知識コミュニティにおける仲介機能

Mediation in the Knowledgeable Community

武田 英明* 飯野 健二† 西田 豊明‡

Summary. 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. We describe several techniques of mediation. 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. We illustrate mediating processes with a testbed system for travel planning.

1 はじめに

我々は、大規模知識ベースの実現として、知識の生産・共有・利用の枠組み**知識コミュニティ** (the Knowledgeable Community) を提唱し、そのプロトタイプの開発を進めている [NT93] [Nis93]。

大規模知識ベースを実現する場合、二つの方法が考えられる。一つの方法は一つの知識表現フォーマットを準備して、 そのフォーマットに従って知識を収集して、単一のシステムとして大規模知識ベースシステムを実現するものである。これに対して、知識共有アプローチでは、知識は共通の表現フォーマットで収集・記述されるのではなく、知識は個々の知識システムに収集・記述され、全体の統合は知識システム間の相互作用によって達成される。

実際の知識は分野や目的により、その表現形式・表現内容が多種多様であり、一つの表現フォーマットで統一するのは困難である。また、集積した知識の規模が大規模になればなるほど、その管理は困難になる。このため、われわれは後者のアプローチを取り、マルチエージェントシステムという形で大規模知識ベースを実現する知識コミュニティ・プロジェクトをはじめている。本研究ではこの**知識コミュニティ**の中で重要な役割を果たす mediation(仲介) について、その意味と実現方法について述べる。

*Hideaki Takeda, 奈良先端科学技術大学院大学
†Kenji Iino, 奈良先端科学技術大学院大学
‡Toyoaki Nishida, 奈良先端科学技術大学院大学

2 知識コミュニティの枠組み

知識コミュニティは、知識の断片を担った**エージェント**から成る**マルチエージェント**型の情報処理方式を採用している。各エージェントは自律的・部分独立性を持つ**知識**である。情報処理として見た場合、エージェントは様々なタイプの情報 — 情報サービス、推論エンジン、計算サービス、問題解決サービス — を提供する。知識コミュニティ全体の情報処理はエージェントの相互作用によって実現される。

知識コミュニティの実現は KC_0 と KC_1 の2段階に分けられる。第1段階の KC_0 は主として手作りで構築し、第2段階の KC_1 には学習や自己組織化の機能を持たせて、自律的に成長させることを目指す。現在は、KC_0 に焦点を当てて、試作を行なっている。

2.1 エージェントの設計指針

知識コミュニティの目標は全体として自律的 / 自己組織的知識システムとして働くことであるが、それは必ずしも個々のエージェントが全てそのような条件を満たすことを意味するわけではない。**知識コミュニティ**におけるエージェントは、極めて知的な応答をするものから、データベースシステムのような単純な応答しか許さないものまで多様なものを許すべきである。以下の項目はコミュニティの成員に期待されている能力である。

- **Communication 能力**

  他のエージェントと適切な相互作用ができるための最低限の規約である。communication のための共通のプロトコルを理解しなければならない。

- **自己の概念体系の公開**

  各エージェントの知識が立脚する概念体系は公開されている必要がある。また、それらは他のエージェントが利用できる形でなければならない。

- **自己の能力の公開**

  各エージェントはなんらかの方法で自分の持つ能力をコミュニティに対して公開しなければならない。これはエージェントが自分の能力として外部に知らしめたい能力であり、網羅的である必要もないし、場合によっては意図的に誤りの可能性もある。

人間のコミュニティのアナロジーでいえば、人間間で有用な会話が成立するには、会話能力があること (1番目) はもちろん、ある程度背景が共通していること (2番目)、さらに自分の役割 / 能力を理解していること (3番目)、といえよう。ここで注意する点は、外界のモデルを持つことは必須ではない、ということである。自分以外にどんなエージェントがいて、どのような処理ができるかは必ずしも知っている必要はない。

現在の**知識コミュニティ**の implementation では communication 能力実現のための共通なプロトコルとして、メッセージ伝達行為の表現である KQML(Knowledge Query and

