

## 非同期階層型エージェントアーキテクチャについて

- Asynchronous-Hierarchical Agent Architecture -

江谷為之, シドニー・フェルス, アルミン・ブルーダリン, シルビオ・エッサー, 間瀬健二  
Tameyuki Etani, Sidney Fels, Armin Bruderlin, Silvio Esser, Kenji Mase

ATR 知能映像通信研究所

ATR Media Integration & Communications Research Laboratories

e-mail: {etani, fels, armin, esser, mase}@mic.atr.co.jp

### Abstract

This paper is to report the current progress of our project that defines an agent architecture to provide a conceptual and practical framework for agent systems design. Our agent architecture is based on multi-layered structure model that controls multiple functional units in different knowledge abstraction levels. The hierarchy is composed of 5 levels: 1. motivation and needs, 2. plan/policy, 3. behaviour, 4. actions/sensations 5. action/sensor units. Each agent has a knowledge monitor that manages interlayer communication and access to the world in asynchronous manner. This knowledge monitor has 3 major functions: activity manager, knowledge base, and resource manager. We believe this multi-layered structure in asynchronous execution provides an effective solution against complex problems that require both long term thinking and realtime response. We have started using our architecture to design multiple agents playing board game called Malefitz. The current status of this project and it's future direction is reported.

### 1 はじめに

エージェントに対する定義/要件は、その適用分野や用法によって多様化しており、一意的に決めることは困難となっている [1]。しかしながら、それらの様々な文脈における共通的な特徴を抽出し、性格づけをするとすれば、「なんらかのタスクを実行するにあたり、環境条件を認識し、環境下における他の実体、エージェントまたは人間と競合/協調しながら、目的を遂行する自律型計算機プログラム」ととらえられる [2] [3]。ここで自律とは、計算機プログラムが人間の干渉、介入無しで環境認識や裁量範囲における合理的な意志決定/実行を行えることを言う。

本稿ではこのような定義のもと、まずエージェントシステムを構成するための基本アーキテクチャとして現在設計している、「非同期階層型エージェントアーキテクチャ Asynchronous-Hierarchical Agent (A-HA) Architecture」を検討する。A-HA では、1. 動機 (Motivation) 2. 計画 / ポリシー (Plan/Policy) 3. 振舞い (behavior) 4. 行動 (Actions/Sensations) 5. 行動単位 (Action Units/Sensor Units) の 5 階層に分割され、各層が隣接する層とメッセージングを行うことで、エージェントの動作が決定する。またこのメッセージングや環境との入出力を非同期的に実行するため知識モニタについても述べる。

さらに、このアーキテクチャを用いて部分的に実装中であるマルチエージェント対戦型ボードゲーム (Malefitz) をもとに、エージェントに要求される機能要件について検討する。

### 2 エージェントの階層的機能分割

エージェントが環境適応および自律的な行為選択を行うためには、「メッセージを送る」「移動す

る」などといった個々の単一動作の他に、環境推定を行ったり、大域的行動計画を立案するなど広範囲なレベルに渡る処理を行う必要がある。これは柔軟性、反射性など相反する特性をもった処理を同時に行う事でありモノリシックな構造によってこれらを実現するのは容易ではない。

この問題の解決には、従来より、計算主体内に処理レベルに応じた階層的な連携構造を持ち、問題を各階層ごとに分割する方法などが提案されている [4] [5] [6]。例えば文献 [7] [8] において提案されている the InterRRaP system では、階層型アーキテクチャと Belief, Desire, Intention (BDI) アーキテクチャを統合し、プラグマティックなアプローチでエージェントをモデル化しているが、これは我々のアプローチと同様のものである。このモデルによれば、エージェントの機能階層は知識レベルの抽象度に応じて behavior-based layer (BBL), local planning layer (LPL), cooperative planning layer (CPL) の 3 階層に分割され、各階層の抽象度に応じた知識ベースを持ち、外部環境に対し world interface unit (WIF) を通じてアクセスする。これは我々のモデルにおいて、5つの各階層が知識モニタを通じて各レベルの知識にアクセスし、外部環境に対しては行動単位層を経由して行動および認識を行う事に対応していると考えられる。

