

メディアエータの探索機能を支援するオントロジーの実現方式

横田一正

緒方啓孝

劉渤江

岡山県立大学情報工学部

{yokota,ogata}@c.oka-pu.ac.jp

〒719-1197 総社市窪木 111

岡山理科大学総合情報学部

liu@mis.ous.ac.jp

〒700-0005 岡山市理大町 1-1

分散した情報源を統合する際のもっとも大きな問題のひとつは、オブジェクト (概念) の同定に関してである。名前付け、汎化レベル、汎化階層、定義域、成立条件など、さまざまな違いが考えられる。これら差異を認識してオブジェクトの同定を行うためには、オブジェクトをいかに記述すべきか、いかに差異を認識するか、いかに同定処理を行うか、など解決しなければならない。高度な情報源を対象にしたメディアエータシステムを目指して現在研究開発中の分散知識表現言語 QUIK ではメタデータをデータ辞書に格納しているが、さまざまなオブジェクト記述を行うために、データ辞書を拡張したオントロジーを検討している。本稿では QUIK におけるオントロジーの内容とその実現方式、さらにそれを利用した情報の分散探索方式について述べる。

Ontologies for Supporting Distributed Query Processing in Mediator Systems

Kazumasa Yokota

Nobutaka Ogata

Bojiang Liu

Okayama Prefectural University

{yokota,ogata}@c.oka-pu.ac.jp

Soja, Okayama 719-1197

Okayama University of Science

liu@mis.ous.ac.jp

Ridai-cho, Okayama 700-0005

One of the most critical problems for integrating distributed information sources is about identification of objects, or concepts. There are many differences between distributed objects: e.g., in naming, generalization levels, generalization hierarchies, discourse domains, and existence conditions. To resolving these differences for identifying objects, there are many problems to be solved such as how to describe objects or concepts, how to know their differences, and how to process identification of related objects. We have reserached and developed a distributed knowledge representation language, QUIK, for advanced mediator systems, where each information sources may have knowledge information. In this paper. we extend QUIK data dictionaries to ontologies, which can have not only metadata but also various object or concept descriptions. We describe contents of an QUIK ontology and the procedural semantics, and how to use them for distributed search.

1 はじめに

分散された情報源を統合することを目指して分散知識表現言語 QUIK を研究開発している。この特徴は以下のようになる。

- 情報源として演繹オブジェクト指向データベースのような、知識表現を含む高度な情報源を対象としていること
- 情報源間の共通モデルとメタデータ仕様言語として QUIK を採用していること
- 情報源間の矛盾を避けるために、大域的概念階層の動的構築を行なうこと
- 分散情報源の探索のために、アブダクションを利用した導出を行うこと
- 不完全情報の探索をサポートするため条件付き問合せと仮説生成の機能を提供していること

すでに Java でプロトタイプシステムを開発し、その有効性を検討しており、同時に拡張機能をもった QUIK の新しい版を検討している。

分散情報源の探索での本質的な問題は、情報源間の異種性をいかに解消するかである。その出発点として QUIK ではふたつの機能を持っている。

- 情報源間の概念の関連 (同値関係) をデータ辞書内に定義する。これによって名前の違いを解消している。
- 各情報源がもっている概念階層を融合し大域的概念階層を生成する過程で発生する矛盾を、データ辞書内の概念間の関連の定義に反映する。これによって汎化関係の違いを解消している。

しかし各情報源で使用している概念の異種性は単に、名前の異種性や汎化関係の異種性にとどまらず、属性値の条件によって異なった概念になったり、問題領域によって異なった意味を与えられたりする。つまり概念記述あるいは定義での同定あるいは異種性の判断が必要になることが多い。したがって本稿では、概念の記述をオントロジーという枠組の中で検討し、それが分散環境での情報探索にいかにも有効に働くかを、QUIK 言語を例にとり検討する。

まず 1 節では本稿で使用するオントロジーの意味を明確にし、4.1 節では QUIK の分散探索方略を概説する。4.2 節では QUIK におけるオントロジーの実現方式を検討し、それを使用することによる導出方略を検討する。5 節では他の関連研究との比較を行なう。