Manupulation Language) [FWW+92]を利用している。KQMLではメッセージ伝達行為を表現するために、1対1メッセージ伝達を基本として、メッセージの種類を表すkeyword、送り先や送り元などのパラメータ、メッセージ本体からなるフォーマットを規定している。メッセージ内容の表現としてはKIF(Knowledge Interchange Format) [GF90]を利用している。KIFは一階述語論理を基本とする知識表現言語であり、知識の交換に利用されるものである。また、概念体系の表現としては、ontologyとして表現し、具体的にはOntolingua [Gru92]を採用している。Ontoligua はontologyを書くための言語で、KIFの表現を拡張して、classや関係を定義できるようにしている。

2.2 エージェント間結合の方法

エージェント群の中には、お互いに依存度の強い密な結合をするものから、単発的なメッセージのやり取りをする疎な結合まで様々である。このため、エージェント間結合を実現する方法は柔軟である必要がある。**知識コミュニティ**では以下の2点をエージェント間結合を実現する方法とする。

- **Ontologyの共有**

  ontologyを顕在化かつ外在化することにより、各エージェントのontologyを他のエージェントが知ることができる。エージェント間でontologyの重なり具合をみることでエージェント間の関係を知ることができる。

- **Mediation**

  先に述べたように、個々のエージェントは必ずしも他のエージェントを知っているとは限らない。そのような場合でも、メッセージ伝達がうまく行くように、メッセージの仲介をする特別なエージェントmediatorを用意する。Mediatorは宛先などがない不完全なメッセージの宛先の補完やメッセージのフォーマットを変換を行なうことにより、メッセージ伝達を成立させる。

この他、特定のエージェント群においては、クラス階層といった、エージェント群の構造化が行なわれる。

2.3 エージェント構成

知識コミュニティでは連邦アーキテクチャ [PFPS+92] [CEF+93]によってエージェントを構成する。ネットワークの各セグメントにはfacilitatorと呼ばれる特別な役割を持つエージェントが必ず一つ存在する。facilitatorは生成、起動、休止、消滅などのエージェントの行動の把握とその補助、およびエージェント間のメッセージ転送の把握とその補助を行なう。各セグメントに属するエージェントは全てのメッセージをまず、facilitatorに送る。Facilatorはそのメッセージの送り先を解釈して、適切なエージェントに転送する。また、送り先が書か

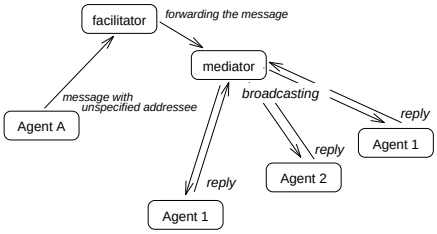

図 1 Mediation by broadcasting

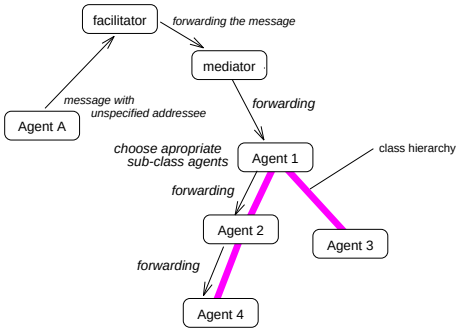

図 2 Mediation by class hierarchy

れていない場合あるいは正しくない場合、facilator はそのメッセージを mediator に委託する。Mediator は faciliator から委託されたメッセージに対して、そのメッセージを処理するのに適切なエージェントを探し、そのエージェントにメッセージを送る。

3 Mediation

3.1 Mediation の方法

知識コミュニティで行なわれる Mediation(仲介) には以下に挙げる方法が考えられる。

(1) Broadcasting による仲介: 最も単純な方法で、全てあるいは一部のエージェントに委託されたメッセージを処理可能かを聞く方法である。エージェントの返答結果をみて、最も適当な返答を元のエージェントに返す (図 1参照)。