行動の階層分割については Norman の「行為の七段階理論」 [9] において一般的な近似モデルが示されている。この七段階は人間が何らかの行動を行う場合の状態を 1. ゴールの設定、2. 行為意図の発生、3. 行為系列の生成、4. 行為系列の実行、5. 外界変化の知覚、6. 知覚の解釈、7. 外界解釈の評価に分類している。我々はこれらの人間の行為/知覚の階層分類はエージェントが環境に対し合理的な行動/認識を行うモデルとして充分な妥当性を持つと

同時に、エージェントの環境適応や個性化のための制御を局所化する指針になると考えている。たとえば、同じ意図に対し行為系列の発生の仕方を変える、また同じ知覚入力に対して解釈/評価基準を変更するだけで、エージェントの振舞いが変わると考えられる。このような制御の局所化はデザインの効率性、柔軟性の点から大きなメリットである。

またエージェントと人間とのインタラクションという側面を考えると、その外部表現として生物や人間などのアニメーションキャラクタなどを利用することが考えられるが、それらのキャラクタの制御についても振舞いレベルから動作単位にいたるまでを階層的に機能分割することで、自然な動作を表現することが提案されている [10] [11]。これらにおいては、生物/人間の動作という明確なモデルに従って制御されるため階層間の関係も明確なものとなる。たとえば「急ぐ」という意図のもとで「ゆっくり歩く」という行為は相反するものであり、また「歩く」という行為で右手と右足を同時に前へ出すことは合理的ではないといったように、自然な動作表現のためには上位階層の動作が下位階層に対する制約として作用する。

### 3 非同期階層型エージェントアーキテクチャ (A-HA)

我々はエージェントアーキテクチャをデザインするにあたり、これまでに提案されてきた様々な成果を統合して、多様なエージェントの形態に、より共通かつ基本的なレファレンスプラットフォームを提供する事を目的とした。そのため前述のようなマルチレベルの制御、行為の階層性、擬人性のある外部表現などの一環性を考慮し5階層のモデルに基づく階層型アーキテクチャを検討した。従来提案されてきたエージェントのモデルでは、知識処理もしくは振舞い/行動レベルのどちらかに重点をおいたものが多く、その両者を統合的に扱うものはまだ少ない。またこれらの統合にあたっては、その階層分割とともに実時間性、適応性の異なる処理を非同期的に実行制御する基本メカニズムが重要であると考えられ、それについても検討を行った。

アーキテクチャの構成を図1に示す。大きく分けて多層処理部、知識モニタ (Knowledge Monitor) の2つの基本要素で構成される。

まず多層処理部はトップダウン型の行動系およびボトムアップ型の認識系からなり、上位より下位へ向かって次のような階層特性を持つ。

- 動機 (Motivation/Needs)
- 計画 / ポリシー (Plan/Policy)
- 振舞い (Behavior)
- 行動 / 環境認識 (Actions/Sensations)
- 行動単位 / 認識単位 (Action Units/Sensor Units)

各層は知識モニタを通じて隣接する層と通信を行い、層間での指令や情報のフィルタリング、環境情報などを交換する。エージェントと環境/他エージェントとのインタフェースは行動単位/認識単位層である。この層で認識された環境情報などが、上位層へ向かって伝搬する過程で各層における、情報解釈、推論などが行われ付帯情報として、知識モニタ内に蓄積、更新される。各層は、それらの環境情報と付帯情報を用いてそれぞれ独立に行動決定/修正や、環境認識/推論を行う。

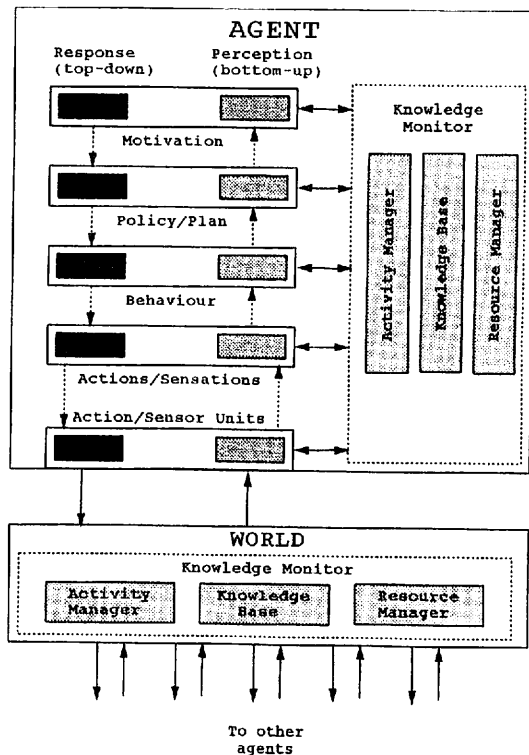


図1: A-HA エージェントの基本構造

次に知識モニタ (Knowledge Monitor) はつぎの3つの部分から構成される。

- アクティビティマネージャ (Activity Manager)
- 知識ベース (Knowledge Base)
- リソースマネージャ (Resource Manager)