2 分散環境での情報探索とオントロジー

2.1 オントロジーの概念

オントロジー (ontology) はもともと哲学の用語で「存在論」を意味しており、さまざまな存在を関連付けたり同定することによって、すべての存在を体系的に説明することを目指している。計算機分野ではオントロジーは知識ベースの共有と再利用のために脚光を浴びてきたもので、まだ明確に決まった定義はないが、

- ある対象世界 / 問題領域に関する概念化の仕様 ([1])
- 知識ベースを共有 / 再利用するための概念体系を再利用するための概念記述
- 知識ベースを構築するためのビルディングブロック ([6])
- 複数の対象世界が協調するための共通する概念体系およびその構築方法

などと理解されている。またこれらに工学的にアプローチするために「オントロジー工学」が提案されている ([7, 6, 4, 5])。

データベース分野では異種分散データベースやマルチデータベースの研究の中で、異種性の解消へのさまざまなアプローチが研究されてきた。伝統的にデータベースはメタデータをデータ辞書の中に持っており、その拡張された枠組の中にスキーマ間の関連など異種性解消のための情報をもたせることが行なわれている。静的な大域スキーマが期待できない環境では、それら情報を利用して動的に大域スキーマを生成する必要がある。しかし演繹オブジェクト指向データベースのように、オブジェクト階層 (概念階層) をもったデータベース間の協調を考えるためには、複数の概念階層の無矛盾性を解消し、概念の同定を行なう必要があるため、概念記述を含んだ知識ベースのオントロジーの枠組で検討するのがふさわしいと考えられる。情報検索で用いられるシソーラスは用語 (概念) 間のさまざまな関連を示した辞書であるが、明確な概念記述を持っていないために、拡張する必要がある。拡張されたシソーラスとその構築方法をオントロジーと呼ぶこともできるだろう。最近シソーラスに対してオントソーラス (ontosaurus) という用語も作られている。

本稿では、概念 (オブジェクト) の概念階層 (包摂関係) に概念記述を導入し、複数の概念階層を統合すること、つまり既存の (拡張された) 概念階層を再利用することによって、あるいはそれらをビルディングブロックとして考えることに

よって、新たな概念階層を構築することを検討する。それら拡張された概念階層とその構築方法をあわせてオントロジーと呼んでいる。

2.2 いくつかの例

オントロジーがいかに使用されるかをSKC (Scalable Knowledge Composition)[11]で取り上げられている例を使って考えよう。靴 (SHOE) の名前やサイズは小売店 (store) と工場 (factory) で異なっている。ここでは2つの問題領域 / 情報源間の概念の関係を表現している。

- shoe (store) = shoe (factory)
- {pumps,wedgies,loafers,sandals} (store)
ISA shoe (factory)
- pumps (store) = shoe (factory)
IF heel (factory) > 5cm
- IF (location (factory) = 'europe'
THEN size (factory) = sizetable (size (store));
ELSE size (factory) = size (store)
- colorcode (factory) = colortable (color (store))

直感的な意味は明らかだが、この例では概念と属性の区別が曖昧になっている。本稿ではこの例を QUIK で実現し拡張することを目指す。

2.3 QUIK のオブジェクト

QUIK のオブジェクトにはオブジェクト項と属性項の2種類がある。オブジェクト項は基本オブジェクトの集合 O 、ラベルの集合 L 、変数の集合 V から定義できる。

- 1) 基本オブジェクト、変数はオブジェクト項である。
- 2) $o \in O, l_1, l_2, \dots, l_n \in L$ 、オブジェクト項 o_1, o_2, \dots, o_n とすると、 $o[l_1 = o_1, l_2 = o_2, \dots, l_n = o_n]$ はオブジェクト項である。

このオブジェクト項は構文的には通常の述語の拡張になっており、述語そのものも QUIK の構文的にオブジェクト項として認めることにする。

基本オブジェクト項間には包摂関係が定義されており、これはオブジェクト項間の包摂関係に拡張できる。たとえば

$$o_1[l_{11} = o_{11}, \dots, l_{1m} = o_{1m}] \sqsubseteq o_2[l_{21}, \dots, o_{2n}]$$

$$\stackrel{\text{def}}{=} o_1 \sqsubseteq o_2 \wedge \forall 1 \leq j \leq m, \exists 1 \leq i \leq n. (l_{2j} = l_{1i} \wedge o_{1i} \sqsubseteq o_{2j})$$

