

## システム構成の動的な変化を可能にするエージェント間通信機構

横山 和俊 箱守 聡 井上 潮

NTTデータ通信(株) 情報科学研究所

大規模・複雑化した分散システムにおいては、様々な機能を持つプログラム（エージェント）が走行し、また、エージェントの起動・停止や機能拡張などによりシステム構成が動的に変化する。このような環境において、エージェントが協調しサービス処理を実現するには、機能やインタフェースの差異を吸収し、かつ、それらの動的な変化に対応できる、柔軟なエージェント間通信機構を提供することが重要である。

本稿では、エージェント間通信機構への要求条件と課題を明らかにし、複数のデータベースを対象とする情報検索システムを例にした実現方式について述べる。具体的には、エージェント情報（機能、動作状態等）の管理方式、及び、これらの情報を用いて機能を指定した呼び出しを可能にするエージェント間通信方式を提案する。さらに、提案する方式の実装について述べる。

## 1. はじめに

低価格・高性能の計算機が普及し、これらを高速なネットワークで結合した分散システムの構築が盛んに行われている。また、インターネットをはじめとするコンピュータネットワークのインフラが整備され、ネットワークを通じた情報サービスも提供されている。例えば、データベース情報の検索サービスが代表的なものである。今後、分散システムは、ますます大規模化や複雑化が進み、複雑な相互作用を持つ大規模な分散システムに発展していくことが予想される。

このような分散システムでは、システムの構成が頻繁に変化する可能性がある。そのため、予めシステム構成の変化を予測し全体の振る舞いを規定することは困難である。したがって、処理相手や処理順序を動的に決定しながら処理を進める方式が有効である。

このような柔軟なシステムを構築するアプローチとして、自律的に動作するプログラム（エージェント）<sup>[1][2]</sup>を用いる方式がある<sup>[3][4][5]</sup>。文献[3][4]

では、エージェントが協調して処理を行うためのシステムモデルと記述言語を提案している。これらの研究では、エージェント間の協調プロトコルや交渉手法に主眼が置かれており、エージェント間の通信が容易に行われることを前提としている。また文献[5]では、黒板モデルに基づいたエージェント間通信機構を提案している。この研究では、エージェント間の通信は黒板が統一的に制御し、各エージェントは黒板のみを意識する構成により、柔軟性の高いシステムを実現している。しかし、大規模な分散システムにおいては、様々な機能や役割を持ったエージェントが走行することが考えられ、すべてのエージェントを統一的に扱うことは困難である。したがって、エージェントを用いてシステム構成が頻繁に変化する分散システムを実現するには、エージェント間の機能や役割の差異を吸収し、かつ、システム構成の動的な変化に対応できる柔軟なエージェント間通信機構を提供することが必要である。

本稿では、エージェント間通信機構への要求条件と課題を明らかにし、それらを満たす実現方式

---

Agent Communication Architecture for Dynamic Reconfigurable Systems

Kazutoshi YOKOYAMA, Satoshi HAKOMORI, and Ushio INOUE

Laboratory for Information Technology, NTT DATA Communications Systems Corporation

について述べる。具体的には、エージェント情報（機能、動作状態等）の管理方式、及び、これらの情報を用いて機能を指定した呼び出しを可能にするエージェント間通信方式を提案する。さらに、提案する方式の実装について述べる。

## 2. 分散システムの特徴と問題点

分散システムにおける将来のサービス処理は、以下に示す特徴を持つと考えられる。

### (1) サービスの高度化

現在提供されている単独のサービスだけでなく、それらを複数組み合わせたサービスが提供される。例えば、既存のデータベースサービスを組み合わせ利用し、複数のデータベースを対象にした検索代行サービスが提供される。

### (2) サービス処理のライフサイクルの短縮化

利用者の要求が多種多様になり、流行の移り変わりが速くなる。これに応えるため、新しいサービスの提供が頻繁に行われる。また、より良いサービスの提供が望まれるためサービスの機能拡張も頻繁に行われる。

これらの特徴を持つサービス処理が走行するため、将来の分散システムでは、システム構成の変更が頻繁に発生する可能性がある。システム構成の変化には、以下の2つの種類がある。

