

## 階層構造型タプル空間に基づく自律協調の情報場

矢島研自<sup>†</sup> 坂下善彦<sup>‡</sup>

湘南工科大学 大学院工学研究科<sup>†</sup> 情報工学科<sup>‡</sup>

### 1. はじめに

複数のサービスを提供するシステムにおいて、データあるいはモジュールが共有される場合がある。複数の役割を実施する処理をおこなうために、特定の役割あるいは目的に対応するように関係するモジュールやデータを割り当てて、それらの環境の中で動作させる。他の役割あるいは目的に対しては、別途同様に環境を整えて実施しなければならない。

更に、この環境が互いに関連しあう場合があるので、個々の環境で独立に実施させるのでは、個々の仕事の状態を反映した動作ができない。

環境毎に同時に制御させる仕組みを構築できれば、このような制約がなくなる、と考えられる。関連するモジュールや情報を、複数の環境の中に同時に存在できるようにすると、特定の環境に存在すると同時に、他の環境にも属することが可能となる。

我々は、共有空間を構築する Linda システム<sup>4)</sup>をベースとして、複数の環境を構成する機構の構築を目指している<sup>1)2)3)</sup>。

### 2. 情報場

上に述べた環境を対象とするモジュールや情報が存在する「場」として位置づけ、この場の中で、モジュールが情報を参照および操作しながら、役割を遂行して目的を達成する仕組みを実現する。

この目的のために、場をどのように構築し、どのようにデータの登録し、どのようにデータを参照し操作するのが焦点となる。そのために必要な要件は以下である。

データとモジュールを対とした基本構成要素とする

基本構成要素の振舞いに関する条件や制約を規定する環境となる場を導入する

場に対する基本要素の登録・参照・操作のための手段・プロトコルを備える

場は基本要素を共有する形態で、互いに関係をもつ

図 1 は、この要件の概念を示す。矩形で示すモジュールと情報の対が基本構成要素を表し、楕円が対応する場を示す。

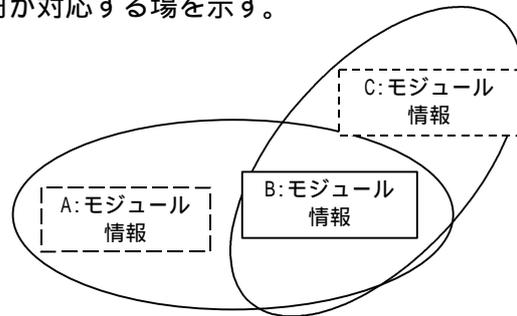


図 1 基本構成要素の図

### 3. 場の構築

共有空間を構築する上で、代表的な物として Linda システムがある。Linda は Carriero らによって発表された分散共有メモリシステムである。分散共有メモリとは、分散環境で共有されるデータ領域のことである。

一般に分散共有メモリは汎用的であるほど遅いが、Linda は共有オブジェクトに特化することで現実的な性能を実現している。Linda では、Tuple スペース (Tuple 空間) という共有メモリで同期および排他制御が行える

前述の基本構成要素は、環境を構成する場 "Field" の中に、情報を格納する "Tuple" と、役割を実施するモジュール "manager" により構成して実現する。Tuple に対する操作は、共有メモリを制御する Linda の手順により行う。この Tuple への操作は、当該 Field の外部からも可能とする。

このように構成することにより、場の中では、"Manager" の機能により場自身が自律的に機能する仕組みを備えることになる。

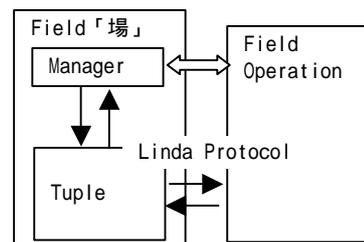


図 2 場の構成図

Autonomous collaborative information field based on Structured Tuple Space.  
Information science of Shonan Institute of Technology

