# 協調作業支援のための JavaSpaces アクセスエージェントの開発

石田 隼人 樋口 昌宏
近畿大学

## 1. はじめに

本研究での協調作業とは，複数の参加者が「場」を共有して場に置かれた「モノ」を基に進めていくような作業を指す．例えば，オークションやテーブルゲームなどがそれにあたる．本研究の目的は，そのような協調作業を支援するためのアプリケーションを JavaSpaces を利用して分散環境上で実現することである．そこで，JavaSpaces を利用した協調作業支援アプリケーションの開発の際に有用と考えられる JavaSpaces アクセスエージェントの開発を行った．

## 2. JavaSpaces

JavaSpaces[1] は，Java オブジェクト「エントリ」の共有空間をネットワーク上に提供する Jini サービスであり，協調作業で用いられる場の表現に適していると考えられる．JavaSpaces にアクセスするクライアント群は書き込み，読み取り，取得（削除），通知の 4 種類のシンプルな命令を用いて JavaSpaces 内のエントリをやり取りし，協調作業を進めることができる．この「命令がシンプルである」という特徴は，JavaSpaces を用いたアプリケーションの実装を非常に容易なものにしている．しかしその反面，クライアント群は JavaSpaces 内に存在する全てのエントリに対してアクセスが可能であるため，協調作業で取り扱う情報に対して不正な操作が可能になってしまう．例えばトランプゲームにおいて伏せたカードがエントリとして JavaSpaces 内に存在した場合に，このカードエントリを不正に読み取ることや秘密裏にすりかえることが可能になってしまう．

## 3. JavaSpaces を用いた協調作業支援プリケーション

本研究の協調作業支援アプリケーションは，各クライアントが個々に JavaSpaces にアクセスして協調作業を進めるためのアプリケーションプログラム (以下 AP) を用意して JavaSpcaes にアクセスする構成を想定している (図 1)．この構成の利点は地理的に分散した協調作業が可能である点や，各クライアントが AP を自由に変更して利便性を高められる点が挙げられる．しかし，一部のクライアントが不正なプログラム AP' を用いて特定のクライアントが有利になるように協調作業を導くことが考えられる．例えば 2 節の最後で述べたトランプの例が挙げられる．AP' による伏せられたカードの読み取りや秘密裏のすりかえを防止するにはカードエントリに対する暗号化やアクセス制御などの機能を AP 内で実装する必要がある．しかし，

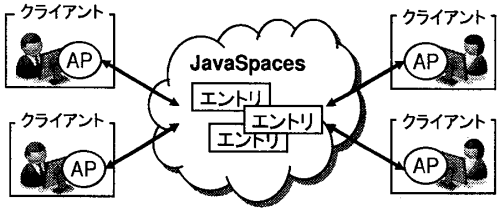


図 1 JavaSpaces を用いた協調作業支援アプリケーション

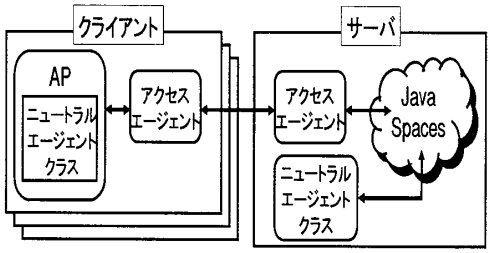


図 2 アクセスエージェント

AP 開発の際にそれらの機能を実装するのは非効率である．そこで，協調作業中に想定される不正行為を防止するために汎用的に用いることができるいくつかの機能をエージェントが提供することが望ましいと考えた．そのエージェントとして，本研究では AP 開発支援のための汎用エージェント「JavaSpaces アクセスエージェント (以下アクセスエージェント)」を開発した．

## 4. JavaSpaces アクセスエージェント

アクセスエージェントを用いた協調作業支援アプリケーションでは，AP から JavaSpaces への操作を全てアクセスエージェントを介して行う (図 2)．アクセスエージェントはクライアント側に配置されるモジュールとサーバ側に配置されるモジュールに分かれる．クライアント側に配置されるモジュールではサーバへの RMI 接続やクライアント識別のための証明書の管理を行う．一方，サーバ側モジュールは，クライアントからの要求に応じて以下で述べるような機能を提供する．また必要に応じてクライアント側モジュールを介して AP 内からクラスをロードする．

**機能 1** エントリ中のフィールドの暗号化
エントリ中のフィールドをアクセスエージェントが暗号化して秘匿を行う．以下，暗号化から復号化までの流れを簡単に解説する．まずクライアント A はアクセスエージェントの暗号化メソッドを呼び出す．暗号化メソッドは引数として，エントリ E，暗号化を

