

# 知識コミュニティにおけるメッセージ仲介機構

## Mediation Mechanism in the Knowledgeable Community

飯野 健二      武田 英明      西田 豊明  
Kenji Iino   Hideaki Takeda   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

### Abstract

The Knowledgeable Community is a framework of knowledge sharing and reuse based on a multi-agent architecture. In this paper, we focus on the organizational structure that facilitates mediation between those agents requesting for a service and those providing the service. Mediation is necessary to realize flexibility and scalability of large scaled, distributed knowledge base system. In particular, we discuss mediation with ontology that is performed by mediator and ontology server. Ontology server manages ontology and relations between part of ontology and agents, and can suggest feasible agents to process given messages. Mediator routes messages with unspecified addressees to acceptable agents which could process them, then gathers and forwards their replies to original requester.

## 1 はじめに

一般に現実の問題を解くためには様々な分野の知識を必要とすることが多い。従って、広範な知識を持つ大規模知識ベースが必要となる。我々は大規模知識ベースの実現としてマルチエージェント系による知識の共有・再利用の枠組、知識コミュニティを提唱している [6]。知識コミュニティでは、情報処理単位であるエージェント同士がメッセージを交換し合うことにより、知識共有・協調処理を行なう。

マルチエージェント型システムにおいて、拡張性の高いシステムを構成するには、エージェント同士の関係が可変的であるのが望ましい。本論文では、知識コミュニティにおいて、エージェント間の緩やかな結合を実現するための仕組みである仲介 (mediation) について、その実現の方法を提案する。

## 2 エージェント組織

知識コミュニティはマルチエージェント型の情報処理方式を採用しており、個々のエージェントは相互に

メッセージを交換することにより、知識の伝達を行なう。現在の知識コミュニティでは、メッセージプロトコルとして KQML (Knowledge Representation and Manipulation Language) [2] を、メッセージ内容の表現として KIF (Knowledge Interchange Format) [4] を採用している。エージェント組織の構成は連邦アーキテクチャ [1] による。このアーキテクチャでは *facilitator* (協調促進器) と呼ばれるエージェントがメッセージの送受信やエージェントの管理、ネットワークの管理などのサービスを提供する (図 1)。

知識コミュニティの指向する分散型知識ベースシステムでは、大規模なシステムを構築する際に、情報処理単位であるエージェントが個々に他のエージェントに関する情報 (エージェントの機能・通信プロトコルなど) を管理することは非常にコストがかかる。またシステムに拡張性を持たせるためには、エージェント間の論理的な関係を可変的なものとする方がよい。そこでシステム全体の構成や各エージェントの能力など、エージェントの環境に関する知識 (外部モデル)

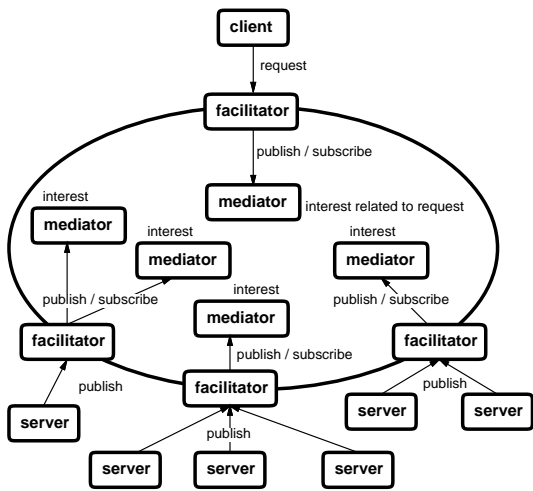


図 1: 知識コミュニティにおけるエージェント組織

はエージェントの外部で保持あるいは管理することを考える。このとき環境知識を管理する特別なエージェントが存在して、各エージェントはそこに問い合わせることにより、環境知識を利用することができる。このことにより、エージェント間の結合は緩やかなものとなり、システムに可変性・拡張性を持たせることができる。