(2) Class hierarchy による仲介: エージェントにクラス - サブクラス関係やクラス - インスタンス関係があるときは、上位のエージェントに転送して、下位のエージェントへの仲介を委託する。メッセージを受けとった上位のエージェントは適切な下位エージェントを選択して、メッセージを送る (図 2参照)。

(3) Ontology による仲介: Ontology を ontology server に聞くことにより、仲介する。Ontology server は概念同士の ontological な関係、概念とその概念を表現するエージェント間の関係、エージェント間の ontological な関係を保持している。Ontology server は与えられたメッセージの内容の中の概念に対するエージェントをあれば返す。直接対応するものがなければ、関係する概念あるいはその概念に関係するエージェントを返答する。Mediator はこの答を利用して、送り先を決定する (図 3参照)。

(4) Heuristics による仲介: Class hierarchy に代表される ontological 関係は整合的である半面、曖昧な表現の理解に必要な連想等の関係を含み得ない。このような表現の処理のためには heuristics に依存した仲介を行なう。例えば concept associators は concept space を利用して、メッセージと概念間の距離を計算して、関係する概念を見つけ出す。Mediator

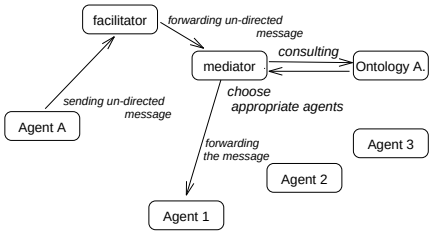

図 3 Mediation with ontology server

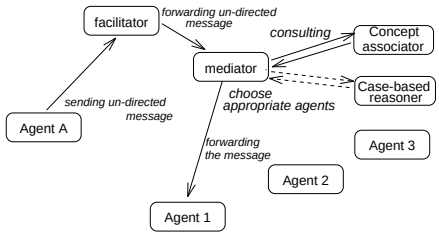

図 4 Mediation by heuristics

はこれを利用して送り先を決定する (図 4参照).

(5) エージェント能力記述による仲介: 各エージェントは自己の能力について記述を facilitator に対して公開する。このエージェントの能力記述はエージェントの特徴の記述や受け付け可能なメッセージなどが指定されたフォーマットに従って記述されている。Mediator はこの記述を集め、与えられたメッセージに適切なエージェントを選択する。この仲介は heuristics による仲介と組み合わせることができる。

3.2 Ontology server を利用した仲介

以上で挙げた方法のうち、ontology を利用した仲介について詳しく述べる。

ここでは現在構築中の KC-Kansai プロジェクトの知識コミュニティを用いて説明する。KC-Kansai は知識コミュニティの概念を実証するための testbed として構築中の地域情報システムである。このシステムでは、関西訪問のプラン立案に必要となる、関西の交通機関、宿泊施設、地域に固有のさまざまなサービス、旅行ヒントなどについてユーザの要求に応えることを目的にしている。例えば、関西方面の旅行のための交通手段・宿泊プランを様々な情報を組み合わせて提案する。KC-Kansai では、交通手段に関する知識・情報を司るエージェント群や宿泊施設の情報を持つエージェント群、個々の寺などの観光地の知識を提供するエージェント群などからなる。

3.2.1 Ontology

Ontology はフレーム型オントロジーに基づくとする。Ontology には class 間の superclass, subclass 関係や各 class に付随するスロット名・スロットの値の class などが定義されている。

Ontology に関する情報を提供するエージェントが ontology server である。ontology server は、上記のようなクラス間の関係やスロットに関するものの他、クラスに関係づけられたエージェントへのタグを持つ。この ontology に関する情報のうち、各エージェントに関わる部分は各エージェントの能力記述として知識コミュニティに公開されたものを用いる。Ontology server では、各エージェントとそれらが用いているオントロジーの部分の関連性を、

