はデータベースのスキーマ統合と非常に近い。

# 備考:オントロジー技術概要(2/2)

Bridge axioms (ブリッジの公理)	ブリッジとなる公理、または明示的な公理は、あるオントロジー言語における公式であり、これらはあるオントロジーを他のオントロジーのエンティティに統合化することができるよう表現されたアライメントである。同じ言語表現で記述されたオントロジーのマージにおいてブリッジ公理が基盤となる。
Ontology translation (オントロジー解釈)	あるオントロジーを別のオントロジー言語へ変換するプロセスのこと。オントロジを解釈するためのプログラムへも用いられる。
ontology transformation (オントロジー変換)	あるオントロジーのエンティティを他のオントロジーのエンティティの観点で表現するプロセスである。すなわち最初のオントロジーに対して二番目のオントロジーが追加された場合の関係となる。よって最初のオントロジーによって導かれる結論は変換結果と同じである。二つの初期のオントロジは変更されないそして、オントロジー変換結果から生成されたオントロジーが作成される。オントロジーを変換するためのプログラムとしても用いられる。
Data translation (データ解釈)	あるオントロジーから対応するデータやインスタンスを変換するプロセス。またはそのために用いられるプログラム。
Mediation (メディエーション)	情報ストリームを動的に変更すること、そのためのインターフェースをソフトウェア要素として持つ。メディエイトするためのプログラムも指す。WEBサービスではあるオントロジーを使ったサービス結果を別のオントロジーの入力へと変換を行う。そのためのデータ変換も行う。QAシステムでは、質問と回答での2重での変換を行う。
Ontology version(バージョンニング)	オントロジーのバージョンは、オントロジーを変更するアプリケーションの結果である。
Ontology reconciliation (オントロジー調整)	ふたつまたはそれ以上のオントロジーを調和させること。通常は、ひとつまた双方のオントロジーの変更が要求される。この場合、オントロジーのマージではなく、ともに進化する。オントロジ調整は二つのオントロジーをマージするためまたはそれぞれ独立してつくるために実行されるであろう。



# 參考資料

---

[1] Change Management and Versioning in Ontologies

- Baden Hugets (Department of computer Science and Soft ware Engineering)  
Semantic Technology Conference March6-9,2006

[2][A Classification of Ontology Change](#) Grigoris Antoniou

2Institute of Computer Science, FO.R.T.H., P.O. Box 1385, GR 71110, Heraklion, Greece

[3][Ontology Versioing. &evolution the state of art Burcu Yildiz October 2006](#)

[4]Ontology Matching

Jerome Euzenat Springer2007