更に、各々の場を組織化することを可能とし、並列的な関係、あるいは従属関係を持たせる。この組織化を階層構造と呼ぶ。

現在、分散共有メモリ方式に基づき Linda システムをベースとし Java 言語で構築された JavaSpaces が存在する。この JavaSpaces は、共有空間を提供し、オブジェクトを処理・共有・移住させる機能を備えている。

我々は、JavaSpaces の Tuple のデータ構成を利用して、次章に述べる方式により階層化された Tuple 空間を実現する。

#### 4. JavaSpaces によるの場の構築

JavaSpaces 内における Tuple 空間を階層化する手法を以下に示す。

階層構造の概念的構造は、図 3 に示すような木構造型になる。

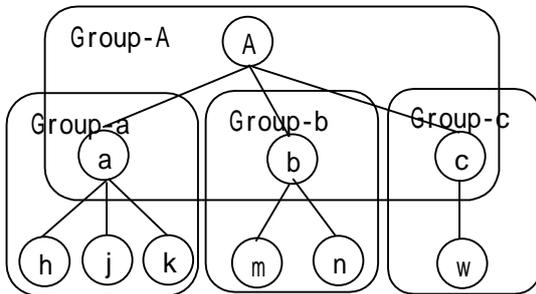


図 3 木構造型の構成図

構造は木構造となり、Group-A は 1 つの Tuple 空間を構成し、A はこの空間を管理・制御し、データ a、b、c を備える。Group-a は Group-A 中の a を親として Tuple 空間を構成している。

このことにより、ある条件でグループ別に分けられた空間内では、外部から、必要としている条件のデータに対する制御を要求された場合、データの種類の参照、存在の有無、そして操作が可能となる。

図 4 に、JavaSpaces における Tuple 空間の内部構成を示す。

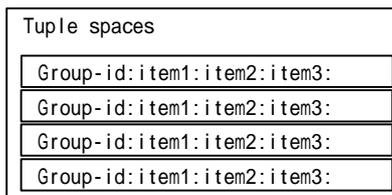


図 4 内部構成

図 3 で示す木構造を構成するために、tuple の一部に、group 種別を表す情報を組入れる。この group 種別情報に基づきシステムはグループ構成

を意識した制御を行う。

#### 5. 分散型場の構築

JavaSpaces に対するアクセスが、ネットワークを介して行うことを可能とするために、RMI を利用してシステムを以下のように構成する。

- Linda protocol を RMI の内部に包含させる
- 各ノードのインタフェース部分において、RMI への接合および分離を行う

図 5 にシステムの構成例を示す。

rmiregistry の起動。

JavaSpaces を起動し、エントリー用のスペースを提供する。

RMI を使い遠隔メソッドによるデータの受け渡しを確立する。

“write”、“read”、“take”などの処理が実行される

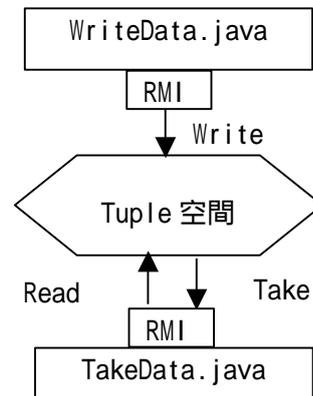


図 5 操作手順

#### 6. おわりに

本システムは、Java、JavaSpaces、Jini で作成した。開発環境は、JDK1.5.0 と Windows XP Professional Service Pack 2 である。

#### 参考文献

- (1) "マルチキャストディスカバリによる LookUpServer と CorbaServer の共用" 情報処理学会第 67 回全国大会、5V-5、矢島、鈴木、加藤、坂下
- (2) "マルチキャスト型 P2P による資源検索 Lookup システム"、情報処理学会、マルチメディア、分散、協調とモバイル(DICOMO) シンポジウム、2004、矢島研自、坂下善彦
- (3) "ネットワーク“場”における情報収集のモデル"、情報処理学会、マルチメディア、分散、協調とモバイル(DICOMO) シンポジウム、2003、坂下善彦、稲守久由
- (4) "Linda and Friend" IEEE Computer、August 1986、pp2634、Sudhir Ahuja、Nicholas Carriero、David Gelernter