この際、個々のエージェントは他にどのようなエージェントが存在しているのか必ずしも知っている必要はない。言い替えると、あるエージェントが質問や要求を他のエージェントに対して行なう際に、どのエージェントにメッセージを送ればよいのかを個々のエージェントが必ずしも知っている必要はないということである。これらの宛先のないメッセージを、それを処理するのに適当なエージェントに伝達することを仲介 (mediation) と呼ぶ。仲介は、システムの構成やエージェントの能力・エージェント間の関係などの環境知識を用いて行なわれる。実際の仲介処理は mediator と呼ばれる特別なエージェントが行なう (図 1)。

仲介処理の際のメッセージの送信は次のようにして行なわれる。まず、質問を発するエージェントからメッセージが送信される。メッセージは facilitator が受けとり、メッセージの宛先が特定されている場合は宛先のエージェントに対してメッセージを送る。メッセージの宛先が特定されていない場合、facilitator はメッセージを mediator に転送する。Mediator はメッセージを受けると、環境知識を用いてメッセージに対して適当なエージェントを定め、そのエージェントへメッセージを転送する。

### 3 Ontology を用いた仲介

#### 3.1 Ontology server

複数のエージェント間で知識交換を行なうためには、その知識の背景にある概念体系 (ontology ; オントロジー) をエージェント間で共有していなければならない。オントロジーはエージェントの環境知識の一種であり、その記述を管理するエージェントを ontology server と呼ぶ。Ontology server はエージェント間で共有される概念体系の記述の他、エージェントとエージェントの持つ知識の背景となる概念との関係を保持している。これは個々のエージェントの処理可能な領域・処理可能な概念を保持しているともいえる。

オントロジーの表現形式はフレーム型の表現によるオントロジーを用いている。ここではオントロジー記述言語として Ontolingua[5] を用いている。オントロジーには class 間の superclass, subclass 関係や各 class に付随するスロット名、スロット値の class などが定義されている (図 3)。

#### 3.2 Ontology server を利用した仲介

ここでは、ontology server を用いた仲介の方法を述べる。この処理は次のような手順で行なわれる (図 2)。

1. 仲介を必要とするメッセージが facilitator を経由して mediator に送られる (メッセージ 1, 2)。
2. Mediator は ontology server に対し、メッセージ内容と関連のあるエージェントの問い合わせを行なう (メッセージ 3)。
3. ontology server では次の手順でメッセージに関連するエージェントのリストを作成し、mediator へ返答する (メッセージ 4)。

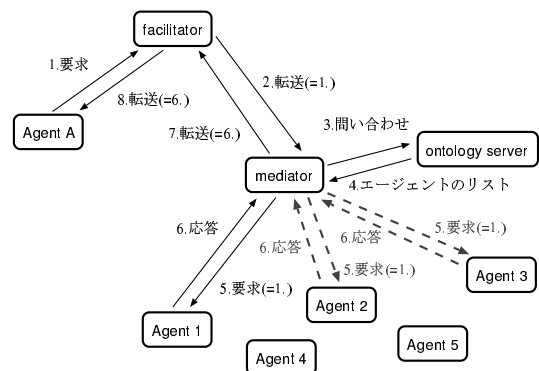


図 2: オントロジーによる仲介

- i) メッセージに記述されている概念のクラスを特定する。
  - ii) 3i) で特定されたクラスに関連するエージェントがあれば、メッセージに対するエージェントの第一候補とする。
  - iii) 3i) で特定されたクラスの下位クラスに関連するエージェントがあれば、それを次候補のエージェントとする。
  - iv) 以下下位クラスがなくなるまで再帰的に 3iii) を実行する。
  - v) 3i) で特定されたクラスの上位クラスに関連するエージェントがあれば、それを次候補のエージェントとする。
  - vi) 以下上位クラスがなくなるまで再帰的に 3v) を実行する。
4. mediator は ontology server から送られたエージェントのリストをもとに、第一候補から順に(あるいは全ての候補に一齐に)もとのメッセージを送る(メッセージ 5)。有意な返信が得られれば(メッセージ 6)、それをもとのメッセージの発信元へ送る(メッセージ 7, 8)。