半順序関係は容易に東に変換できるので、一般性を損なうことなく包摂関係は東であると仮定する。

属性項は、明示的あるいは暗示的に示されるオブジェクト項間の包摂制約をオブジェクト項に付加したものである。暗示的にオブジェクトを表現するドット項は以下のように定義できる。

- 1) オブジェクト項 $o, l \in L$ ならば $o.l$ はドット項である。
- 2) ドット項 $d, l \in L$ とすると $d.l$ はドット項である。

オブジェクト項またはドット項 o_1, o_2 に対して、 $o_1 \sqsubseteq o_2$ を包摂制約という。包摂制約の集合に対しては決定可能な制約解消系が定義できる。オブジェクト項 o 、包摂制約の集合 C に対して、 $o|C$ を属性項と呼ぶ。この属性項が QUIK 同様、QUIK におけるオブジェクトとなる。包摂制約の略記として

$$o|C \cup \{l \sqsubseteq o'\} \Leftrightarrow o/[l \rightarrow o']|C$$

$$o|C \cup \{l \supseteq o'\} \Leftrightarrow o/[l \leftarrow o']|C$$

$$o|C \cup \{l \cong o'\} \Leftrightarrow o/[l = o']|C$$

が用いられる。

オブジェクト項中の属性を固有、属性項中の包摂制約としての属性を付随と呼ぶ。

上記オブジェクトの定義ではラベルを別に定義しているが、オブジェクト項と属性項で使用されるラベルもオブジェクト項でも構わない。この場合の包摂関係は以下のように定義される。

$$o \sqsubseteq o' \wedge l \supseteq l' \wedge O \sqsubseteq O' \Leftrightarrow o[l = O] \sqsubseteq o'[l' = O']$$

たとえば以下が成り立つ。

$$shoe[size = medium] \sqsubseteq shoe[size(cm) = medium]$$

$$heel(cm) \text{ はオブジェクト項の述語表現である。}$$

2.4 QUIK におけるオントロジー

複数の情報源に存在するオブジェクト間の関係は、いくつかに分類することができる。まずここでは分類ごとに直観的な説明を行なう。

(1) 同一のオブジェクトの別名

$$s_1 : o_1 = s_2 : o_2$$

情報源 s_1 の基本オブジェクト o_1 は情報源 s_2 の基本オブジェクト o_2 と同一である。

ユーザが明示的に指定することも可能だが、QUIK では関連する情報源の包摂関係からそれらにまたがる大域的な包摂関係 (東関係) を生成する過程でそれを半自動生成している。すなわち、情報源から収集された包摂関係の前順序から

束を生成する際に検出した矛盾 ($o_1 \sqsubseteq o_2 \wedge o_2 \sqsubseteq o_1$) から、それを解消するために、基本オブジェクト間の同値関係のあるいは否定すべき包摂関係の集合を候補解を生成している[8]。この中から選択された基本オブジェクト間の同値関係がオントロジーに格納される。

(2) 複合オブジェクト (オブジェクト項) の別名

$$s_1 : o_1 = s_2 : o_2 [l = a]$$

情報源 s_2 のオブジェクト項 $o_2 [l = a]$ は情報源 s_1 の基本オブジェクト o_1 と同一である。

単一の情報源でこのような別名 / 概念定義を行なうと、

$$o_1 = o_2 [l = o_3] \wedge o_2 = o_1 [l = o_4]$$

のような定義が現れたとき、包摂関係で

$$o_1 \cong o_2 [l = o_3] \sqsubseteq o_2 \cong o_1 [l = o_4]$$

となり、 o_3, o_4 の値とは関係なく、基本オブジェクト間の同値関係や包摂関係の削除では解消されない矛盾が生じる。したがって従来は構文的にこのような指定を禁止していた。しかし分散環境では、このような形の汎化階層は自然で、このような状況の生じないことを保証できない。

