

## CORBA ベースネットワークにおけるフレキシブルグルーピング

奥田 隆史† 足立 整治† 井手口 哲夫†  
安川 博† 田 学軍† 岡崎 直宣††

企業が求めている情報インフラが満たす条件として、(1) 機種・OS・通信プロトコルに依存しない、(2) スケーラブルで拡張変更がダイナミックに出来る、(3) 負荷バランス調整や最短ルート選定機能等のパフォーマンス対策ができる、(4) 運用保守が容易である、(5) 旧来のシステムともシームレスにインテグレートできる、等がある。分散オブジェクト技術は、このような条件を比較的容易に満たすことができる技術として期待されており、その仕様として OMG(Object Management Group) が定めた CORBA(Common Object Request Broker Architecture) がある。本論文では CORBA をベースとするネットワークシステムにおいて、組織内外におけるダイナミカルかつフレキシブルなグルーピングを可能とするシステムを提案する。

### A Dynamic Flexible Grouping over CORBA-based Network within and across Organizations

TAKASHI OKUDA,<sup>†</sup> SEIJI ADACHI,<sup>†</sup> TETSUO IDEGUCHI,<sup>†</sup>  
HIROSHI YASUKAWA,<sup>†</sup> XUEJUN TIAN<sup>†</sup>  
and NAONOBU OKAZAKI<sup>††</sup>

Use of the Internet and IT (Information Technology) has spread in business communications. Many kinds of group (collaborative) work and communication styles can be supported by the use of them. For example, a typical management application in today's enterprise network involves remote cooperation of a number of skilled people within and across organizations. This management is named "dynamic flexible grouping". For the achievement of this flexible grouping, we adopt the Common Object Request Broker Architecture (CORBA) framework, since software design becomes easier with this framework. As an open standard for distributed object computing, CORBA provides a strong, universal, powerful framework for developing applications within distributed heterogeneous environments. In this paper, we have examined some uses of CORBA for flexible grouping within and across organizations, and a prototype has been developed.

#### 1. はじめに

多様なコミュニケーションスタイルの普及により新しいワークスタイルが定着してきた。そのため、あらゆるコミュニケーションメディアをサポートできるような、仕事の種類や目的に応じたフレキシブルで

かつダイナミックなグルーピングの重要性が増してきた。一方で、企業が求めている基幹部分の情報インフラを再構築するための技術として、分散オブジェクト技術があり、その仕様が OMG(Object Management Group) により CORBA(Common Object Request Broker Architecture) として定められている<sup>1)~3)</sup>。

これらを背景として、本論文では、CORBA をベースとするネットワークシステム上で、組織内外における動的なフレキシブルなグルーピングを目的とする基本プロトコル(結合, 分離)を提案する。以下、2 ではコミュニケーションの多様性について、3 ではコミュニケーションの多様性により生じた新しいワークスタイルとダイナミック・フレキシブル・グルーピングの重要性について、それぞれ述べる。4 では CORBA の特色を述べるとともに、本研究で用いるシステム

† 愛知県立大学情報科学部  
Faculty of Information Science and Technology, Aichi Prefectural University

†† 三菱電機(株)・情報技術総合研究所  
Information Technology R&D Center, Mitsubishi Electric Corporation

\* 奥田隆史: 〒680-1198 愛知県長久手町熊張  
愛知県立大学情報科学部地域情報科学科  
TEL: (0561)64-1111 FAX: (0561)64-1108  
email: okuda@ist.aichi-pu.ac.jp

構成について記述する。5ではダイナミック・フレシキブル・グルーピングの基本となる結合および分離プロトコルを提案する。最後に6でまとめる。

## 2. コミュニケーションの多様化

公衆網にアクセスするための加入者網が固定電話を中心に構成されていた時代には、組織内および組織外でのビジネスコミュニケーションの形態は、次のように分類されていた<sup>4)~6)</sup>。

- (a) 同時間同場所 (*Same Time and Same Place*): 一般的な会議やミーティング。
- (b) 同時間異場所 (*Same Time and Different Place*): 音声やビデオ電話による会議 (参加者の同期をとる必要がある)。
- (c) 異時間同場所 (*Different Time and Same Place*): 交代勤務者に渡すメッセージによる情報伝達。
- (d) 異時間異場所 (*Different Time and Different Place*): 電子メールあるいはFAXによる情報交換。