#### (A) 物理構成の変化

サービスが組み合わされるため、それらを個別に提供していたシステムが相互に接続される。また、サービス処理の高速化のため、計算機の数や種類、ネットワークの構成が変化する。

#### (B) 論理構成の変化

新しいサービス処理の提供が提供されたり、走行しているサービス処理が中止される。また、サービス処理の機能拡張が行われる。さらに、単独のサービス処理が統合され、より高度なサービス処理が提供される。これらにより、サービス処理を構成するプログラム間の通信関係や処理手順が変化する。

このような分散システムでは、予めシステム構成の変化を予測し全体の振る舞いを規定することが困難である。そのため、従来のトップダウン的な設計では、構成変更に対して以下の問題点があ

る。

#### (a) 物理構成の変化による問題点

計算機の数や種類、ネットワーク構成の変更に応じて、既に走行しているプログラムを修正する必要がある。例えば、異機種の計算機や異種のネットワークが加わった時には、それらのための通信制御方式をプログラムに組み込む必要がある。

#### (b) 論理構成の変化による問題点

サービスを構成するプログラムの処理相手や処理手順を変更するには、既に走行しているプログラムを修正する必要がある。例えば、新たに提供されたデータベース検索処理を利用するには、そのデータベース検索処理へのアクセス方法をプログラムに加える必要がある。

このうち、(a)の問題については従来から研究されている分散OS<sup>10)</sup>や分散オブジェクト管理技術<sup>11)</sup>により対処されつつある。したがって、今後の技術的課題としては(b)の問題が重要となる。

## 3. エージェントによる分散システムの構築

### 3.1 エージェントシステムのモデル

前節で述べた問題点を解決するためには、論理構成の変化に応じて、処理相手や処理手順を動的に変化させながら処理を進める方式が必要となる。これには、自律的に動作するエージェントが複数動作しそれらが協調して処理を進めるエージェントシステムによるシステム構築手法が有望である。このとき、システム内には、様々な役割を持つエージェントが走行する。

エージェントを用いた分散システムモデル例を図1に示す。例えば、利用者に最も近い立場で利用者の秘書役を行うエージェント(図1のU)や、サービス処理を提供するエージェント(図1のA,B,C,E,F,G,H,I)が存在する。また、個々のサービスを統合して、より高度なサービスを提供するエージェント(図のD)が走行すると考えられる。

エージェントシステムにおいては、論理構成の変化は以下の形態になる。

#### (1) サービスの提供/中止

サービスの提供と中止は、エージェントの起動

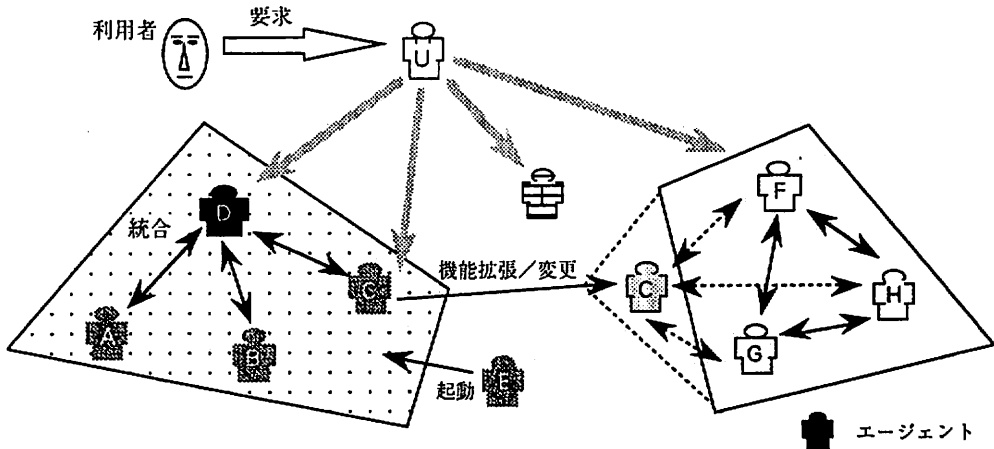


図1 エージェントを用いた分散システム

と停止によって表される。例えば、エージェントA,B,Cと同じ機能を持ったエージェントEが起動されたとする。エージェントEは自分を利用してもらうためにエージェントDから認識されなければならない。そのため、エージェントEはエージェントDを探し、自分が起動したことをエージェントDに通知する。通知されたエージェントDは、エージェントEの機能を判断し統合サービスに組み入れる。