## 4 仲介の例

### 4.1 Ontology

図 3にあるオントロジーを例として用いる。図において、各黒枠が一つのクラス概念を示し、その間は class-subclass 関係で結ばれている。各 class が持ちうるスロットは矢印で示されており、その値の class を斜体文字で示してある。また、同時にオントロジー上の各概念に関連するエージェントも示している(薄い枠線)。

### 4.2 仲介の過程

具体的にどのようにして仲介が行なわれるか質問の例をあげて示す。質問の表現形式は KQML 及び KIF による。

#### 質問 (1)

```
(broker-one :content
  (ask-one :content (and (hotel ?x)
                        (name ?x "Nara-hotel")
                        (nearest-station ?x ?y))
    :language KIF)
  :reply-with q1)

(奈良ホテルの最寄り駅を知りたい)
```

質問 (1)において、(hotel ?x)とは変数 ?x がクラス hotel のインスタンスであることを意味しており、(nearest-station ?x ?y)は ?y が ?x の最寄り駅であることを意味している。また ask-one は質問を、broker-one は仲介依頼を意味する KQML の performative である。

質問 (1)が発信されると、メッセージは facilitator を経由して mediator に送られる。Mediator は ontology server に対してメッセージに答えることのできるエージェントを尋ねる。Ontology server はメッセージ内容からクラス hotel である変数 ?x に関する質問であると判断し、関連するエージェントがあるか調べる。ここではクラス hotel に関連する“hotel-agent”と、クラス hotel の上位概念である place に関連する“chiri-agent”の順で関連が強いと判断され(図 3参照)、そのリストを mediator に返信する。次に mediator はまず“hotel-agent”に対して質問を行なう。ここで“JR-Nara”という有意な返答を得たので、これを質問元へ送信して処理を終了する。

#### 質問 (2)

```
(broker-one :content
  (ask-one :content (and (visit-place ?x)
                        (name ?x "today-ji")
                        (nearest-station ?x ?y))
    :language KIF)
  :reply-with q1)

(東大寺の最寄り駅を知りたい)
```

ここでは東大寺を観光地(visit-place)と捉えている。質問(2)のメッセージも同様に mediator に送られ、mediator は ontology server に対して適当なエージェントを尋ねる。Ontology server はメッセージがクラス visit-place である変数 ?x に関する質問であると判断して、関連するエージェントを調べる。ここでは、“sight-seeing-agent”(visit-place に関連)、“nara-park-agent”, “akishino-ji-agent”, “today-ji-agent” (visit-place の下位クラスに関連)、“chiri-agent” (visit-place の上位クラスに関連)の順でエージェントのリストが返される。Mediator はこの順にエージェントに対して質問を行なう。“sight-seeing-agent”, “nara-park-agent”, “akishino-ji-agent”, はいずれも東大寺の最寄り駅を知らず、知らないあるいは処理できないことを意味する sorry performative を返信する。“today-ji-agent”が知っているとする