(3) 分散オブジェクト間の包摂関係

$$s_1 : o_1 \sqsubseteq s_2 : o_2$$

「包摂関係」として情報源内で定義されているものと異なり、大域的束の生成時に機能する。 o_2 がオブジェクト項の場合別名定義を付加する。

(4) 分散オブジェクト間の束演算

$$s_1 : o_1 = s_2 : o_2 \downarrow s_3 : o_3$$

ひとつの情報源の中では束の meet 演算 (1) として定義されているが、情報源間では大域的束を生成するときの条件として機能する。meet 演算の引数にオブジェクト項が使用されたときには、上で述べたオブジェクト項に対する別名を加える。join 演算についても同様。

(5) 属性値によるオブジェクトの分類 属性値により異なったオブジェクトと対応するような記述も必要となる。QUIK では固有属性と付随属性の2種類を考えなければならない。

i) 固有属性値による条件分けは通常のオブジェクト項と対応させる。

$$s_1 : o_1 = s_2 : o [l = v_1]$$

$$s_1 : o_2 = s_2 : o [l = v_2]$$

⋮

条件が排他的でない場合には meet 演算に対応させる。

$$o_1 \downarrow o_2 = o [l = v_1 \downarrow v_2]$$

ii) 付随属性値による条件分け

単一の情報源内では考えられないが、分散環境では

$$s_1 : o_1 = s_2 : o [l = v_1]$$

$$s_2 : o_2 = s_2 : o [l = v_2]$$

⋮

という定義が考えられる。 o_1, o_2 を包摂関係の中で考えれば明らかに矛盾する。しかし l の属性値が未評価であればこの定義は意味を持っている。

(6) 他のオブジェクトの属性値による分類

$$s_1 : o_{11} = s_2 : o \Leftarrow s_2 : o_2 [l = v_1]$$

$$s_1 : o_{12} = s_2 : o \Leftarrow s_2 : o_2 [l = v_2]$$

⋮

$$s_1 : o_{1n} = s_2 : o \Leftarrow s_2 : o_2 [l = v_n]$$

情報源 s_2 のオブジェクト o_2 の付随属性値によってオブジェクトの同値関係が決定される。ただし o_2 の属性値が互いに排他的でない場合には、

$$s_1 : o_{1i} \downarrow = s_1 : o_{1j} \Leftarrow s_2 : o_2 [l = v_i \downarrow v_j]$$

のように meet を取ることにする。この分類に使用する属性として固有属性値を用いると、 $i \neq j$ であっても

$$s_1 : o_{1i} = s_1 : o_{1j}$$

が成立する危険性があり、その場合、束の動的な変更が生じるので禁止される。

以上の (1) ~ (6) の結果生成される同値関係などのプロジェクト (概念) 記述が QUIK のオントロジーに格納される。

3 QUIK でのオントロジーの手続きの意味

前節で概説したオントロジーがどのような意味をもっているかの手続きの意味を説明する。

3.1 分散環境でのオブジェクトと包摂関係

単一の情報源の場合、F-logic ではルールによって包摂関係の生成を許しているがそれは属性継承に寄与しないものとして位置付けられ、また多重継承は非決定的な選択によって処理している。一方 *Quixote* では多重継承は包摂制約のマージとして処理しているが、包摂関係の動的な生成は構文

の上で禁止していた。QUIK は単一情報源の場合 *Quixote* の方式を踏襲しているため、問合せ実行前に関連する情報源に共通の包摂関係が生成されてなければならない。そうでなければ属性継承の一貫性が保証されず、問合せ処理自体を無意味にしかねないからである。したがって、分散環境での複数情報源の場合、関連する包摂関係をマージして大域的束を生成するというアプローチをとっている。F-Logic のように新しく生成されたものを通常の包摂関係から除外するアプローチは、分散環境の場合には関連する別の情報源の既存の包摂関係を処理の対象外にするのに等しく、適切ではない。

ただしオブジェクト項の扱いに2種類があることに注意が必要である。オブジェクト項がオブジェクトとして使用されるときは包摂関係に反映されるが、ラベルであれば単に翻訳ルールとしてしか機能しない。つまり構文としては

$$o/[o = o]$$

が現れたとき、それは $o \wedge o.o \cong o$ と解釈されるので、3つの o の意味は異なっている。最初の o はオブジェクトの存在を、2番目の o はリンク情報を、そして最後の o は制約である。1番目と3番目の o は包摂関係を反映し、2番目は反映しないことになる。

