構成的オントロジーによるオントロジーの多重化と分散化

Multiple and Distributed Ontologies by Constructive Model of Ontology

武田 英明　　飯野 健二　　西田豊明

Hideaki Takeda　　Kenji Iino　　Toyoaki Nishida

奈良先端科学技術大学院大学 情報科学研究科

Graduate School of Information Science, Nara Institute of Science and Technology

In this paper, we discuss model of ontology for distributed and heterogeneous knowledge-base systems. We model ontology as combination of aspects each of which can represent a way of conceptualization. Aspects are combined either as *combination aspect* which means integration of aspects or *category aspect* which means choice of aspects. Since ontology by aspect allows heterogeneous and multiple descriptions for phenomenon in the target world, it is appropriate for heterogeneous knowledge-base systems. We show translation of messages as a way of interpreting multiple aspects. A translation agent can translate a message with an aspect into a message with the other aspect by analyzing dependency of aspects.

1　はじめに

大規模知識ベースシステムの実現の一方法として、多数の異なる知識システムの協調による実現という方法が近年研究されている。我々はこの方法に基づき知識コミュニティと呼ぶ知識の共有と再利用の枠組みを提唱して、様々な側面からこの枠組みのシステム化を行なってきた (例えば [5])。

異なる知識システム間での知識の共有と再利用で重要になるのが、オントロジーの共有である。ここでいうオントロジーとは対象領域の概念化の結果であり、概念と概念間の関係によって構成される体系である。

共有オントロジーを研究する場合、主に単一オントロジーによって行うものが多い (例えば [2])。しかし、実際のシステムを構築する場合、対象は同一でも複数の概念化が行なわれていたり、逆に同一の概念に見えても、別のものを指していることも多い。そこで、本研究ではアスペクトという単位を考え、アスペクトを単位としてオントロジーを構築することで、異なる概念化からの概念の共存や、異なる概念の同一化を可能とすることを試みる。

2　アスペクト

オントロジーは対象世界を記述する体系であるが、それは対象世界の概念化 (conceptualization) の結果である。しかし、単一の概念化によって、記述の詳細度の幅をある記述や広範な対象世界の記述を行なうことは困難であり、実際的ではない。実際には、我々は複数の概念化を適宜組み合わせることによって、対象世界を、広い範囲を詳細度の幅がある形で、とらえている。オントロジーの構築のおいても、この複数の概念化の利用を反映した方法をとる必要がある。本研究では、一つの概念化による概念の体系をアスペクトと呼び、オントロジーをアスペクトの組み合わせによって構成する。

ここではアスペクトを基本 (atomic) アスペクトと複合 (composite) アスペクトに分けて定義する。基本アスペクトは、他のアスペクトとは独立に定義されるもので、アスペクトの名前と用いられる概念の定義および概念間の関係からなる。これに対して、複合アスペクトはアスペクトの組み合わせによる新たなアスペクトの構成を行なうもので、他のアスペクトを利用して定義される。複合アスペクトはさらに 2 種類あり、異なる対象領域のアスペクトを合わせることによって定義する組み合わせ (combination) アスペクト、対象領域が重なり合うアスペクト間の関係によって定義されるカテゴリー (category) アスペクトからなる。前者は基本的には二つのアスペクトの和を構成するものであるに対して、後者は異なる概念化の関係を記述している。例えば、旅行に関するオントロジー「旅行アスペクト」は「観光ホテル・アスペクト」と「行程アスペクト」などからなる組合せアスペクト、「ホテル・アスペクト」は「観光ホテル・アス

連絡先: 奈良先端科学技術大学院大学 情報科学研究科, 〒 630-01 生駒市高山町 8916-5, 電話:07437-2-5262, fax:07437-2-5269,
E-mail: takeda@is.aist-nara.ac.jp,