ところが、移動体通信等の無線利用やデータ通信の高速化により、“Unpredictable” という概念が時間および場所の概念に加わり、次のようなコミュニケーションスタイルが可能になった。

- (e) 同時間不規則場所 (*Same Time and Different Unpredictable Place*): 携帯電話等で連絡をとるケース。
- (f) 異時間不規則場所 (*Different Time and Different Unpredictable Place*): モバイル環境による電子メールやFAXへのアクセス。
- (g) 不規則時間同場所 (*Different Unpredictable Time and Same Place*): 予約なしの臨時に開催されるミーティング。
- (h) 不規則時間異場所 (*Different Unpredictable Time and Different Place*): 電子メールによる情報伝達。
- (i) 不規則時間不規則場所 (*Different Unpredictable Time and Different Unpredictable Place*): モバイル環境同士による電子メールへのアクセス。

## 3. ダイナミック・フレシキブル・グルーピング

上記(a)~(i)のようなコミュニケーションが可能となったことにより、在宅勤務、テレワーク、テレコミュニケーション、SOHO(Small Office Home Office)等のような多様なワークスタイルが実践されるように

なってきた<sup>15)</sup>。これらのワークスタイルは、企業にとっては生産性の向上等のメリットがあり、従業員にとっては通勤する手間が省けるだけでなく、仕事に集中できるメリットがある<sup>7)</sup>。

さらに、文献<sup>8),9)</sup>で紹介されているデルコンピュータ社のビジネスモデルでは、顧客を満足させるためには、単なる情報の共有を目的とする会社と顧客とのグルーピングでなく、情報を使って顧客やサプライヤーとの関係を再構築するようなグルーピングが必要であることも述べられている。すなわち、個人は特定のグループだけに属するだけでなく、仕事の内容によっては、様々なグループに同時に加わりグループワークをこなすことが重要になってきた<sup>4),5)</sup>。

上記のようなワークスタイルの現状をふまえて、本研究でサポートを可能とするダイナミック・フレシキブル・グルーピング (DFG:Dynamic Flexible Grouping) の概念を図1を用いて説明する。

図1において、組織XはスタッフA,B,C、組織YはスタッフD,E,Fから構成されているものと仮定する。この場合、スタッフAがメンバーとして所属するグループには、

- (I) 同一組織内の小グループ (*(Small) Group within Same Organization*): .
- (II) 同一組織=グループ (*Group equals Same Organization*): .
- (III) 組織内グループに組織外スタッフが加わったグループ (*Group across Organizations*): .
- (IV) 統合化組織=グループ (*Group equals Integrated Organizations*): .
- (V) 組織外スタッフと構成する組織横断型グループ (*(Small) Group across Organizations*): .
- (VI) 複数のグループ (*Some Groups*): .

に分類することができる。このようにスタッフAは、様々なグループの構成員となりながら仕事をこなす必要がある。また、Aが目的を達成するためには、A自信が必要な人をグループに取り込むという考え方も生じる。従って、管理者は、スタッフAのスケジューリングあるいはグループ属性をマネジメントするためにはソフトウェアによるサポートが必要不可欠となる。

## 4. CORBA ベース DFG システムの構成

2節や3節で述べたように、多様なコミュニケーションのサポートによるより新しいワークスタイルが定着してきた。そのため、仕事の種類や目的に応じたフレシキブルでかつダイナミックなグルーピングの重要性が増してきた。

