

分散環境におけるサービス提供のメカニズム

2D-5

樋地正浩¹ 高橋勉¹ 布川博士² 宮崎正俊³

¹ 日立東北ソフトウェア ² 宮城教育大学理科教育研究施設 ³ 東北大学大学院情報科学研究科

1. はじめに

コンピュータネットワークの普及や利用の拡大にともない、コンピュータネットワークには、利用者が利用したいサービスを提供するサービス提供システムとなることが求められてきている。ネットワーク上に構築されるこのようなサービス提供システムをここでは分散環境と呼ぶ^[1]。すなわち分散環境とは、利用者がさまざまなサービスの中から必要なサービスを選択し、必要に応じてそれらのサービスを連携させ、利用できるシステムである。これらのサービスは、コンピュータネットワークに接続されたコンピュータ上のアプリケーションプログラム（AP）により提供される。分散環境を構築するためには、コンピュータネットワークの物理的構成を意識することなく必要なサービスを利用できること、複数のサービスを連携させる枠組を与えること、複数のサービス間の連携の仕方を利用者が与えられること、が必要である。この枠組を実現するためにサービスエンティティとCo-Clientの2つを用いる。サービスエンティティ（Service Entity）は、APを一つのサービスを提供する実態としてとらえたものである。Co-Clientは、サービスエンティティの間の連携の仕方やサービスの利用方法を持ち、各サービスエンティティの間を移動し、各サービスを利用しながら、最終的に利用者の元に戻り、利用者の求めているサービスを与えるものである。

本稿では、分散環境を構築するためのサービスエンティティとCo-Clientの構造と機能、及びその実現メカニズムについて述べる。

2. 分散環境の基本構成

2.1. サービスエンティティの構造と機能

サービスエンティティは、利用者に対して何等かのサービスを提供する単位であり、分散環境を構成する構成要素である。各サービスエンティティは、基本的に全て対等な関係であり、利用者の要求に応じてサービスエンティティの間の関係は変化する。すなわち、あるサービスエンティティは、他のサー

ビスエンティティにサービスを要求することもあれば、他のサービスエンティティからの要求を受け取り、サービスを提供することもある。サービスエンティティの粒度は、特に規定していない。また、ユーザインタフェースやCo-Clientも一つのサービスエンティティとして考えることができる。

サービスエンティティは、実際のサービスを実行するAP本体（AP Core）とAPをサービスエンティティとして分散環境に組み込むためのインタフェース（IF）から構成される（図1）。インタフェースは、サービスエンティティが受け取ったメッセージをAPに応じた形式に変換する機能、APの処理結果をメッセージとしてCo-Clientに送る機能、APの実行を制御する機能を持つ。実行制御機能は、単にAPの持ついくつかのコマンドを実行するだけではなく、APの実行過程で他のサービスが必要になる場合の連携の仕方を持つ。これにより、複数のサービスを組み合わせることができ、あるサービスを提供するために必要な他のサービスを利用者に意識させることなく提供することができる。

2.2. Co-Clientの構造と機能

Co-Clientは、利用者が分散環境上のサービスを利用するために用いるものであり、サービスの利用方法や複数のサービスエンティティの間の連携の仕方を持つ。利用者は必要に応じたCo-Clientを分散環境に与えることにより、必要としているサービスを分散環境から得ることができる。すなわち、Co-Clientにより、基本的に対等であるサービスエンティティの間に、必要なサービスを得るための関係付けを与えることになる。この関係に基づきCo-Clientは、各サービスエンティティの間を移動し、利用者の求めているサービスを与えることになる。

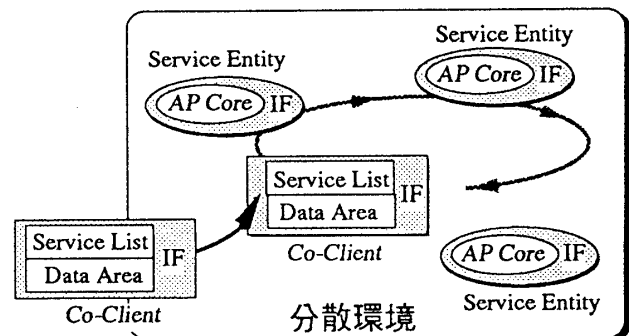


図1 サービス提供・利用の基本メカニズム

A Mechanism for Providing Services on Distributed Environment

¹Masahiro HIJI, ¹Tsutomu TAKAHASHI,

²Hiroshi NUNOKAWA, ³Masatoshi MIYAZAKI

¹ Hitachi Tohoku Software, Ltd.

² Institute for Science Education, Miyagi Univ. of Education

³ Graduate School of Information Sciences, Tohoku Univ.