### (2) サービスの機能拡張

サービスの機能拡張は、エージェントへの依頼方法(アクセスインタフェース)や提供する機能が変ることによって表される。例えば、エージェントBへの依頼方法が変わったときには、エージェントDへ依頼方法の変更を通知する。また、エージェントCの機能が変ったときには、エージェントDの統合サービス処理から外され、他のエージェントと処理依頼関係を持つことになる。

### (3) サービスの統合

サービスを統合するエージェントが新たに走行することによって表される。例えば、エージェントDが新設されたとき、統合対象のエージェントA,B,Cの情報を取得する。エージェントA,B,Cは、エージェントDを意識する必要がなく、これまでと同様の処理を実行すればよい。

## 3.2 エージェント間通信機構への要求条件

3.1節で示したエージェントによる分散システムモデルにおいては、エージェント間通信機構はエージェントの役割によって、システムの論理構成を隠蔽したり、逆にシステムの情報容易に提供できる両方の機能が求められる。

例えば、図1のエージェントUにとっては、システム内のエージェントの数や機能を意識しないことが望ましい。逆に、エージェントDのように、幾つかのエージェントを統合する役割を持つエージェントは、他のエージェントの情報を知る必要がある。すなわち、以下の2つの要求がある。

### (1) エージェント情報を隠蔽する通信機構

エージェント情報を意識しないエージェントにとっては、他エージェント情報の隠蔽が求められる。具体的には、「通信機構がエージェントの依頼に応じて適切なエージェントを自動的に選択し処理依頼を転送すること」が求められる。

### (2) エージェント情報を提供する通信機構

他のエージェント情報を利用したサービス処理を行うエージェントでは、他エージェントの情報を把握する要求がある。具体的には、「通信機構が他のエージェントの情報を容易に提供できることや、他エージェントの起動や停止による論理構成の変化を迅速に通知すること」が求められる。

#### 4. 提案するエージェント間通信機構

前節で述べた2つの要求を満たすために、提案する通信機構では以下の機能を提供する。

##### (1) エージェント情報管理

エージェントの位置や機能を管理する。また、エージェントの処理依頼関係を保持する。

##### (2) エージェント間通信

エージェントを指定した処理依頼形式とエージェントを指定しない処理依頼形式を提供する。エージェントを指定しない場合は、通信機構は依頼された機能を持つエージェントを自動的に選択する。

##### (3) エージェント情報提供

エージェント情報を容易にエージェントへ提供するインタフェースを提供する。また、エージェントが起動/停止したときの変化を処理依頼関係のあるエージェントに通知する機能を提供する。

#### 4.1 エージェント情報管理

エージェントの位置を保持する。また、エージェントを指定しない通信を可能とするため、エージェントの機能を通信機構が保持する。これらの情報はエージェント間通信における処理依頼の転送や、エージェント情報を提供する時に参照される。通信機構が保持するエージェント情報を以下に示す。

##### (1) エージェント名 (agent\_name)

システム内に走行するエージェントを表す。

##### (2) 機能 (func\_name)

エージェントが提供する機能を表す。

##### (3) インタフェース名 (interface\_type)

インタフェースの形式を表す。

##### (4) 位置

エージェントが走行する計算機に関する情報を表す。

##### (5) 依頼関係

エージェントの処理の依頼関係を表す。

通信機構がこれらのエージェント情報を得るために、表1中の管理の分類で示す関数を各エージェントに提供している。各エージェントはこれらの関数を使用し、エージェント情報を更新する。通信機構は各関数から呼ばれ、エージェントのシステムへの追加や削除を行う。

[例1] 図2にエージェントシステム例を示す。図2では、エージェントBとCは機能func\_Xを提供している。それぞれのインタフェースは、interface\_Bとinterface\_Cである。エージェントAは、機能func\_Xに依頼関係を持っている。