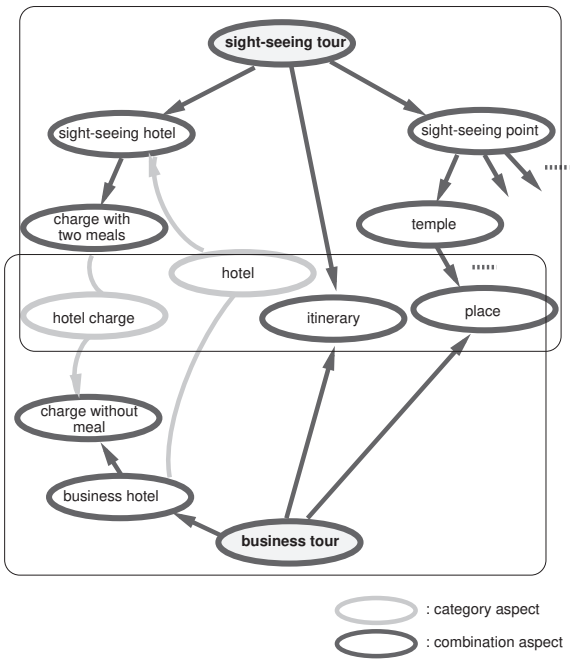

図 1: 共有可能な複数オントロジー

基本アスペクト

(**define-aspect** <アスペクト名>
Ontolingua definitions are here
...
)

組み合わせアスペクト

(**define-aspect** <アスペクト名>
(**:include** <要素アスペクト名>*)
(**:rename** (<要素アスペクト名>*!*<述語名> . <述語名>*)*)
Ontolingua definitions are here
...
)

カテゴリ・アスペクト

(**define-category-aspect** <アスペクト名>
(**:include** <要素アスペクト名>*)
(**:category-type** <カテゴリ・タイプ>)
(**:rename** (<要素アスペクト名>*!*<述語名> . <述語名>)*)
(**:default-aspect** <デフォルト・アスペクト名>)
(**:translation**
(=> <変換前概念記述> <変換後概念記述>)
([**:query-precedence** <質問文変換時条件>]
[**:inform-precedence** <伝達文変換時条件]
(-> <変換前概念付帯記述> <変換後概念付帯記述>))+)*
)

太字は予約語、[]は省略可能、 + は一回以上の繰り返し、
＊は 0 回以上の繰り返しを示す。

図 2: ASPECTOL の文法

ペクト」、「ビジネスホテル・アスペクト」などからなるカテゴリ・アスペクトとして定義できる (図 1参照)。

3 アスペクトの形式的定義

ここではアスペクトは論理的枠組みによって定義することを試みる。アスペクト A はその名前と無矛盾なアスペクト理論 $T(A)$ からなる。名前を $aspect(A)$ とするとき、

$$aspect(A) \leftrightarrow T(A)$$

であると定義する。アスペクト理論 $T(A)$ には概念の定義と概念間の定義が含まれる。

$A_1, \ldots, A_n$ からなる組み合わせアスペクトは、

$$T(A_{COM}(A_1, \ldots, A_n)) = aspect(A_1) \wedge \ldots \wedge aspect(A_n) \wedge I(A_1, \ldots, A_n)$$

と定義する。ここで $I(A_1, \ldots, A_n)$ はアスペクト間理論 (inter-aspect theory) である。

これに対して、カテゴリー・アスペクトは

$$T(A_{CAT}(A_1, \ldots, A_n)) = \Diamond aspect(A_1) \wedge \ldots \wedge \Diamond aspect(A_n) \wedge I(A_1, \ldots, A_n)$$

と定義することができる。ここでは様相論理系 S4 を用いて定義している。これはカテゴリー・アスペクトは組み合わせアスペクトと異なり、利用するアスペクトの理論が同時に真であることを意味していないからである。これらの定義を用いて、アスペクト間の各種の関係を議論することができる。例えば、競合性は次のように定義される。