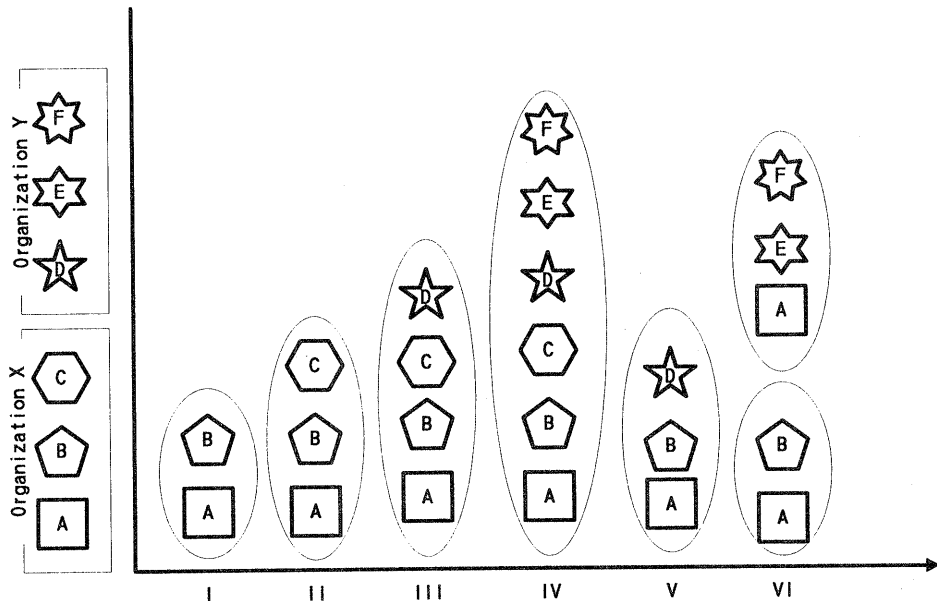


図 1 組織内外におけるグルーピング  
Fig. 1 Grouping within and across Organizations

一方で、企業が求めている情報インフラが満たす条件として、(1)機種・OS・通信プロトコルに依存しない、(2)スケーラブルで拡張変更がダイナミックに出来る、(3)負荷バランス調整や最短ルート選定機能等のパフォーマンス対策ができる、(4)運用保守が容易である、(5)旧来のシステムともシームレスにインテグレートできる、等がある<sup>1)</sup>。分散オブジェクト技術は、このような条件を比較的容易に満たすことができる技術として期待されており、その仕様として OMG(Object Management Group) \*が定めた CORBA(Common Object Request Broker Architecture) がある<sup>2),3),10)</sup>。

このような理由から、本論文では CORBA をベースとするネットワークシステムにおいて、組織内外におけるフレキシブルなグルーピングを可能とするシステムを提案する。

#### 4.1 CORBA の主要要素

CORBA を構成する主要な要素技術は

- ORB(Object Request Broker)
- IDL(Interface Definition Language)
- IDL コンパイラ (言語マッピング)

\* OMG の活動目的は「異機種分散環境において、再利用可能かつ相互運用性のあるアプリケーションの開発を容易にする、ベンダー中立的な標準仕様の策定」である。

である。ORB はクライアント・プログラムからサーバーオブジェクトへのメッセージを仲介するソフトウェアで、これを使用することにより、クライアントはサーバー・オブジェクトのメソッドを透過的に呼び出すことが出来る。すなわち、クライアントはオブジェクトの物理的な位置や実装言語、サーバーとの OS の違いを意識する必要がなくなる。

IDL はクライアント・サーバーオブジェクト間のインターフェイスを記述するための宣言や定義を記述するための言語で、その仕様は CORBA で規定されている。

IDL コンパイラは、IDL に従って記述されたインターフェイス定義ファイル (IDL ファイル) をプログラミング言語にマッピング (言語マッピング) する。IDL コンパイラが生成するソースコードは、クライアント用のスタブ・コードとサーバ用のスケルトン・コードがある。クライアントのプログラムは、IDL により生成されたクライアント側のスタブとサーバ側のスケルトンを経由することにより、ネットワークの存在を意識することなく、ネットワーク透過な呼び出しを行うことが出来る。