これらは、エージェント内部での各層における非同期実行制御、データリポジトリ、リソースの競合制御といった、エージェントの動作を制御する基本部である。また知識モニタはエージェントの外部環境側にも存在し、エージェント間通信、環境変更などの制御を扱う。

以下、各階層における基本タスクおよび知識モニタの仕組みについて述べる。

### 3.1 エージェントの階層レベル

エージェントの階層における、最上位のタスクスペースは、**動機 (Motivation/ Needs) 層**であり、これはエージェントの動作の基本的方向性を規定するいくつかのパラメータ (指針) を取り扱う。後述するボードゲームにおいては動機として、

- 「ゲームに勝利する」
- 「最短時間でラウンドを終了する」

などがあり、これらの動機の優先度はエージェントに与えられた制約条件 (時間制限、ゲームの進行度合など) およびその条件に関する認識系からの環境入力によって決定される。また動機層は隣接する計画 / ポリシー層に対し、これらの動機を充足するプランニング、戦略の生成を指令する。

計画 / ポリシー層 (Plan/Policy) では、与えられた動機条件に応じて環境モデルをもとに大域的な全体計画を立案し、それを下位の振舞い層への指令として分解する。ボードゲームの場合、「勝利」「最短時間」などの動機に準拠した経路プランニング / ゲームプランニングを行い、それを実行するための戦略を抽出し振舞い層へ指令する。その際環境モデルからの入力としては、有効経路、プラン実行の阻害要因 (競合エージェント、障害物の存在) 他エージェントのポリシーなどがある。

第3の振舞い層 (behavior) における振舞いとは、行為目標達成のための行動群を指し、ここでは「敵を阻害する」「仲間を助ける」「前進する」といった、上位層 (計画 / ポリシー層) から与えられた行為目標に対して、何らかの行為系列を生成することを言う。たとえば、ボードゲームにおいて、「敵を阻害する」という行為目標達成のためには、「障害物を置く」「敵を押し戻す」などの行為系列が生成可能であり、それらの選択は環境条件によって決定される。

行動 / 環境認識層 (Action/Sensation) は振舞い層で生成された行動群のそれぞれの単位であり、単独では目標指向性をもたない。またそれぞれの行動が外部環境に対して作用するレベルに分割したものが、行動単位 / 認識単位層 (Action Units/Sensor Units) である。これは、たとえば「歩く」と言う (目標指向でない) 行動が、外部環境に対しては、「右足を前に出す」「左足を前に出す」といった、小行動から成立していることに相当する。

以上のような構造によりエージェントのメンタルレベルから、アクションレベルまでの動作を連続的な階層単位の連係によって実現できる。

### 3.2 知識モニタ

まず知識モニタのうちアクティビティマネージャの仕組みについて述べる (図 2)。アクティビティマネージャは、エージェント内部における各層の非同期スケジューリングおよび、スケジュールルールの

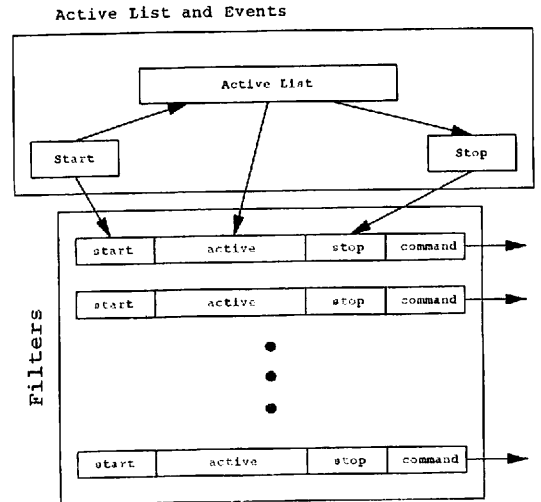


図 2: アクティビティマネージャのイベント / アクション管理

変更、更新を行う。そのためここでは次の3種類の情報を用いる。

発生イベント (start)