競合性 アスペクト A と B が競合するとは、 A と B が同一であるか、 A と B を要素として持つアスペクト C が存在するか、 A、 B それぞれが競合性を持つアスペクト A'、 B' を持つときである。

この他、強い包含性、弱い包含性、排他的カテゴリーなどを定義することができる [4]。

また、カテゴリ・アスペクトのアスペクト間理論に適当な論理式を書くことで、要素アスペクト間の関係を記述することができる。アスペクト A_1 において式 f_1 が成り立つならばアスペクト A_2 で式 f_2 が成り立つことを、

$$\Diamond(aspect(A_1) \wedge f_1) \rightarrow \Box(aspect(A_2) \rightarrow f_2)$$

とかくことができる。

4 アスペクト記述言語

前項では論理的性質について述べたが、ここでは実際にアスペクトが記述可能な言語 ASPECTOL を定義する。 ASPECTOL は Ontolingua[1] の拡張として定義される (図 2参照)。基本アスペクトはアスペクト理論に相当する Ontoligua の定義文によって定義される。。組み合わせアスペクトは利用するアスペクトの宣言をさらに含む (4(b) 参照)。カテゴリー・アスペクトは利用するアスペクトの宣言とアスペクトに含まれる概念間の関係が論理式として定義される (図 4(a) 参照)。

5 アスペクト間の変換

知識コミュニティ [3] にようなエージェントに基づく分散協調システムでは、エージェントによって必ずしも同一のアスペクトに基づいているとは限らず、この場合異なるアス

```
(define-category-aspect FEE
  (:use fee/a fee/b)
  (:category-type nil)
  (define-translation FEE
      (=> (fee/A!fee ?fee) (fee/B!fee ?fee))
    ((:query-precedence nil
      :inform-precedence nil
      (-> (fee-value ?fee ?value)
          (and (adult ?fee ?fee1)
               (student ?fee ?fee2)
               (fee-value ?fee1 ?value)
               (fee-value ?fee2 ?value))))))
    ...
 )
```

(a) Category Aspect **fee**

```
(define-aspect temple/A (TEMPLE)
  (:use fee/A)
  (define-class temple (?x)
    :def (and (has-one ?x name)
              (has-one ?x fee)))
  (define-function name (?x) :-> ?n
    :def (and (temple ?x) (string ?n)))
  (define-function temple-fee (?x) :-> ?f
    :def (and (temple ?x) (fee ?f))))
```

(b) Combination Aspect **temple/A**

図 3: アスペクト定義の例

ペクト間で知識 (ここではエージェント間のメッセージ) を変換する必要がある。

競合するアスペクトにおいては、一方のアスペクトに基づく知識を他方に変換することが可能な場合がある。可能である場合とは、知識に含まれているアスペクトに関連するカテゴリー・アスペクトに適用可能な論理式が含まれている時である。 ASPECTOL では以下のようアルゴリズムで変換を行なうことができる。なお、ここでは単純文の連言のみについての変換を対象としている。

[1] KIF によるメッセージ文から次の手順によりメッセージ・データベースを作成する。
  (a) 関数項を全て述語に置き換える
  (b) 引数項の概念定義述語をもし含まれていなければ、オントロジーを参照して追加する。

[2] オントロジーを参照して、変換記述のリストからデータベースと以下の意味でマッチする変換規則を求める。
  (a) 質問文の場合：
    i. データベース中の概念定義述語または述語の上位概念の概念定義述語が変換規則の変換後概念記述に一致する。
  (b) 情報伝達文の場合：
    i. データベース中の概念定義述語または述語の上位概念の概念定義述語が変換規則の変換前概念記述に一致する。
    ii. データベース中の述語が変換規則の変換前概念付帯記述に一致する。