#### 4.2 システム構成

本研究のシステム構成を図 2 に示す。システム開発

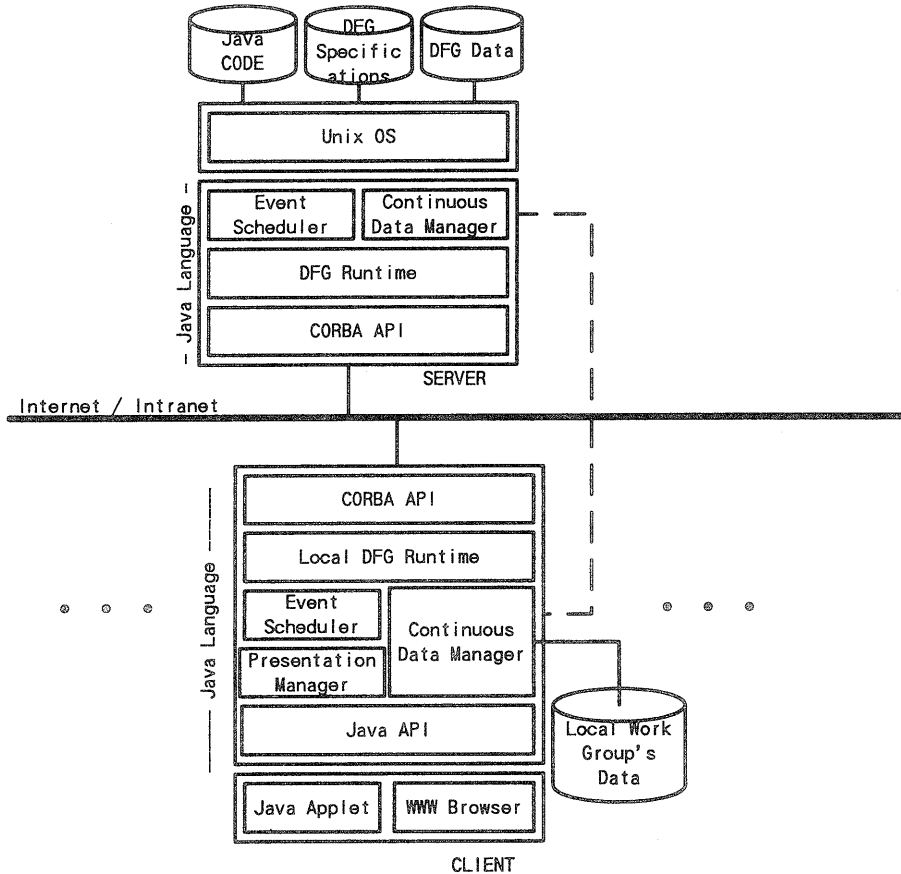


図 2 システムの構成

Fig. 2 Architecture for Dynamical Flexible Grouping by Using CORBA/Java

のプログラミング言語には、CORBA と組み合わせることによりシステムの中立性が増し、なおかつシステムのインテグレーションやアップグレードが容易になるため、Java を採用することにし、開発環境はインプライズコーポレーション社<sup>11)</sup>の VisiBroker for Java を採用することにする。

サーバー側の各モジュールの機能は以下のとおりである。

- *Event Scheduler*: 予めグルーピングをスケジュールするような場合に用いる。例えば、定期的なミーティングの開催等がスケジュールされる。

- *DFG Run Time*: クライアント・サーバモデルの核となる部分で、(1)CORBA の実行型とコミュニケーションシ (情報収集とそれに伴う関数起動)、(2) システムに依存する部分のインターフェイスの役割を果たす。
- *Continuous Data Manager*: クライアント側に存在している複数のグループについて、各グループのメンバーやその属性などを管理するデータベース。
- *CORBA API*: CORBA Application Programming Interface。例えば、パッケージ<sup>12)</sup>は

CORBA API と Java との言語マッピングを行う。クライアント側の各モジュールの機能は以下のとおりである。

- *Local DFG Runtime* : クライアント側の核となる部分で、(1)CORBA の実行型式とコミュニケーションし (情報収集とそれに伴う関数起動)、(2)システムに依存する部分のインターフェイスの役割を果たす。
- *Local Event Scheduler*: クライアント側でのイベントの管理。
- *Presentation Manager* : Java API で利用されるデータのフォーマット管理を行う。アプリケーション側でデータを利用 (プレゼンテーション) するための一種のデータ変換モジュールと見なすことができる。
- *Continuous Data Manager*: 同一グループにおいて共有する情報等を提供する。各グループのメンバー (クライアント) はグループにいる間は管理しなくてはならない。
- *CORBA API* : サーバー側と同様。
- *Java API and Applet, WWW Browser* : 提案システムはグループユーザーの共通の作業インターフェイスとしてブラウザーを利用することにする。