3.2 オントロジーの変換手続き

翻訳は簡単なので、以下包摂関係の扱いを中心に2.4節で述べたオブジェクト定義は以下のように展開される。

(1) はすでに QUIK で実現されている ([8])。同値関係によってプログラムは書き直されるのではなく、制約解消等が必要になった場合に参照される。

(2) の場合、以下の定義

$$o_1 \cong o_2[l = o_3] \sqsubseteq o_2 \cong o_1[l = o_4]$$

によって矛盾が引き起こされるかどうかは

- 各情報源のオントロジー中に定義されている基本オブジェクトの同値関係
- 大域的束生成時に生成された基本オブジェクトの同値関係

を各情報源の (2) の定義とつき合わせることで簡単に判断できる。このような矛盾解消の候補解として、 $o_1 \cong o_2$ の他に o_1, o_2 の固有ラベル l に関する属性と制約を無効にする。もし矛盾が生じたらこの候補解をユーザに問い合わせることによって矛盾解消する。

$s_1 : o_1 = s_2 : o_2[l = o_3] \wedge s_1 : o_1[l = o_4] = s_2 : o_2$ で、単純

に属性を無視する場合には s_1 に

$$s_1 : o_1 = s_2 : o_2$$

s_1, s_2 にそれぞれ

$$s_1 : o_1 \Leftarrow s_1 : o_1[l = X].$$

$$s_2 : o_2 \Leftarrow s_2 : o_2[l = X].$$

のルールを追加する。同値関係の追加によって新しく矛盾が生じる危険性があるので、再度束生成を実行する。属性の削除は実行時に無視することで可能になる。

(3) は基本オブジェクト間の包摂関係であれが大域的束生成のために使用し、そうでなければ問題を分割する。つまり

$$s_1 : o_1 \sqsubseteq s_2 : o_2[l = v_2]$$

の場合、 s_1 に新しいオブジェクト o_3 を導入し、

$$s_1 : o_1 \sqsubseteq s_1 : o_3$$

$$s_1 : o_3 = s_2 : o_2[l = v_2]$$

と展開し、(2) の問題に還元する。

(4) は以下の分解を行ない、得られた基本オブジェクト間の包摂関係を大域的束生成に使用する。

$$o = o_1 \downarrow o_2 \implies o \sqsubseteq o_1, o \sqsubseteq o_2$$

$$o = o_1 \uparrow o_2 \implies o \supseteq o_1, o \supseteq o_2$$

$$o = O_1 \downarrow O_2 \implies o \sqsubseteq o', o \sqsubseteq o'', o' = O_1, o'' = O_2$$

$$o = O_1 \uparrow O_2 \implies o \supseteq o', o \supseteq o'', o' = O_1, o'' = O_2$$

ただし O_1, O_2 は複合オブジェクト項で、 o', o'' は新たな基本オブジェクトである。

(5)-i) は (2) と同様であり属性値が互いに排他的でない場合も、オブジェクト通常の実演算で対応できる。

(5)-ii) は付随属性値を参照する情報源 s_2 に

$$o[l = X] \Leftarrow o/[l = X]$$

というルールを付加し、固有属性値と同じに考える。基本オブジェクトを定義している情報源に (5)-i) の固有属性値の定義を追加する。

(6) の矛盾解消の基本的なアイデアは、オブジェクト定義をしている情報源に下記の新たなルールを追加することである。つまり s_1 内に $1 \leq i \leq n$ に対して

$$s_1 : o_{1i} = s_2 : o_{2i}$$

を格納し、 s_2 にはルールとして

$$s_2 : o_{2i} \Leftarrow s_2 : o_{2i}[l = v_i]$$

を追加する。しかしこれでは $o_{2i} = 0$ という関係が出てこないし、それを定義すれば新たな矛盾が生じる。そこでオブジェクト o が固有属性値 v_i に依存しているのでその値で制限する。つまり s_1 内に $1 \leq V_i \leq n$ に対して上記 o_{2i} を $o[V = v_i]$ とする。

3.3 例再考

2.2 で取り上げた例を QUIK で表現しよう。2つの情報源 store と factory 間の関係である。

- 1) shoe (store) = shoe (factory)

QUIK では名前による同一性を前提にしているのでこれは不要となる。

- 2) {pumps,wedgies,loafers,sandals} (store)
ISA shoe (factory)