#### 4.2 エージェント間通信

エージェントから他のエージェントへの処理を依頼する場合、表1中の通信の分類に示す関数を提供する。ここで、処理依頼を送信するときには、

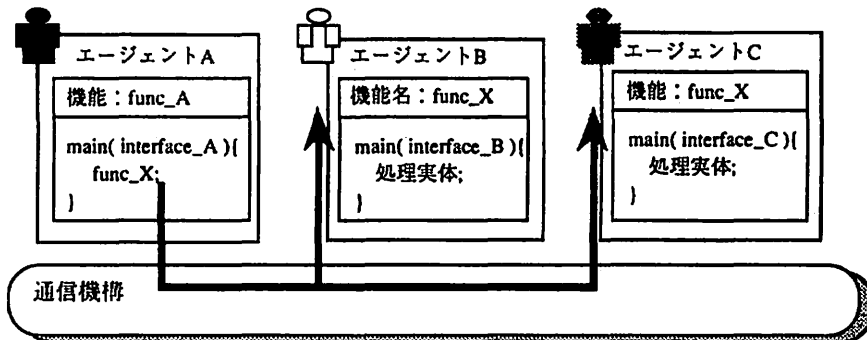


図2 エージェントシステム例

表1 通信機構が提供する関数

分類	関数名	機能
管理	run_agent(agent_name)	エージェント名を登録し走行を宣言する。
	run_function(func_name)	提供する機能を宣言する。
	regist_interface(func_name,interface_type)	インタフェース情報を登録する。
	stop_agent(agent_name)	エージェントが停止する。
	stop_function(func_name)	エージェントの一部の機能を停止する。
通信	send_request(agent_name,func_name,argument)	他のエージェントへ依頼メッセージを送信する。
	send_result(agent_name, func_name, argument)	処理結果を返却する。
取得	get_agent_status(agent_name)	エージェントの走行状態を取得する。
	get_function_status(func_name)	機能の走行状態を取得する。
	get_agent(agent_name)	エージェントが提供する機能を取得する。
	get_function(func_name)	機能を提供できるエージェントを取得する。
	get_interface(agen_name,func_name)	機能のインタフェース情報を取得する。
通知	regist_agent(agent_name)	依頼関係を持つエージェントを登録する。
	regist_function(func_name)	依頼関係を持つ機能を登録する。

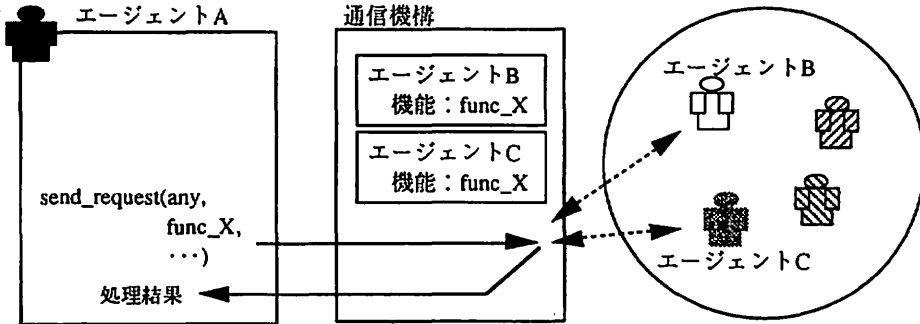


図3 機能指定による処理依頼

依頼相手を指定するために2つの形式がある。

(1) エージェント指定

送信元エージェントが送信先エージェントと機能を指定してメッセージを送信する。通信機構は、エージェントの位置を参照しメッセージを転送する。

(2) 機能指定

処理の依頼先エージェントを指定せずに、機能を指定してメッセージを送信する。この場合、通信機構は、指定された機能を提供するエージェントを選択し、さらに位置を解決してメッセージを転送する。

[例2] 前述のエージェントシステム例において、エージェントAはfunc\_Xを使用するため、エ

ージェントBかエージェントCに処理を依頼する。エージェントBを指定して要求を送信する場合は、send\_request (agent\_B, func\_X, argument) の形式でメッセージを送信する。エージェントを指定しない場合は、send\_request (any, func\_X, argument) の形式でメッセージを送信する (図3参照)。