新規発生した環境事象 (もしくは隣接層からのアクショントリガ)

有効イベント (active)

既発生で、かつ現在も有効な環境事象

終了イベント (stop)

終了 / 消滅した環境事象 (もしくは隣接層からのアクショントリガ)

アクティビティマネージャはこれらのイベントとエージェントの各動作 (処理) との制御フィルタ (Filters) を管理しており、それぞれのイベントの組合せが、各層内の処理起動条件となる。エージェント各層とはクライアント / サーバモデルで連携され、エージェントからの

- start 'event'
- stop 'event'

メッセージにより環境事象の発生・終了を受信すると、制御フィルタのリストを検索し、対応する層に処理活性化要求メッセージを送信する。図 2 は各イベントと制御フィルタの関係を表している。

このようなアクティビティマネージャを介した間接的なメッセージングは、プランレベルとアクションレベルのような即時性要求の異なる層の処理が、互いに独立して動作する事ができ、また環境に対するアクセスとそれらの処理とを分離できるため、エージェント階層の非同期制御にとって重要と考えられる。

アクティビティマネージャが管理するこれらの制御フィルタは、エージェントの各層から操作・変更が可能であり、現在フィルタ操作メッセージは次の4つが用意されている:

#### addFilter

アクティビティマネージャの制御フィルタリストに新規に追加する。

#### removeFilter

アクティビティマネージャの制御フィルタリストから削除する。

#### activateFilter

既登録の制御フィルタリストを活性化する。

#### deactivateFilter

現在有効な制御フィルタを不活性化にする。

文献 [12] における the Will system でも同様の考え方により、環境事象入出力と制御を分離することが提案されている。the Will system では、環境からの事象入力をシステム内の Memory モジュールおよび Concerns システムが各モジュール毎にその関連度 (relevance) に応じて、ソーティングを行うことで、モジュールの独自性を保証する。これに対する我々のアクティビティマネージャの特徴としては、各モジュール自身が関連する環境事象をアクティブにセンサリングし、かつその事象に対する振舞いを制御フィルタ操作によって変更することができる点にあると言える。これは特定モジュールによる集中ソーティングに比べ環境事象に対する応答性、モジュールの自律性に優れていると考えられる。

知識ベースは、エージェントの外部環境のモデルや状態、エージェントの内部状態などを表すデータであり、エージェントが外部環境に対する行動決定を行うためのデータレポジトリとして作用する。たとえば後述のボードゲームでは、盤上の駒の位置、エージェントどうしの信頼関係 (敵、味方、中立) などの情報を保持している。

リソースマネージャは内部リソースの競合制御 / 優先制御を行う。おもにこれらは階層間での環境モデルやフィルタ操作の順序 / 優先度の管理を行う。

また図 1 にもあるように、以上のような知識モニタはエージェントの外部環境側にも存在し、エージェントの環境操作、環境変化の検出、エージェント間の通信制御などを行う。

## 4 エージェント環境: Malefitz

エージェントの環境としては、ロボットによる経路探索やネットワークにおける情報探索、情報管理といったものが従来からあるが、現在では RoboCup Soccer などゲームを素材としたものが進められている [13]。これはこういったゲームが実世界性と分散協調のためのモデル環境として適度な複雑性を有しているためと考えられる。現在我々は、エー

ジェントの動作環境として、Malefitz というマルチプレイヤー型ボードゲームの環境を用いて、エージェントアーキテクチャの検証を行っている。以下 Malefitz Game の簡単な説明と環境としての妥当性について述べる。

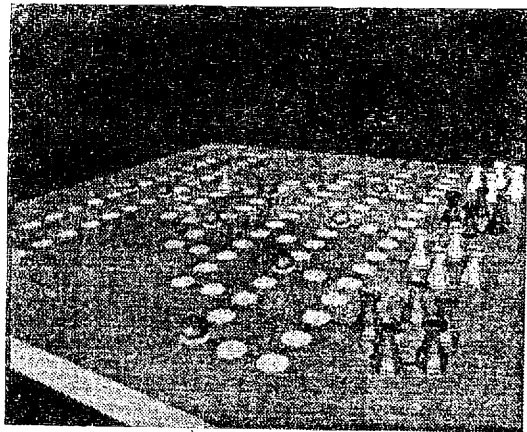
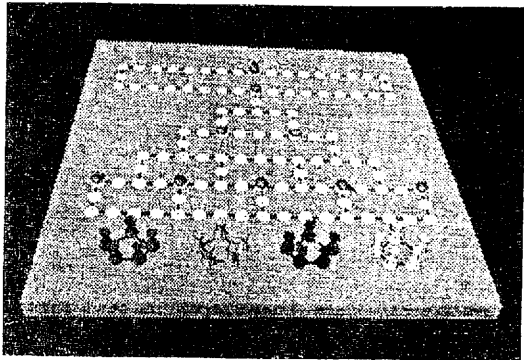