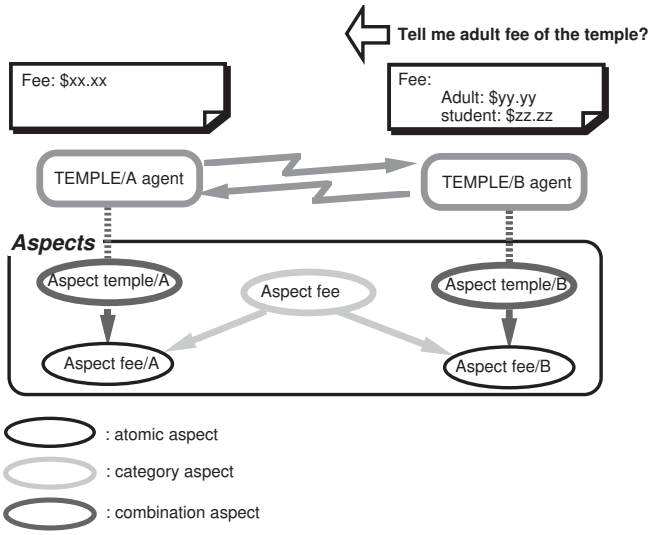

図 4: エージェント間コミュニケーションの例

このとき、文置換対 (規則を適応した時にデータベースから除かれる述語と足される述語の対)、項置換対 (文置換に伴って入れ換えるべき変数の対) を求める。

[3] 優先順位に従って、データベースに対して文置換対と項置換対を適用する。

[4] データベース中の文について、目的のアスペクトに適合しているかどうか調べる。存在しない述語は削除。

[5] メッセージを送る。情報伝達文ならば終り。

[6] (質問文の場合) 応答を受け取り、変数への束縛リストを得る。もし、有用な応答でなければ、 [2] へ行き、別の変換記述を用いる。代替変換記述がなければ、もとの質問に対して sorry を返す。

[7] 応答メッセージを元の質問メッセージの応答になるよう、 [2] の項置換対、文置換対を利用してメッセージの述語や変数を復元し、質問したエージェントに返答する。

例えば、図 4に示すような、料金の概念化が異なる二つの「寺」エージェント間のコミュニケーションを考える。今、料金区分という概念があるエージェントが料金区分のないエージェントに質問をした場合、図 5に示すような手順により、料金区分のないメッセージに直される。返答はこの逆の変換が行なわれる。

6 知識コミュニティ KC_0 における実装

知識コミュニティ KC_0 ではアスペクト間の変換機構を導入することで、メッセージ仲介機構 [6] をより効果的に利用することができる。仲介とアスペクト変換を組み合わせた場合のメッセージのやり取りを以下に示す。ここでは、 5 節

```
(temple ?x)
(name ?x ?y)
(fee ?x ?f)
(adult ?f ?f1)
(student ?f ?f2)
(fee-value ?f1 ?v1)
(fee-value ?f2 ?v2)
(< ?v1 500)
(< ?v2 400)
```

(a) The given message

```
(temple ?x)
(name ?x ?y)
(fee ?x ?f)
(adult ?f ?f1)
(student ?f ?f2)
(fee-value ?f1 ?v1)
(fee-value ?f2 ?v2)
(< ?v1 500)
(< ?v2 400)
(string ?y)
(temple-fee ?f)
(temple-fee-elm ?f1)
(temple-fee-elm ?f2)
```

(b) Adding class definitions

```
(temple ?x)
(name ?x ?y)
(fee ?x ?fee)
(fee-value ?fee ?value)
(< ?v1 500)
(< ?v2 400)
(string ?y)
(temple-fee ?f)
(temple-fee-elm ?f1)
(temple-fee-elm ?f2)
```

(c) Applying a translation formula

```
(temple ?x)
(name ?x ?y)
(fee ?x ?fee)
(fee-value ?fee ?value)
(< ?v1 500)
(< ?v2 400)
(string ?y)
(temple-fee ?f)
```