行うフィールドのリスト，復号化を許可するクライアント群を受け取り，返り値として暗号化されたエントリ CE を返す．CE は JavaSpaces にて書き込まれ，他のクライアント B がそれを読み取る．B は CE を復号化するため，アクセスエージェントの復号化メソッドを呼び出す．復号化メソッドは引数として CE と復号化を行うフィールドのリストを受け取り，B が復号化を許可されたクライアントであった場合にのみ返り値として復号化された E を返す．尚，A は復号化を許可するクライアント群を随時変更することができる．この機能を利用することで，例えばカードを伏せたまま別のクライアントに渡す，といった操作の表現が容易になる．

**機能 2** 読み取り・取得（削除）操作の制限
JavaSpaces に書き込むエントリに対して読み取り・取得（削除）操作が可能なクライアント群を設定し，エントリへの操作ができるクライアントの制限を行う．この機能は機能 1 とは異なり，エントリ自体の秘匿やエントリへの操作自体を制限するものとなる．また，JavaSpaces はシンプルな操作しか持たないため JavaSpcaes 内のエントリの検索の際に無関係なエントリを間違えて読み取る場合がある．この機能を利用することで，無関係なエントリへの操作を防止することも可能である．

**機能 3** エントリへの書き込み情報の付加
JavaSpaces に書き込むエントリに対して書き込んだクライアント名と時間を記述して，なりすましやすり替えを防止する．

**機能 4** ニュートラルエージェント機能
クライアントは AP の一部として java.lang.Runnable クラスを継承したニュートラルエージェントクラスを開発し，サーバ側モージュールにそのインスタンスを配置してクラス内の run メソッドを呼び出させることができる．これは，現実の協調作業において作業参加者以外に中立的なエージェントが必要な場合，そのようなエージェントを分散環境上で容易に表現するための機能である．この機能の単純な活用例としては，秘匿数値情報をその値を知ることなくインクリメントするなどが挙げられる．ニュートラルエージェントの処理は公正な処理を保証する必要があるため，メソッド呼び出しの際には事前に設定された規定数の承認クライアントの許可が必要である．この許可は承認クライアントが事前に登録したニュートラルエージェントクラスと呼び出す予定のニュートラルエージェントクラスが同じ場合にのみ与えられる．

機能 1，2 の利便性を高めるために，アクセスエージェントはクライアントのグループ化機能を提供する．また，クライアントの識別はアクセスエージェントが発行する証明書によって行われる．

## 5. アクセスエージェントの有用性確認実験

アクセスエージェントの有用性確認として，スコットランドヤードと呼ばれる推理ボードゲームを分散アプリケーションとして開発した．比較として今回は以下の 3 種類の AP を開発した．

| — | クラス数（行数） | エントリ数（行数） | 不正防止項目 |
|---|---|---|---|
| AP1 | 23（約 2400） | 2（約 140） | (3)*(4)* |
| AP2 | 26（約 3100） | 4（約 390） | (1)(2)(3)*(4)* |
| AP3 | 23（約 2700） | 2（約 300） | (1)(2)(3)(4)(5) |

表 1　AP の規模・不正防止項目

**AP1**　不正防止機能を一切持たない AP
**AP2**　アクセスエージェントを用いずに不正防止機能を実装した AP
**AP3**　アクセスエージェントを用いて不正防止機能を実装した AP

また，指標として AP は以下の 5 種類の不正防止が可能であるかどうかを調査した．

(1)　エントリ内の秘匿情報の読み取りの防止
(2)　エントリ内の秘匿情報の書き換えの防止
(3)　エントリ書き込み者のなりすましの防止
(4)　エントリの削除の防止
(5)　エントリの秘匿情報に対して操作が可能

各プログラムの規模・不正を防止できた項目を表 1 に記す．不正防止項目で*の付いた項目は防止ではなく，検知のみ可能な項目である．ただし，不正行為を行ったクライアントを特定することは困難である．AP2 では主に暗号化を用いて不正防止項目を達成し，AP2 で増加したエントリはそれに関するものである．AP3 では，不正防止項目の (1) を機能 1 を利用して達成し，項目 (2) を機能 1 と 3 で，項目 (3) を機能 3 で，項目 (4) を機能 2 で，そして項目 (5) を機能 4 を利用して達成した．

プログラムの規模に関して AP1 と比較した場合，AP2 は 700 行の増加に対して AP3 は 300 行の増加で済んだ．これは，AP2 での不正防止に関する記述の大部分がすでにアクセスエージェント内で提供されているからである．AP3 で追加された 300 行の約半分は現実の協調作業でも行われるようなゲーム進行の公正性の確認作業で，AP1 では元々必要のなかった部分である．例としては，他のプレイヤーがボード上の駒を動かす際にルール通りの移動をしているかという確認などが挙げられる．

## 6. 結論

本研究では，協調作業支援を目的としたアプリケーションを JavaSpaces を用いて分散環境上で実現する際に有用と考えられるアクセスエージェントを開発した．開発したアクセスエージェントを活用することで AP 内の不正に対する処理を簡明に記述することが出来た．今後は開発支援に重点をおいた改良を行っていきたいと考えている．

## 参考文献

[1] JavaSpaces Service Specification Version 2.0 Sun Microsystems(2003)