

分散マルチメディアエージェント環境

7H-2

畠中 啓 中松 芳樹 長坂 篤

沖電気工業(株) マルチメディア研究所

1 はじめに

コンピュータ処理能力の向上と高速ネットワークの発展にともない、分散環境上でマルチメディア情報のサービスを提供する環境の提案が行なわれている。我々は高機能なマルチメディアアプリケーションを開発する枠組を構築するため、オブジェクト指向の考え方を基本として、オブジェクトの集団からなる環境（これを場と呼ぶ）に存在する自律的なオブジェクト（これをエージェントと呼ぶ）が相互協調するという視点を導入する。本稿では分散環境を前提としたマルチメディアエージェント実行モデルとその全体構成、エージェント機構の実現方式について述べる。

2 実行モデル

2.1 分散マルチメディアエージェント

図1は、分散マルチメディアオブジェクトを含むエージェントが場を介して相互作用する処理モデルを示している。

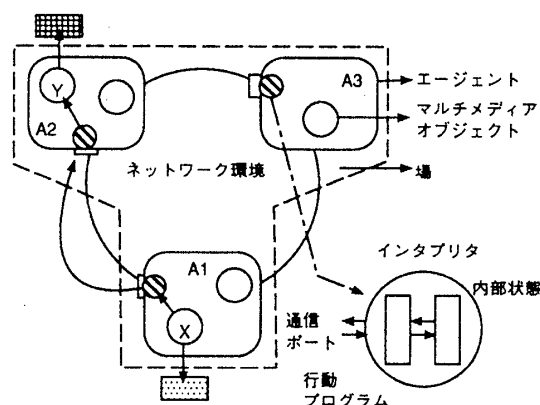


図1 分散マルチメディアエージェントの実行モデル

(1) 分散マルチメディアオブジェクト

本環境が想定するマルチメディアオブジェクトのモデルは、VTR やスピーカなどの物理デバイスとある形式のメディアを含むファイルなどを抽象

化した仮想デバイス間に論理的な接続である仮想コネクションを張り、そのコネクション上にある形式のストリームが流れることを基本とする。

(2) マルチメディアエージェント

マルチメディアエージェントは分散環境上でのマルチメディア情報サービスを提供する主体であり、クライアントであるエージェントとの間で次のような通信インタフェースを持っている。

- マルチメディアオブジェクトのライフサイクル管理
- 仮想デバイスやストリームの制御
- マルチメディアストリームの転送
- 同期情報を始めとする種々のイベントの通知

2.2 エージェントモデル

エージェントは、分散環境上に存在する自律性を持つオブジェクトであり、分散環境上の場に含まれる。それぞれのエージェントは、エージェント間通信機能とプログラム実行のためのインタプリタを持つ。

(1) 場

場は、エージェント間通信を媒介する仮想的な環境を表すオブジェクトであり、集団を構成するエージェントのリストを持っている。

(2) エージェント間通信

エージェント間の通信は、基本的にエージェントあるいはエージェントの集合に依頼するタスクと、その引数を渡すことによって行なわれる。エージェントの並列性と自律性を向上させるために、次のような通信プリミティブを用意している。

- *Post e t*

環境 e のエージェントにタスク t を実行させる。これは呼び出し側のエージェントと並行して実行され、この値を参照する計算は値が返ってくる時点まで待たされる。

- *Send e t*

環境 e のエージェントにタスク t を実行させる一方向の通信である。

- *Event i*

タスク *i* の依頼を受け取り実行する。

3 構成

本環境の全体構成は、図2のとおりである。

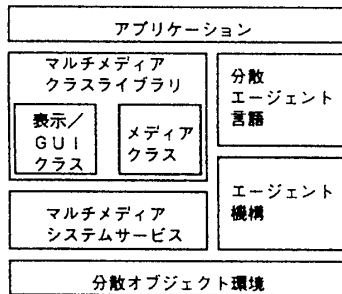


図2 分散マルチメディアエージェント環境の構成

分散オブジェクト環境上にマルチメディアシステムサービス [1] とエージェント環境を構築する。マルチメディアシステムサービスは、個々のメディアやユーザインタフェースを抽象化したマルチメディアオブジェクトを操作する。アプリケーションインタフェースとして、エージェントモデルを反映した分散エージェント言語と、マルチメディアシステムサービス層の上に作成されるマルチメディアクラスライブラリを提供する。以下、いくつかの構成要素について説明する。

● 分散オブジェクト環境

ネットワーク上に存在するオブジェクト間の通信プラットフォームとして、OMG(Object Management Group) で標準化が進められているCORBA(Common Object Request Broker Architecture) [2] に準拠した環境を使用する。

● マルチメディアクラスライブラリ

マルチメディアシステムサービス層では抽象化されていた、動画や音声などメディア固有のクラスと、仮想デバイスの出力を計算機ウィンドウ上に表示したりユーザインタフェースを実現する表示/GUIクラスを実装したクラスライブラリである。

● 分散エージェント言語

分散エージェント言語は、Dylan[3] をベースとする言語であり、タスク依頼のためのプリミティブ *Post*, *Send*, *Event* の追加と実行モデルの変更、イベント定義と割り込み機能の追加などの拡張を行っている。

エージェント間の通信におけるタスクの依頼は総称関数と *Post*, *Send*, *Event* によって実現し、*Post*

の実行はCORBAを介しての通信とエージェントにおけるメソッド実行によって行なわれる。

4 実現方式

図3に分散環境上でエージェント機構を実現する方式を示す。

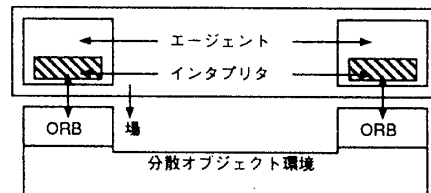


図3 エージェント機構の実現方式

エージェントのインタプリタにCORBAのORB(Object Request Broker)との通信インタフェースを持たせ、ここを介してタスクの依頼を行なう。ORBはタスクと引数を受信エージェントの管理する通信バッファへ転送後、エージェントに対して、シグナルを発行する。一種の割り込みハンドラの機能も持つインタプリタは、シグナル受信後、タスクの内容を解釈して、適切なメソッドを実行する。

各エージェントは、自分が存在する場オブジェクトへの参照を持っており、このオブジェクトを介して、通信を行なう対象エージェントの存在位置を知ることができる。

現在、Sun ワークステーションおよび HyperDesk 社の HD-DOMS を使用したエージェント機構の構築と、言語処理系の開発を並行して進めている。

5 今後の計画

今後は、マルチメディアシステムサービスの実装と、表示/GUIクラスとメディアクラスから構成されるマルチメディアクラスライブラリの実装を行なう予定である。

参考文献

- [1] "Multimedia System Services Version 1.0", HP, IBM, SunSoft, Jun. 1993.
- [2] "Common Object Request Broker: Architecture and Specification", Object Management Group document, Dec. 1991.
- [3] "Dylan: An object-oriented dynamic language", Apple Computer Inc., Apr. 1992.