(d) Removing unnecesary literals

図 5: 変換例

で述べた変換機構は translator というエージェントによって行なわれる。また、オントロジーはアスペクトを単位として ontology server と呼ばれるエージェントで管理され、概念の定義、そのアスペクト、それを用いるエージェントなどの情報を提供する。

1. 宛先の指定されていないメッセージは facilitator によって mediator へと送られる (図 6 ①, ②)。
2. mediator は agent recommender へそのメッセージに答えられると期待されるエージェントのリストを質問し、候補となるエージェントのリストを得る (図 6 ③)。
3. mediator は得られた候補に対して順時メッセージを送信する。この際に、オントロジーの異なるエージェントへ質問する場合は、 translator を介して質問を送る (図 6 ④, ⑤)。
4. translator は ontology server に問い合わせることで、メッセージの属するアスペクトやカテゴリ・アスペクトにある変換規則を得る。これを適用することでメッセージを変換する。
5. mediator は応答のうちで有意なものの中から必要なものを集め、もとの質問者へ返答する (図 6 ⑥, ⑦)。

7 結論と展望

本研究では柔軟なオントロジー構築を目指して、アスペクトという単位によるオントロジーの構築とその利用について述べた。アスペクトによるオントロジー構築は (i) 範囲

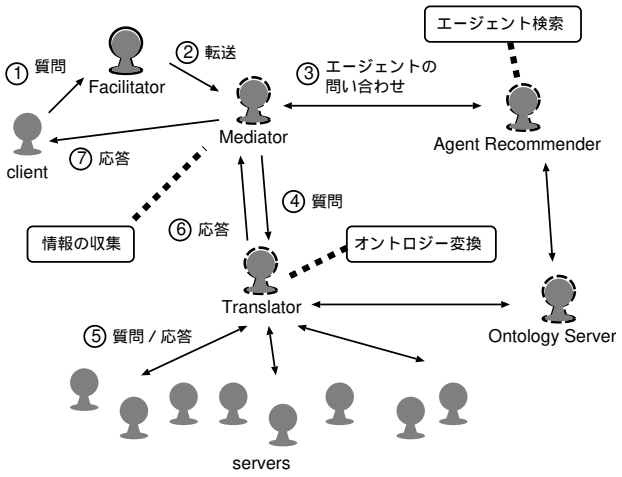

図 6: 質問の場合の仲介

を明示的に限定したオントロジー構築による構築の容易性と信頼性の向上、 (ii) 構成的にオントロジーが構築可能、 (iii) 例えば概略的定義と詳細定義の混在など、概念の多重性を許したオントロジーも可能、などの利点がある。

しかし、今回の研究では、 (i) アスペクト変換記述および変換方法が必ずしも一般的ではない、 (ii) 変換記述の作成が容易ではない、などの問題がある。今後はより一般的な変換記述やその自動獲得などを研究していく必要がある。

参考文献

[1] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.

[2] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSl 93-04, Knowledge Systems Laboratory, Stanford University, August 1993.

[3] T. Nishida and H. Takeda. Towards the knowledgeable community. In *Proceedings of International Conference on Building and Sharing of Very Large-Scale Knowledge bases '93*, pp. 157–166, Tokyo, 1993. Japan Information Processing Development Center.

[4] Hideaki Takeda, Kenji Iino, and Toyoaki Nishida. Ontology-supported agent communication. In *Working notes of 1995 AAAI Spring Symposium on Information Gathering in distributed environments*, pp. 157–165, 1995.

[5] 高岡和宏, 武田英明, 西田豊明. 知識コミュニティにおける知識メディア — virtual tour. 人工知能学会全国大会 (第 8 回) 論文集, pp. 283–286, 1994.

[6] 武田英明, 飯野健二, 西田豊明. 知識コミュニティ kc0 における知識共有メカニズム. 人工知能学会全国大会 (第 8 回) 論文集, pp. 279–282, 1994.