現在は個別に包摂関係を定義することになっているが、構文糖衣を導入すれば以下のように表現できる。

```
store : {pumps,wedgies,loafers,sandals}
      ⊆ factory : shoe
```

- 3) pumps (store) = shoe (factory)

IF heel (factory) > 5cm

heel を shoe の属性と考える。

```
store : pumps = factory : shoe
      ⇐ factory : shoe/[heel = X] | {X ≤ 5}
```

cm の単位が必要ならラベルに別属性として導入する必要がある。

```
store : pumps = factory : shoe
      ⇐ factory : shoe/[heel(cm) = X] | {X ≤ 5}
```

この構文は動的に同値関係を生成するので QUIK のルールとしては認められていないので、上記のように

```
store : pumps = factory : shoe[heel(cm) = X] | {X ≤ 5}
factory : shoe[heel(cm) = X]
```

```
⇐ factory : shoe/[heel(cm) = X] | {X ≤ 5}
```

をオントロジー定義のために導入し、最初の同値関係は静的に評価することにする。

- 4) if (location (factory) = 'europe')

THEN size (factory) = sizetable (size (store));

ELSE size (factory) = size (store)

location, size, sizetable に曖昧さがあるが、まず

```
store : sizetable.size = factory : size
      ⇐ factory : factory/[location = europe]
```

```
store : size = factory : size
```

```
⇐ factory : factory/[location = X] | {X ≠ europe}
```

と単純に変換でき、さらに以下ようになる。

```
store : sizetable.size
      = factory : size[location = europe]
```

```
factory : size[location = europe]
      ⇐ factory : factory/[location = europe]
```

ただしドット項 *sizetable.size* は変数と同様で静的には評価できないので、制約の評価は受動的にする必要がある。否定の場合も同様である。

- 5) colorcode (factory) = colortable (color (store))

これは単純に以下のように変換される。

```
factory : colorcode = store : colortable.color
```

この制約評価も前の例と同様に受動的に行わなければならない。

4 分散環境での探索

QUIK での基本的な探索方略は、情報源内での欠落情報がアプリケーションによって見つかったら、それをデータ辞書/オントロジーを経由して他の情報源を探索することである。

4.1 QUIK の探索方略^[14]

QUIK プログラム P は、オントロジー O 、メディアータ間関係 I 、包摂関係 S 、ルールの集合 R からなっている。つまり $P = (O, I, S, R)$ の四つ組である。オントロジー導入前に比べると、データ辞書/ディレクトリ (DD/D) D を、2節に述べた定義情報を持つよう O に拡張された。一つの情報源に対し、少なくとも一つの QUIK プログラムが定義される。通常のプロダムと同様、他のプログラムを静的に取り組むことができるので、オントロジー部分を独立させて複数のプログラム間で共有させることも可能である。

各ルールは以下の形式をしている。

$$a \Leftarrow m_1 : a_1, \dots, m_n : a_n | C$$

m_i はオブジェクト項のメディアータ識別子で、 a, a_i は属性項、 C は情報源識別子を含む包摂関係を集めた制約である。 $n = 0$ のときそのルールはファクトと呼ばれる。メディアータ識別子としては一意的にメディアータを識別するものではなく、機能名を用いることもできる。この場合、機能名とメディアータ識別子との対応はオントロジーで定義されている。

QUIKOTE と同様に問合せのための導出過程を定義している [13]。したがって導出列は

$$(G_0, A_0, \emptyset) \Rightarrow (G_1, A_1, C_1) \Rightarrow \dots \Rightarrow (\emptyset, A_n, C_n)$$

の3つ組のノードの列となる。ここで G_i は未解消オブジェクト項の集合、 A_i は未解消制約の集合、 C_i は解消済制約の集合である。オブジェクトのゴールが充足されれば、充足されない制約 A_n のみを仮説として生成し、解のひとつとして (\emptyset, A_n, C_n) を生成してきた。これはオブジェクトの存在は仮説としては生成しないが、その属性は仮説とすることに対応していた。