## 5. 基本 DDG プロトコル

図 1 で説明した DFG を行うために、プロトコルに必要とされる基本となるオペレーションである結合 (Conjunction) プロトコルおよび分離 (Disjunction) プロトコルを試作した。なお、サーバには、グループ ID と、そのグループを構成するエンティティのリストおよび管理エンティティの組が (全て) 登録されている。単独のエンティティからなるグループも含まれる。各エンティティは、複数のグループに所属することができるものとする。

以下、グループ  $g_i$  の管理エンティティを  $em_i$ 、グループ管理サーバー  $S = (g, em, M)$  で表現する。ただし、 $g$  はグループ id,  $em \in M$  はグループ  $g_i$  の管理エンティティ,  $M \in E^+$  はグループ  $g$  を構成するメンバーを意味する。

### 5.1 結合プロトコル

結合プロトコルの基本概念を図 3 を用いて説明する。ただし、エンティティ  $em_0$  がエンティティ  $em_1$  を結合したい場合である。

**Step 1** 結合アクションの起動側グループのマネージャエンティティ (イニシエートエンティティ)  $em_0$  が、リクエスト  $gIdReq(gId, mList)$  をサー

バー  $S$  に送信する。ただし、 $gId$  は  $em_0$  のグループ ID,  $mList$  は新たに含みたいエンティティのリストである。

**Step 2**  $S$  は、 $mList$  を参照し、 $gList$  を作成し  $gIdResp(gList)$  を  $em_0$  へ送信する。ここで、 $gList = \{g | g \subseteq mList, g \neq \emptyset\}$  を満足する。

**Step 3**  $em_0$  は  $gList$  を参照し、新たに含みたいエンティティを全てカバーするグループ  $gMinSet = \{g_1, \dots, g_k\} \in gSet$  を選択する。ただし、 $gMinSet$  は  $gSet = \{\{g_1, \dots, g_k\} | \{g_1, \dots, g_k\} \subseteq gMinSet, g_1 \cup \dots \cup g_k = mList\}$  を最小にするように選択される。

**Step 4** **Step 3** で得られた  $gMinSet$  に含まれる各グループのマネージャエンティティ (図では  $em_1$ ) へ結合を要求するため、 $em_1 \rightarrow conjInd(gId)$  を送信する。

**Step 5** 結合を要求された  $em_1$  (レスポндаエンティティ) は非発火状態 (inactivation) へ遷移した後、 $S$  へ、 $em_1$  が管理していた  $gId$  の消滅をリクエストするため、 $delIdReq(gId)$  を送信する。

**Step 6**  $S$  は  $gId$  を消去し、 $em_1 \rightarrow delIdResp(gId)$  を送信する。

**Step 7** 消滅処理を確認したレスポндаエンティティ  $em_1$  は確認応答  $conjConf(gId)$  を、イニシエートエンティティ  $em_0$  へ送信する。

**Step 8**  $em_0$  は非発火状態の  $em_1$  へ消滅を要求する  $termInd(gId)$  を送信する。

**Step 9**  $em_1$  は消滅要求を受けてグループを消滅するとともに、確認応答  $termConf(gId)$  を  $em_0$  へ送信する。

**Step 10**  $em_0$  は、**Step 4** ~ **Step 9** の処理を  $gMinSet$  に含まれる全てのグループに対して行う。

**Step 11**  $em_0$  は  $gMinSet$  に含まれる全ての結合相手の消滅確認応答  $termConf(gId)$  を確認した後、非発火状態へ遷移する。次に、新しく含めるメンバーとなるエンティティを含めた登録要求  $regIdReq(gId, mList)$  をサーバー  $S$  に送信する。ただし、 $gId$  は新しく生成するグループ ID,  $mList$  は  $gId$  構成するエンティティのリストである。