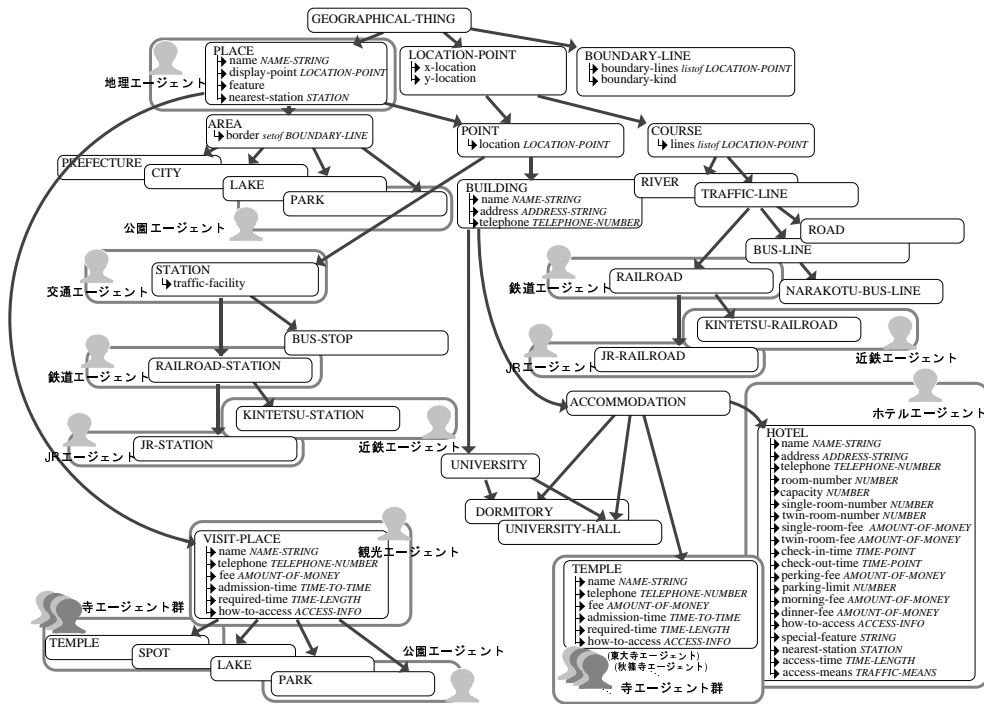


図 3: 地域情報に関するオントロジーの例

ると、ここで“Kintetsu-Nara”という答が得られ、これを質問元へ送信して処理が終了する。

## 5 おわりに

本論文では協調型大規模知識ベースの実現としての知識コミュニティにおいて重要な役割を果たす仲介機構について述べ、その一手法としてオントロジーを用いる方法を提案した。大規模な分散協調システムにおいて、仲介機構を用いることにより、柔軟性・拡張性の高いシステムを構築することができる。オントロジーを用いる仲介は、エージェントの能力をエージェント間で共有されるオントロジーと対応づけ、オントロジー上で複数のエージェントを関係づけるものである。

なお現状において、実際に 93 の class 及び 75 の relation を持つオントロジーを作成し、mediator と ontology server を用いて 4.2 節で挙げた例題について処理することができた。今後は、具体的な問題に対して mediator, facilitator を含めた問題解決システムの構築を進め、システムの大規模化をはかることを検討している。また、複数のオントロジーが存在する場合や、他の手法による仲介機構についても検討する必要がある。

## 参考文献

- [1] R. S. Patil, R. E. Fikes, P. F. Patel-Schneider, D. McKay, T. Finin, T. R. Gruber, and R. Neches. The DARPA Knowledge Sharing Effort: Progress report. In Charles Rich, Bernhard Nebel, and William Swartout, Ed., Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, Cambridge, MA, Morgan Kaufmann, 1992.
- [2] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritson, D. McKay, J. McGuire, P. Pelavin, S. Shapiro, and C. Beck. Specification of the KQML Agent-Communication Language. Enterprise Integration Technologies, Palo Alto, CA, Technical Report EIT TR 92-04, 1992. (Updated July 1993).
- [3] M. Cutkosky, R. Englemore, R. Fikes, T. Gruber, M. Genesereth, W. Mark, J. Tenenbaum, and J. Weber. PACT: An experiment in integrating concurrent engineering systems. IEEE Computer, January 1993.
- [4] M. R. Genesereth and R. E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Computer Science Department, Stanford University, Technical Report Logic-92-1, March 1992.
- [5] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Stanford University, Knowledge Systems Laboratory, Technical Report KSL 91-66, March 1992. Revision.
- [6] Toyoaki Nishida and Hideaki Takeda. Towards the Knowledgeable Community. In *Proceedings of International Conference on Building and Sharing of Very Large-Scale Knowledge bases '93*, pages 157-166, Tokyo, 1993. Japan Information Processing Development Center.
- [7] 飯野健二, 武田英明, 西田豊明: “知識コミュニティにおける仲介機能”. 第 3 回マルチエージェントと協調計算ワークショップ. 1993.