まず従来通りのデータ辞書機能はサポートされる。QUIKプログラム (O, I, S, R) 、問合せ $?-G$ に対する失敗した導出列の最終ノードが $N_i = (G_i, A_i, C_i)$ であるとする。オントロジー O が、オブジェクトに対して情報源識別子を返すとする。 $G_i = G'_i \cup \{s_j : o\}, s_j : o \notin G'_i$ のとき、 O が o に対して $I_C = \{s_1, s_2, \dots, s_n\}$ を返すとすれば、 N_i から

$$N_{i_1} = (G'_i \cup \{s_1 : o\}, A_i, C_i)$$

$$N_{i_2} = (G'_i \cup \{s_2 : o\}, A_i, C_i)$$

⋮

$$N_{i_n} = (G'_i \cup \{s_n : o\}, A_i, C_i)$$

の代替ゴールを生成することができる。さらにオブジェクトの存在が解消された後、属性に関する制約解消にもオントロジーを用いることができる。情報源識別子に対応する変数が実行時に束縛されていない場合、このオントロジーによって可能性のある情報源を見つけることができる。未充足の制約の場合ドット項の評価がオントロジーを経由して行なわれる。

QUIK における導出には、最終ノード (G_n, A_n, C_n) の種類に応じて以下の種類がある。

- 1) $(\emptyset, \emptyset, C_n)$: 成功
- 2) (\emptyset, A_n, C_n) :
 - (a) オントロジーを用い A_n を他の情報源で探索
 - (b) 結果によって 1) か 2) を繰り返す
 - (c) A_n が解消できなければ仮説付き解として成功
- 3) (G_n, A_n, C_n) :
 - (a) オントロジーを用い G_n を他の情報源で探索
 - (b) $G_n = \emptyset$ となれば 1) か 2) の処理
 - (c) G_n が解消できなければ $(G_{n-1}, A_{n-1}, C_{n-1})$ に対して 3) を繰り返す
 - (d) $n = 0$ となれば失敗

4.2 オントロジーを利用した探索方略

QUIK での DD/D は以下のように定義されていた。

$$I_i = (ID, Loc, Func, E)$$

ここで ID は自己識別子である。所在情報, Loc , は以下の形式になっている。

$$\{(ID_i, Ploc_i, \{(O_{ij}, \{A_{ijk}\})\})\}$$

ID_i はメディアータ識別子で、 $Ploc_i$ は対応する物理アドレスで、 O_{ij} は ID_i に含まれるオブジェクト識別子、 L_{ijk} は O_{ij} に含まれるラベルである。機能名テーブル $Func$ は以下の形式をしている。

$$\{(f_i, \{ID_{ij}\})\}$$

ここで f_i はローカルに定義された機能名でメディアータ識別子の代わりに使用される。 $\{ID_{ij}\}$ は対応するメディアータ識別子である。 E は関連するメディアータ間のオブジェクトの同値関係である。本稿で検討した定義情報がこれに新たに加わることになる。

オントロジーに関連する機能を挙げると以下のようになる。

● 静的な処理

- オブジェクト記述に関連した同値関係、包摂関係、ルールの生成と追加。
- 関連する情報源から包摂関係を収集し大域的包摂関係(束)を生成。これで新たに生成された同値関係等をオントロジーに登録。

● 動的な処理

- オブジェクトの所在情報の管理と情報提供。これは問合せ結果によっても保守を行う。
- ドット項に関連した制約情報の評価
- 問合せ中のオブジェクトの翻訳

導出の安全性を保証するために基本的には静的な評価が大部分となっている。問合せ処理中には前節の導出手続き中の 2)(a), 3)(a) での基本機能としての問合せと、他情報源への問合せの変換を処理することになる。

5 おわりに

オントロジーに対する関心が高まるにつれてそれまで曖昧だった概念を整理し、工学のひとつの分野として確立しようとされている [7, 6, 4, 5]。ネットワーク上にもさまざまな情報が整備されている [10]。この研究の出発点は、大規模知識

ベースの共有と再利用であり、研究のアプローチには知識工学、大規模知識ベース構築、そしてデータベースの3つの立場からのものがあるように考えられる。

本稿で取ったアプローチは、知識ベースはデータベースの拡張と考えるとの視点から、データベース的オントロジー [11, 12] の拡張を目指している。本研究のオントロジーの特徴は以下のようにまとめることができる。