**Step 12**  $S$  は登録作業 (データベース更新) を行うとともに、 $em_0 \rightarrow regIdResp(gId)$  を送信する。

**Step 12**  $em_0$  は、新たなメンバとなるエンティティに対してセッション鍵の配布を行った後、発火状態に遷移し、新たなグループ作業を開始する。

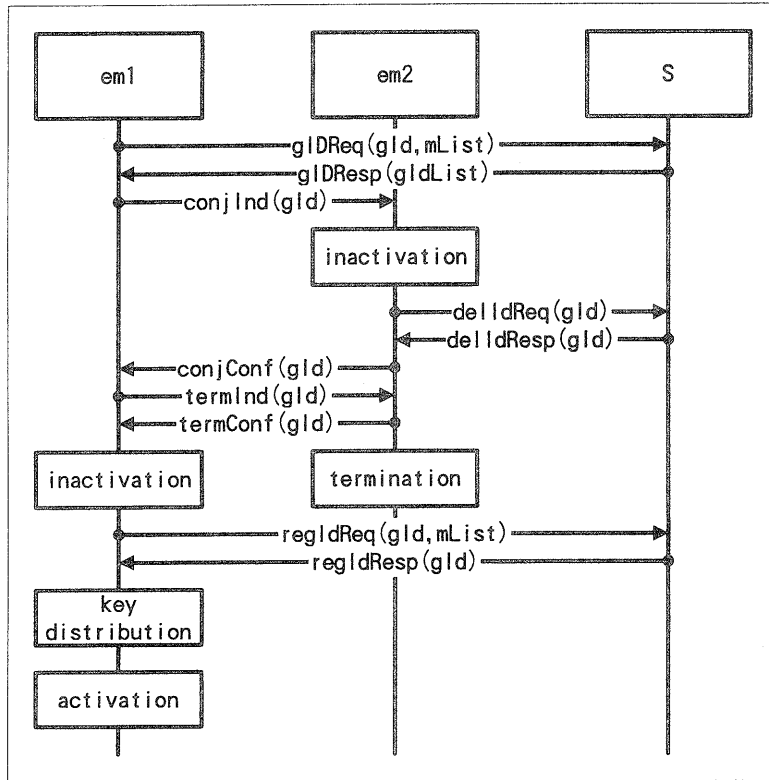


図3 結合プロトコル  
Fig. 3 Sequence of a Conjunction Procedure

## 5.2 分離プロトコル

分離プロトコルの基本概念を図4を用いて説明する。

- Step 1** 分離アクションを起動するグループのマネージャエンティティ (母エンティティ)  $em_1$  が非発火状態に遷移し、分離するメンバからなる新たなグループのマネージャエンティティ (娘エンティティ)  $em_2$  を起動する。
- Step 2** 母エンティティ  $em_1$  は残されたメンバーからなるグループを、既存のグループIDで登録するために、 $regIdReq(gId, mList)$  をサーバー  $S$  に送信する。
- Step 3** サーバー  $S$  は  $gId$  を登録するとともに、 $regIdResp(gId)$  を母エンティティ  $em_1$  へ返信する。
- Step 4** 娘エンティティ  $em_2$  は、分離したメンバからなる新たなグループを登録するために、 $regIdReq(gId, mList)$  をサーバー  $S$  へ送信する。ただし、 $gId$  は  $em_2$  のグループID、 $mList$  は分離したエンティティのリストである。
- Step 5** サーバー  $S$  は  $gId$  を登録するとともに、

$regIdResp(gId)$  を娘エンティティ  $em_2$  へ返信する。

- Step 6** 母エンティティ  $em_1$  は、グループ内の全てのメンバに対して新たなセッション鍵を配布した後、発火状態に遷移する。
- Step 7** 娘エンティティ  $em_2$  は、グループ内の全てのメンバに対して新たなセッション鍵を配布した後、発火状態に遷移する。

## 6. おわりに

多様なコミュニケーションスタイルの普及により新しいワークスタイルが定着してきた。そのため、あらゆるコミュニケーションメディアをサポートできるような、仕事の種類や目的に応じたフレキシブルでかつダイナミックなグルーピングの重要性が増してきた。一方で、企業が求めている基幹部分の情報インフラを再構築するための技術として、分散オブジェクト技術があり、その仕様がOMGによりCORBAとして定められている。

