

情報場を備えた JavaSpaces による情報共有空間の制御

矢島 研自 坂下 善彦
 湘南工科大学 大学院工学研究科

1. はじめに

分散処理システムを構成する場合の課題の1つは通信インタフェースの数とその管理である。

N 個のノードが存在する場合、それらの間で通信を行うには N! のオーダーのインタフェースの数が存在する。ノードの数が増大した場合におけるそのインタフェースを効率よく確実に管理するには、大きな負荷が存在する。また、関連するノードの間で実行されている全体の振る舞いの様子を知ること容易ではない。

分散処理システムでよく用いられる共有メモリの方式は、これらの課題を解決するために用いられることが多い。

本研究では、関連するノード間での相互コミュニケーションを簡便に行う方法として、更には関連するシステム全体の振る舞いをノード自身からも容易に観える仕組みを、共有メモリ方式に基づく JavaSpaces を利用して構築している。特に、関連するノードが参照する情報が存在する環境を情報の「場」と定義して操作している。

2. 共有メモリ方式による情報場

2.1 情報場

上記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¹⁾²⁾。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

基本構成要素に基づく振る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

複数の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、これら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。

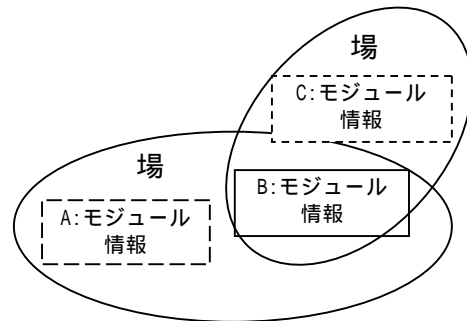


図 1 基本構成要素

2.2 JavaSpaces

JavaSpaces は、共有メモリ方式に基づく、Linda システムをベースとして構築されている。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³⁾。

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、Tuple スペース (Tuple 空間) と呼ぶ共有メモリを利用して、同期および排他制御を行うことができる。

3. 情報共有空間の制御

3.1 共有メモリ機構による情報の共有形式

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、役割に基づき振る舞う "モジュール" により構成する。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。

場の中では、モジュールは Tuple のデータを参照して自律的に機能する仕組みを備えることになる。

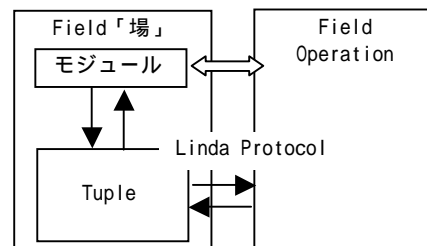


図 2 場の構成図

Operation to Information Share Space "Information Field" on JavaSpaces
 Kenji Yajima, Yoshihiko Sakashita
 Information science of Shonan Institute of Technology

また、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とを可能とし、この組織化を階層構造と呼ぶ。

JavaSpaces には、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、次に述べる方式により階層化された情報空間を実現する

3.2 階層構造

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 に示すような構造になる。

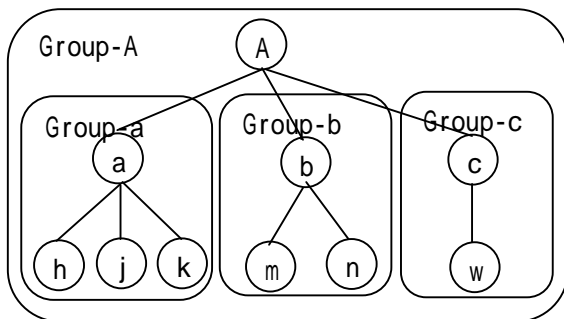


図 3 構成図

Group-A は 1 つの情報空間を構成し、A はこの空間を管理・制御し、モジュール a、b、c を備える。Group-a は Group-A 中の a を親とした、階層化された情報場を形成している。

この場合、ある情報場に存在する Tuple データは、他の情報場から、参照及び操作が可能とする。

図 4 に、階層構造への制御を行うために、Tuple の一部に、情報場の種別(図中"Group-A")を表す情報(図中"Group-id")を備える。

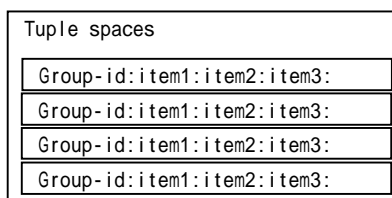


図 4 内部構成

3.3 制御

モジュールは、group 種別情報を識別子(id)としてグループ構成を意識した制御を行う。

Tuple 制御の具体的操作は、図 4 で示す構成を用いて、Group-id の部分に、Group を構築した代表者の A が id となる。A から派生して a、b、c 各々

が存在する。a、b、c が新しい Group を作る時は、新たな代表者となり、Group-id となる。item の部分は、代表者が生成したグループ内に存在する情報が、item として保存される。これにより、Group 同士の階層化と関連付けを可能とする。

4. 神輿モデルの実装

本研究では、この情報共有空間の機能を検証するために、神輿モデルを選んだ。神輿モデルは、分散協調ロボットのモデルであり、複数のプロセッサを用いて 1 つの物体を運搬 / 操作するためのモデルである。

神輿本体を親の場、前後左右の担ぎ手の集合を各々 4 つの子の場とする

4 つの担ぎ手の部分は、そこに参加している担ぎ手を場のメンバーとして扱い、そこで提供される担ぐ力を集約する

親の場は、4 つの担ぎ手の場のメンバーとの扱い、4 つの集約された力を更に集約することによって神輿の動きに反映させる。

上のように実装することにより、神輿の担ぎ手グループのメンバーはそのグループにある神輿の状態の情報を参照することにより、自らの役割を自律的に果たす。

5. おわりに

共有メモリ方式に基づく情報場を構築した。それぞれのサービス実行モジュールはそれ自身が関係する領域となる情報空間に置かれ、この空間がモジュールの実行環境となる構成とした。

実行環境が複数存在する場合に相互に関連を持ちながら実行させるために情報場を構造化させた。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 SP2 である。

参考文献

- (1) 矢島研自、坂下善彦、"階層構造型タプル空間に基づく自律協調の情報場"、情報処理学会第 67 回全国大会、5V-5
- (2) 坂下善彦、稲守久由、"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
- (3) Sudhir Ahuja, Nicholas Carriero, David Gelernter "Linda and Friend" IEEE Computer, August 1986, pp2634