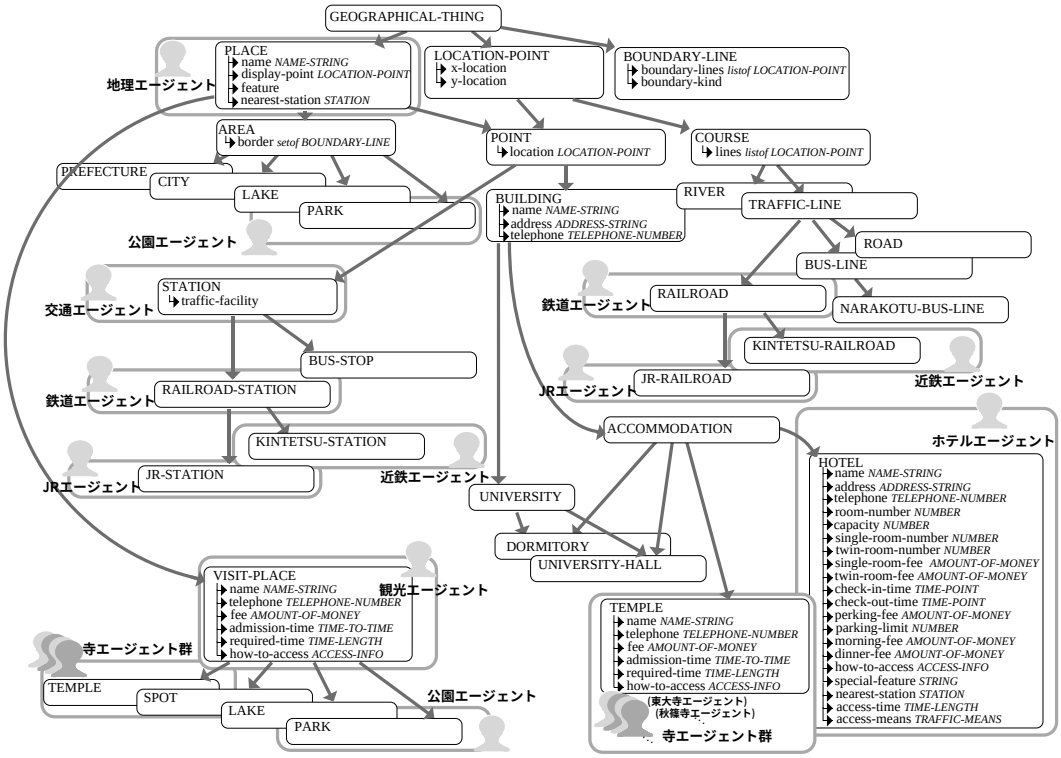

図 5 Ontology on travel plan

ontology の要素に対するタグ (エージェント・タグ) として記述する。

Ontology server が mediation において果たす役割は、ある宛先不明のメッセージの内容についてオントロジー情報を用いて、メッセージにおいて問題となっているクラスを判断し、クラスに強く関係づけられているエージェントを特定することである。

図 5 に KC-Kansai で用いられる ontology の一部を示す。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。また、同時に ontology 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)。

3.2.2 Mediation の手順

ontology server を利用した mediation は次のような過程で行なわれる。

1. client は mediation を依頼するメッセージを facilitator に送る。
2. facilitator はメッセージを mediator に対して転送する。
3. mediator では :content の内容について処理能力を持つ可能性のあるエージェントを知るため、ontology server に対して問い合わせを行なう。
4. ontology server では次の手順で該当するエージェントのリストを作成する。
   (a) KIF 表現から、メッセージで問題となっているクラスを特定する。

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

図 6　message(1)

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

図 7　message(2)