4.3 エージェント情報提供

システム内のエージェントに関する情報を利用したサービス処理を行うエージェントのために、システム内のエージェント情報を容易かつ迅速に提供する。

(1) 情報取得

エージェントがシステム内の他のエージェント

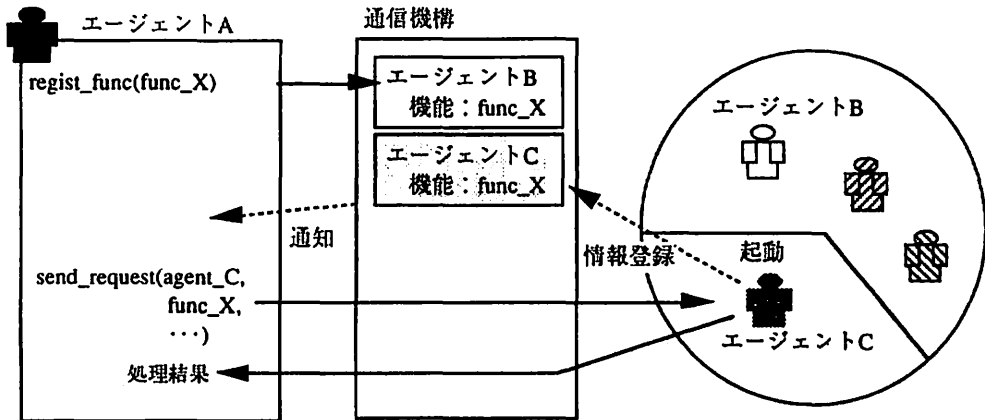


図4 エージェント起動による情報通知

の情報を簡単に取得する機能を提供する。具体的には、表1中の取得の分類に示す関数を提供している。これらの関数を使用することにより、知りたい情報を容易に取得することを可能にしている。

## (2) 情報通知

エージェントが追加/削除されたり、エージェントの機能が変化したときの通知機能を提供する。このことを可能にするため、表1中の通知の分類に示す2つの関数を提供する。各エージェントが、処理の依頼関係を持つエージェント名や機能名を登録することにより、関連するエージェントを通信機構が判断することを可能としている。この情報を用い、登録されたエージェントや機能が起動/停止/変更されたことを、依頼関係を持つエージェントに自動的に通知する。

[例3] 前述のエージェントシステム例において、エージェントCが新たに起動された場合の処理の流れを図4に示す。エージェントCが起動し通信機構に組み込まれると、エージェントCの起動がエージェントAに通知される。エージェントAは、通知を受け取った後、エージェントCを指定した要求を送信できる。

## 5. 情報検索システムにおける実現例

4章で提案したエージェント間通信機構を実現

し、これを利用した情報検索プロトタイプシステムを構築している。本システムはネットワークに多数のデータベースが接続された環境において、利用者が情報を検索することを支援する。本システムの構成を図5に示す。また、通信機構が管理するエージェント情報を表2に示す。プロトタイプシステムでは、インタフェースエージェント (IFA)、検索代行エージェント (RPA)、データベースエージェント (DBA) が協調し、情報を検索する。それぞれのエージェントの役割は以下の通りである。

[IFA] 利用者のあいまいな要求を解釈し、検索に必要な情報形式に変換する。

[RPA] RPAに利用者の検索要求を代行する。具体的には、システム内に多数存在するデータベースから適切なデータベースを選択し、該当するデータベースのDBAに検索実行を依頼する。