- 演繹オブジェクト指向という高度なデータモデル、あるいは知識表現でのオントロジーの実現を目指している。
- 包摂関係 (属性継承) の無矛盾性の保持を大域的束生成として処理し、そこで生成された同値関係をオントロジーに組み込んでいる。
- QUIK の仮説生成・仮説推論の問合せ環境 [13] は、オントロジー構築のための実験 / 検証環境の提供にもなっている。

本稿での検討は、オントロジーの導入が *QUIXOTE* で始めた演繹オブジェクト指向アプローチの枠組に素直に収まることを示している。

今後の研究としてまず、本稿で提起したオントロジーの意味論を厳密に詰める必要がある。たとえばオブジェクト項の拡張 (3.2節の (5)ii) と (6)) と縮小 (3.2節の (2)) の意味的変更、包摂制約の受動的な評価、などを考えなければならない。次に別に検討を行なっている問合せの拡張 [9] との関連を考えなければならない。つまり、移動エージェントはオントロジー情報をどこまでもつべきか、協調方略とオントロジーをいかに関連すべきか、などがある。また現在 QUIK の応用として、比較文学研究のために文学データベース (ケルト文学のフェアドラ伝説) を検討しているが、本稿で検討したオントロジーの有効性もここで評価する予定である。

参考文献

- [1] Tom Gruber, "What is an Ontology?" (<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>).
- [2] Tom Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol.5, no.2, pp.199-220, 1993. (<http://ksl-web.stanford.edu/knowledge-sharing/papers/ontologia-intro.ps>)
- [3] Tom Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *Int. Workshop on Formal Ontology*, (Stanford Knowledge Systems Laboratory Report KSL-93-04), 1993. (<http://ksl-web.stanford.edu/knowledge-sharing/papers/onto-design.ps>)

- [4] JIPDEC (編), "大規模知識ベースに関する調査研究 — オントロジー工学に関する調査研究," 1997年3月.
- [5] JIPDEC (編), "大規模知識ベースに関する調査研究 — オントロジー工学に関する調査研究," 1998年3月.
- [6] 溝口理一郎, 池田満, "オントロジー工学序説 — 内容指向研究の基盤技術と理論の確立を目指して," *人工知能学会誌*, vol.12, no.4, pp.559-569, 1997.
- [7] 溝口理一郎, "形式と内容 — 内容指向人工知能研究の勧め," *人工知能学会誌*, vol.11, no.1, pp.50-59, 1996.
- [8] 中井康裕, 三宅孝典, 能勢隆雅, 横田一正, 劉渤江, "複数の包摂関係の矛盾解消のための候補解の生成と環境", *情報処理学会研究会報告*, 97-DBS-114, 114-13, 1998年1月.
- [9] 緒方啓孝, 堤慎一郎, 横田一正 "QUIK メディエータでの問合せ機能の拡張", *情報処理学会第55回全国大会*, 4AA-10, 1997年9月.
- [10] ● Ontology resources:
<http://mnemosyne.itsc.it:1024/ontology.html>
● The Ontology Page:
<http://www.medg.lcs.mit.edu/top/>
● Ontology: A Resource Page:
<http://www.csi.uottawa.ca:80/dept/Ontology/>
- [11] Gio Wiederhold, "Scalable Knowledge Composition (SKC)," <http://www-db.stanford.edu/SKC/proposal.html>.
- [12] Gio Wiederhold, "An Algebra for Ontology Composition", *Proc. 1994 Monterey Workshop on Formal Methods*, Sep., 1994. (<http://www-db.stanford.edu/pub/gio/1994/onto.ps>)
- [13] K. Yokota, T. Nishioka, H. Tsuda, and S. Tojo, "Query Processing for Partial Information Databases in *QUIXOTE*", *Proc. TAI'94*, pp.359-365, New Orleans, Nov., 1994. (<http://alpha.c.oka-pu.ac.jp/~yokota/paper/ictaicr.ps>).
- [14] Kazumasa Yokota, Yutaka Banjou, Takashi Kuroda, and Takeo Kunishima, "Extensions of Query Processing Facilities in Mediator Systems", *Proc. KRDB'97*, pp.17.1-8, Greece, Aug. 30., 1997. (<http://alpha.c.oka-pu.ac.jp/~yokota/paper/krdb.ps>)