これらを背景として、本論文では、CORBAをベ-

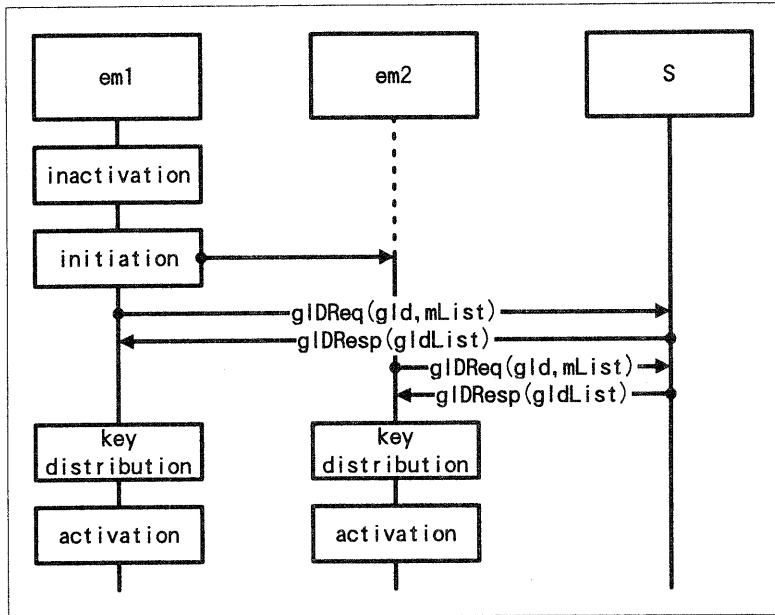


図4 分離プロトコル

Fig. 4 Sequence of a Disjunction Procedure

スとするネットワークシステム上で、組織内外における動的なフレキシブルなグルーピングを目的とする基本プロトコル（結合，分離）を提案した。

今後は、提案した基本プロトコルの評価、運用時の最適負荷分散について研究を進める。さらに、動的にグルーピング情報を的確に管理するために、データベースを分散管理することでアクセス負荷を動的に平準化する方法<sup>13)</sup>や季節的要因や経済的要因などによるグループ特性の変動を予測する方法<sup>14)</sup>等を、様々なグルーピングのケースに適用することを考えている。

#### 参 考 文 献

- 1) <http://www.tcn-j.com>.
- 2) <http://www.corba.org>.
- 3) <http://www.omg.org>.
- 4) Efraim Turban, Ephraim McLean and James Wetherbe, *Information Technology for Management: Improving Quality and Productivity*, John Wiley & Sons Inc. (1996).
- 5) Joel Kugelmass, *Telecommuting: A Manager's Guide to Flexible Work Arrangements* Lexington Books (1995).
- 6) Jack Woodall, Deborah K. Rebeck and Frank Voehl, *Total Quality in Information Systems and Technology*, Lexington Books (1995).
- 7) 米国商務省著，室田泰弘訳，「デジタル・エコ

ノミー (*The Emerging Digital Economy*)」，東洋経済新聞社 (1999)。

- 8) マイケル・デル，キャサリン・フレッドマン著，國領二郎監訳，吉川明希訳，「デルの革命：ダイレクト戦略で産業を変える」，日本経済新聞社 (1999)。
- 9) Lynn Margherio, Dave Henry, Sandra Cooke, Sabrina Montes, Jack Woodall, Deborah K. Rebeck and Frank Voehl, *The Emerging Digital Economy*, U.S. Department of Commerce (1997)。
- 10) 松野良蔵，「Java+CORBA 分散オブジェクトシステムの構築：インターネット・プログラミングの仕組みと実践」，翔泳社 (1999)。
- 11) <http://www.inprise.co.jp>
- 12) <http://www.javasoft.com/products>
- 13) 平野正則，川瀬克之，今川仁，木ノ内康夫，“分散トランザクション処理における動的な負荷の平準化方法”，電子情報通信学会論文誌，Vol. J82-B, No. 6, pp. 1138-1150, (1999)。
- 14) 正井一夫，根岸和義，土田正士，松澤茂，“更新処理を並列実行するUNIX向けDBMSを開発：負荷分散機能で並列処理の効率を改善”，日経エレクトロニクス，No. 630, pp. 101-114, Feb., (1995)。
- 15) <http://www.mpt.go.jp/policyreports/japanese/papers>