Co-Clientは、必要とするサービスを得るための手順を記述したサービスリスト (Service List) とデータ格納領域 (Data Area), サービスエンティティとのインタフェース (IF) から構成される。サービスリストは、必要とするサービスを得るための手順をサービスエンティティの利用順序と個々のサービスエンティティで利用する機能として記述する。データ格納領域は、個々のサービスエンティティに渡すデータやサービスエンティティから得られるデータを格納する。インタフェースは、個々のサービスエンティティとの間でやり取りするメッセージの生成、処理をする。

Co-Clientは、サービスリストから次に利用すべきサービスエンティティを決定し、そのサービスエンティティに移動する。

3. 分散環境の実装

分散環境を実際のコンピュータネットワーク上に実装するための実現方式を図2に示す。この方式に基づき、Lispをベースに通信機能とGUI機能を付加したDeLis^[1]と呼ぶ記述言語系を用いて分散環境の構築を行っている。

分散環境は、各コンピュータごとに1つずつ起動されるデーモンプロセスと複数のサービスエンティティプロセスから構成される。デーモンプロセスは、それが動作するコンピュータ上のサービスエンティティプロセスより先に起動される。サービスエンティティプロセスは、それが動作するコンピュータ上のデーモンプロセスに対して接続される。このように、サービスエンティティの間にデーモンプロセスを介した通信路を確立することにより分散環境を構築する。また、それぞれのプロセスは1つのDeLisインタプリタを持つ。

デーモンプロセスは、DeLisの通信機能を用いて動的に通信路を生成することができる。また、他のプロセスとの間の通信路の管理、サービスエンティティプロセスの管理、Co-Clientの配送管理を行う。デーモンプロセスは、起動時に他のコンピュータ上で動作しているデーモンプロセスとの間に通信路を確立した後、他のプロセスからの接続要求と通信路を經由して送られてくるCo-Clientを待つ。

サービスエンティティプロセスは、そのコンピュータ上のデーモンプロセスとの間に通信路を接続し、デーモンプロセスに対してそのサービス名を登録する。このサービス名が分散環境上で提供されるサービスを識別するための名称となる。サービスエンティティプロセスは、デーモンプロセスにより通信路から送られてきたCo-Clientを受け取り、Co-Clientからのメッセージに応じた処理を行い、その結

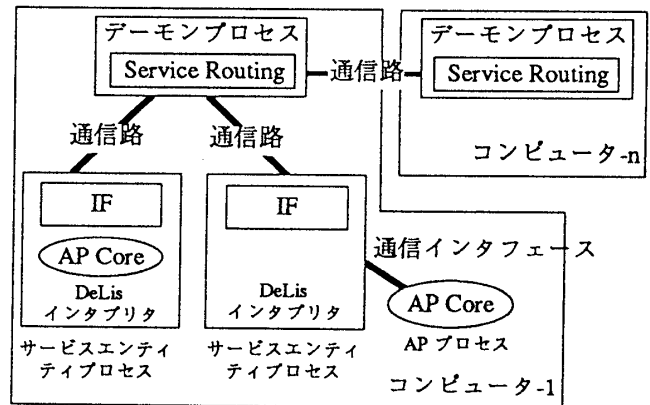


図2 分散環境の実現方式

果をCo-Clientに送る。AP本体がDeLis以外の言語で記述されている場合には、サービスエンティティはAP本体を1つのプロセスとして立ち上げ、その間に標準入出力による通信インタフェースを確立し、その通信インタフェースを介してAPの実行を制御する。

Co-Clientは、DeLisのプログラムとして記述される。DeLisは、プログラムをデータとして取り扱うことができるため、プログラムとして記述されたCo-ClientをデータとしてDeLis間でやり取りし、受け取ったCo-Clientをプログラムとして実行することにより、Co-Clientの機能を実現する。

デーモンプロセスはCo-Clientを受け取ると、そのサービス名が登録されているか否かを判定し、登録されていればそのサービスエンティティプロセスに対してCo-Clientを送る。登録されていなければ、他のデーモンプロセスに対してCo-Clientを送る。すなわち、サービス名に基づきCo-Clientのルーティングを行う (Service Routing)。これによりCo-Clientは、必要とするサービスを探しながら分散環境を移動し、必要なサービスの間の連携を取りながら利用者の目的とするサービスを提供する。

4. おわりに

このように分散環境では、コンピュータネットワーク上の様々なアプリケーションプログラムにより提供されるサービスを、利用者が目的に応じて柔軟に組み合わせて利用することができる。これにより、コンピュータネットワークは、単なるデータ通信のためのシステムではなく、サービス提供のためのシステムとすることができる。

参考文献

- [1] 三石, 布川, 宮崎, 野口: 分散環境のための言語系 DeLis, 情報処理学会プログラミング-言語・基礎・実践-研究会資料 93-PRG-10-8 pp.57-64 (1993)