図 3: Malefitz ゲームの環境

Malefitz は盤面に配置された 4 人のユーザの駒をダイスの目に応じて操作し、最終ゴールに到達することを目的とした、すごろく型ボードゲームである (図 3)。

ゲームプレイヤーはそれぞれ blue, red, green, yellow の 4 つのホームポジションに分かれ、各プレイヤーには 5 つのピース (駒) が与えられる。各プレイヤーは順番にダイスを振り、その目に従ってピースを進める事ができる。基本ルールとしては、

- 最終ゴール地点に一番先にどれか 1 つでもピースが到達すれば、そのプレイヤーの勝ち。
- プレイヤーは 5 つのピースのうち任意のものを動かせる。ただし、一回の順番につき、動かせるのは 1 つのピースのみとする。

- サイコロの目に応じた移動先の地点に自分自身のピースがなければ、移動できる。その場合進路の途中で他のピースがあっても飛び越しができる。
- 移動先の地点に他プレイヤーのピースがあった場合、そのピースの地点に進むことができる。その場合その地点にあったピースは、ホームポジションまで戻される。
- 進路の途中でいくつかの障害物(石)があり、障害物の飛び越しはできない。
- 障害物が移動先地点にあった時には、その障害物を捕ることが出来る。その場合障害物は任意の場所におくことができる。

などの条件がある。

エージェント環境として考えた場合このゲームの特徴は以下のようにまとめられる。

- 「ゲームに勝つ事」「とにかく早く終わる事」などプレイヤーが持ちうる目標が明確であり、かつ複数存在しうる。
- エージェント間のコミュニケーションを持たせることで、エージェント間での協調問題解決(障害物の排除など)、競合(同盟関係の破綻、報復など)といった社会的な関係が発生しうる。
- 障害物の排除、再配置、他プレイヤーへの攻撃、救済などといったゲーム戦略、経路戦略などの推論/意志決定と同時に、不確定要素を含んだ環境(サイの日、他プレイヤーの動きなど)への適応を行う必要があり、これは熟考性と即応性のバランスが問題となる。
- 上記に関連して、相反する処理特性の解決を通してエージェントの非同期階層性の検証を行うことができる。
- プレイヤのうち何人かを人間が行った場合、ゲーム環境を通して人間とエージェントとの直接的なインタラクションが発生する。

以上のように、エージェントの動作環境としては、十分な複雑性をもったモデルと考えられる。

## 5 Melefitz におけるエージェント階層

以上のようなボードゲーム環境における、A-IIA エージェントの階層別動作は現在のところ部分的な実装段階であり次のようなものとなっている。

動機層では「ゲームに勝利する」事を目標として、ゲームの序盤、中盤、終盤にそれぞれ、以下のような基本戦略方針を立てる。

### Conservative Policy

他プレイヤーとの協調性を優先する

### Random Policy

他プレイヤーとの協調/競合をランダムに選ぶ

### Progressive Policy

ゴールへの接近を最優先にする

また計画/ポリシー層では動機層からの戦略方針に対して、他プレイヤーおよび自分のゲーム上での位置、順位関係を考慮して具体的な戦略化を行う。そして振舞い層に対して次のような行為指標を示す。

### Attack

他プレイヤーへの攻撃または進路妨害

### Moderate

他プレイヤーをなるべく避けて、道をあける

### Progress

ゴール地点に向かって前進する

振舞い層ではさらにそれらの指標からより具体的な行為系列へと展開する。その際、ゴール地点からの可動位置までの距離および可動位置上の他プレイヤーや障害物の有無が考慮される。行為系列としては、次のようなものがある。