[DBA] RPAからの依頼に基づいて、データベース検索を実行し結果を返却する。

それぞれのエージェントと通信機構の関連を以下に説明する。

### (1) IFA

IFAは利用者毎に存在する。各IFAはシステム内の個々のエージェントを意識しない設計としている。そのため、他のエージェントへ処理依頼を送信する形式として、

`send_request (any,DBselect, argument)`

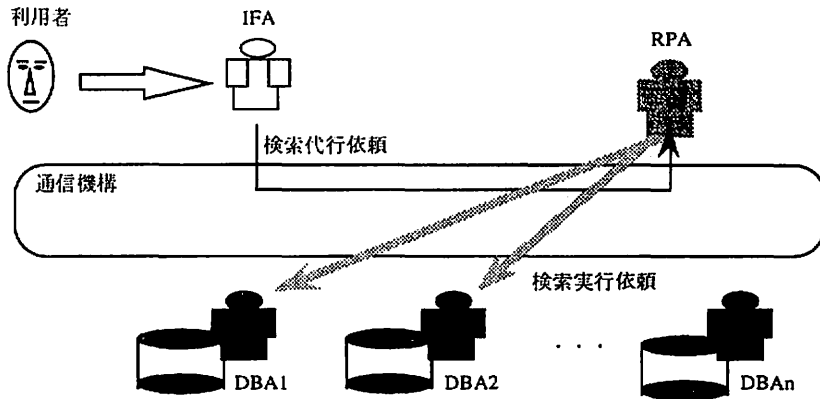


図5 情報検索プロトタイプシステム

表2 プロトタイプシステムにおけるエージェント情報

エージェント名	機能名	位置	インタフェース情報	依頼関係
IFA	インタフェース(interface)	計算機A	IFtype	—
RPA	検索代行(DBselect)	計算機B	DBtype	DBretrieve
DBA1	検索実行(DBretrieve)	計算機C	DBtype	—
...	...	...	...	...
DBAn	検索実行(DBretrieve)	計算機n	DBtype	—

を用いている。これにより、RPAを追加したり機能拡張することを容易にしている。

## (2) RPA

RPAは、データベースエージェントを統合する役割を持っている。そのため、データベースエージェントの情報を把握する必要がある。そのため、依頼関係としてデータベースエージェントが起動/停止したときの通知を得る設計としている。例えば、新たにデータベースエージェントDBAnが起動したときには以下の処理手順で情報通知がなされる。

### [ステップ1]

DBAnは起動した後、run\_agent (DBAn)、run\_function (retrieve) により走行を宣言する。

### [ステップ2]

通信機構は、DBAnを表2に示すエージェント情報を加える。次に、DBAnの機能 (retrieve) を依頼関係に登録しているRPAに新しいデータベースが起動したことを通知する。

### [ステップ3]

RPAは通知を受け取り、DBAnを検索代行サービスに組み込む。

## 6. おわりに

本稿では、柔軟なシステム構成を目指したエージェント間通信機構について述べた。

将来の分散システムの特徴として、システムの論理構成が頻繁に変化することが挙げられる。それに対応した柔軟なシステムを構築するためにはエージェントを用いたシステム構築が有望である。このとき、エージェント間の通信機構への要求条件として、(1)システム構成の隠蔽と(2)システム情報の迅速な提供、の2つの要求がある。

提案するエージェント間通信機構では、システム構成を隠蔽するため、エージェントが提供する機能を管理し、機能を指定した処理依頼を可能にしている。また、システム情報の提供のため、エージェント間の処理依頼関係を保持し、エー

ジェントの起動や追加により影響を受けるエージェントへ自動的に通知する機能を実現している。さらに、複数のデータベースを対象とする情報検索システムを例にした実現方式について述べた。

今後、情報検索プロトタイプシステムの構築を通じて、本エージェント間通信機構の有効性を検証する予定である。

#### 参考文献

- [1] Riecken, D. (ed.), "Intelligent agent", CACM Special Issue, Vol.37, No.7, 1994.
- [2] 所真理雄:"マルチエージェントシステム研究の目指すもの", コンピュータソフトウェア, Vol.12, No.1, pp.78-84, 1995.
- [3] 藤田, 菅原, 白鳥:"エージェントモデルとその記述言語に関する一考察", 信学技報, AI-94-48, 1994.
- [4] 桑原, 石田, 大里:"AgenTalk:マルチエージェントシステムにおける協調プロトコル記述", 信学技報, AI-94-56, 1994.
- [5] Cohen,P.R., Cheyer,A., Wang,M., Baeg, S.C. : "An open agent architecture", Proc. AAAI'94 Spring Symp., 1994.
- [6] 清水謙多郎:"分散OSの研究動向", 情報処理, Vol.36, No.8, pp.699-701, 1995.
- [7] 大野邦夫:"異種分散環境とオブジェクト指向 - CORBA の概要と今後の展望 -", Proc. Advanced Database System Symp., pp.121-135, 1992.