(b) 特定されたクラスに関連するエージェントへのタグが付随していれば、それを第一候補として保管する。

(c) 特定されたクラスの下位クラスにエージェントへのタグが付随していればそれを次の候補として保管する。

(d) 以下下位クラスがなくなるまで検索を続ける。

(e) 特定されたクラスの上位クラスへ順に検索を行ない、エージェントタグが付随していればそれを次の候補として保管する。

5. ontology server からのエージェントのリストをもとに、第一候補から順にメッセージを送る (あるいは全ての候補に一斉に送る)。解答が得られれば、それをもとの質問者 (client) に送る。

具体的にどのようにして mediation が行なわれるか例を示す。

質問 (1) 奈良ホテルの最寄り駅を知りたい

KQML 及び KIF による表現は例えば図 6になる。 ここで、`(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いる。`(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。 `:aspect ?y` は質問者が尋ねたい情報が`?y` であることを意味している。また、`ask-one` は質問を意味し、`broker-all` は mediation 依頼を意味する KQML の performative である。

Client から facilitator に上記のメッセージが送られると、facilitator は図 7のようなメッセージにして、mediator に転送する。そして mediator はさらに ontology server に転送する。ontology server はメッセージから、クラス `hotel` である変数 `?x` に関連する質問であると判断し、ontology に付随するエージェントタグを調べる。クラス `hotel` には、"hotel-agent" というエージェントタグが付随している (図 5参照) ので、"hotel-agent" を回答として、mediator に返答する。

質問 (2) 東大寺の最寄り駅を知りたい

Mediator は同様にまず `temple` というクラスに注目する。今回の設定では寺一般のエージェントが存在せず、各寺毎にエージェントが存在する (図 5参照)。このため、これらのエージ

ェントに対して順にメッセージを送る。もし"sorry"以外の返答があればそれを答とする。しかし、この例では各寺エージェントは自身の最寄り駅情報を管理しいないので、東大寺に関するエージェントでも回答は"sorry"となる。この場合、mediatorはontologyのクラス階層を遡り、次のエージェント候補を探す。"temple"の場合、その上位概念の"visit-place"を観光エージェントが、さらにその上位概念"place"を地理エージェントが管理している(図5参照)。そこで、まず観光エージェントにこのメッセージを送る。それが失敗すると、地理エージェントにメッセージを送る。地理エージェントは場所の地理を管理しているので、このメッセージに正しく返答することができる。

4 おわりに

本論文では共有指向大規模知識ベースの実現としての知識コミュニティの構成とその中で重要な役割を果たすmediationについて述べた。Mediationは大規模な分散協調システムでは必須の機能である。ここでは、その実現方法のひとつであるontologyを用いたmediationについて説明した。これは、各エージェントをそれらが扱うontologyとして抽象化することでmediationを行なうもので、機能的・挙動的に異なるエージェントを統合的に扱うひとつの有効な方法であると考えられる。

参考文献

[CEF+93] Mark R. Cutkosky, Robert S. Engelmore, Richard E. Fikes, Michael R. Genesereth, Thoas R. Gruber, William S. Mark, Jay M. Tenenbaum, and Jay C. Weber. PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, Vol. January 1993, pp. 28–38, 1993.

[FWW+92] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, J. McGuire, P. Pelavin, S. Shapiro, and C. Beck. Specification of the KQML agent-communication language. Technical Report EIT TR 92-04, Enterprise Integration Technologies, 1992. (Updated July 1993).

[GF90] Michael R. Genesereth and Richard Fikes. Knowledge Interchange Format version 3.0 reference manual. Technical Report Logic-90-4, Computer Science Department, Stanford University, 1990.

[Gru92] T. R. Gruber. Ontolingua: A mechanism to support portable ontologies. Technical Report KSL 91-66, Stanford University, Knowledge Systems Laboratory, 1992.

[Nis93] 西田豊明. 知識コミュニティ. 北野宏明 (編), グランドチャレンジ — 人工知能の大いなる挑戦 —, pp. 176–189. 共立出版, 1993.

[NT93] 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*, pp. 157–166, Tokyo, 1993. Japan Information Processing Development Center.

[PFPS+92] 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, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*. Morgan Kaufmann, 1992.