### Distrub

障害物を取得し、他プレイヤーの進路を妨害する

### Kick Out

他プレイヤーを直接攻撃する

### Go Forward

ゴールに一番近い手を選んで前進

### Go Sneaky

ゴールに対して一番遠い手を選んで前進

行動層では、振舞い層で生成されたこれらの行為系列を、ボード上での操作するピースおよび移動先位置の特定を行ったのち、次の2つの行動に分解する。

### Move Piece

あるピースを決められた位置へと動かす

### Move Stone

ある障害物を決められた位置へと動かす

最後に行動単位層では、これらの行動をさらに以下のようなプリミティブに展開し、外部環境(ゲームボード)を操作する。

### Get Piece

ある位置のピースを捕る

### Put Piece

ある位置にピースを置く

### Get Stone

ある位置の障害物を捕る

### Put Stone

ある位置に障害物を置く

これらが Malefitz Game におけるエージェントの基本動作を表しているが、これら以外にも外部環境に対する操作としてダイスを回す、外部制御フィルタを操作するなどがあり、それらは行動単位層経由で行われる。

## 6 おわりに

非同期階層型エージェントアーキテクチャの概要をボードゲーム環境を例にとり、検討してきた。以上のような非同期的に動作する多階層構造は、柔軟性、反射性に優れ、エージェントのような変化する環境における適応性が求められるものには有効であると我々は考えている。それらの検証のためのボードゲームを用いた実験システムは現在のところ、基本部分の実装が終えたばかりであり、エージェントシステムとしては、今後次のような様々な機能を追加し、さらに検討する予定である。

- 階層に応じた学習 / 推論機構
- 多重ゴール、時間制約などより複雑な環境の取扱い
- エージェント間の直接的コミュニケーション

また、本アーキテクチャの応用性を検討するため、ナビゲーションや情報検索などゲーム以外のフィールドへの適用、ならびに仮想的な擬人化キャラクターなどエージェントインタフェースの導入を行っていきたいと考えている。

## 謝辞

本研究を進めるにあたり御指導いただいた(株)エイ・ティ・アール知能映像通信研究所の酒井保良会長ならびに中津良平社長に深く感謝いたします。また本研究内容について日頃より議論していただく研究所ならびに第2研究室の皆様にも厚く御礼申し上げます。

## 参考文献

- [1] 石田孝: “エージェントを考える”, 人工知能学会誌, 10, 5, pp. 663-667(1995).
- [2] S. Fels, K. Mase, A. Brudelin, T. Etani and S. Esser: “Agent 99: Implementing a simple card game using agents”, 知能情報メディアシンポジウム(1996).
- [3] 間瀬健二, シドニー・フェルス, 江谷為之 アルミン・ブルーダリン: “インタフェース・エージェントに関する基礎検討”, 情処研報 SIGHI69-8 情報処理学会, pp. 55-60(1996).
- [4] R. A. Brooks: “A robust layered control system for a mobile robot”, IEEE Journal of Robotics and Automation, 2, 1, pp. 14-23(1986).
- [5] I. A. Ferguson: “TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents”, PhD thesis, University of Cambridge, UK(1992).
- [6] R. P. Bonasso, D. Kortenkamp, D. P. Miller and M. Slack: “Experiences with an architecture for intelligent, reactive agents”, Intelligent Agents II (Eds. by M. Woolridge, J. P. Müller and M. Tambe), Springer, pp. 187-202(1995).
- [7] K. Fischer, J. P. Müller and M. Pischel: “A pragmatic bdi architecture”, Intelligent Agents II (Eds. by M. Woolridge, J. P. Müller and M. Tambe), Springer, pp. 203-218(1995).
- [8] J. P. Müller, M. Pischel and M. Thiel: “Modelling reactive behaviour in vertically layered agent architectures”, Intelligent Agents (Eds. by M. Woolridge and N. R. Jennings), Springer-Verlag, pp. 261-276(1994).
- [9] D. Norman: “The Psychology of Everyday Things”, Basic Books Inc. New York(1988).
- [10] B. M. Blumberg and T. A. Galyean: “Multi-level direction of autonomous creatures for real-time virtual environments”, SIGGRAPH 95, pp. 47-54(1995).
- [11] A. Bruderlin: “The creative process of animation human movement”, Knowledge-Based Systems(1997).
- [12] D. Moffat and N. H. Frijda: “Where there's a will there's an agent”, Intelligent Agents (Eds. by M. Woolridge and N. R. Jennings), Springer-Verlag, pp. 245-261(1994).
- [13] : “Internet Soccer: Trends of researches on Multi-agent systems”, Info-Tech '96, Kobe, Japan(1996).