

動的適応性をもつモジュラー型基盤ソフトウェアの提案

久保田 稔

千葉工業大学工学部 〒275-0016 千葉県習志野市津田沼 2-17-1

E-mail: kubota.minoru@it-chiba.ac.jp

あらまし PC, 携帯電話, 組込み機器やセンサ等のノードが様々な場所に配置され, それらがネットワークで相互接続されている環境において, 十分な機能を持たないノードを動的に組み合わせて, 新たなサービスを構成するために必要なソフトウェアシステムについて述べる. 本稿では提案システムの構成と応用例について述べ, システム構成要素が実現しなければならない機能要求条件を明らかにする. またこれらの条件を満たすために必要な技術として, 動的ソフトウェア更新, ノードの連携, サービスの動的構成について考察する. 提案システムは, ユーザの要求条件と環境条件に応じた適切なサービスを提供することを狙いとする.

キーワード ユビキタスコンピューティング, 動的ソフトウェア置き換え, 組込みシステム

Proposal for Dynamically Adaptive Modular System Software

Minoru KUBOTA

† Faculty of Engineering, Chiba Institute of Technology 2-17-1 Tsudanuma, Narashino-shi, Chiba, 275-0016 Japan

E-mail: kubota.minoru@it-chiba.ac.jp

Abstract The software system is described which allows to compose a new service in the environment where many nodes, including PCs, cellular phones, embedded equipment, sensor and so on, are arranged to be in a wide range of places, and are interconnected by the network. The system composes the new service by dynamically combining the nodes which don't have enough features for themselves. This paper describes the architecture of the proposed system and its applications, and clarifies functional requirements that system components must realize. Also, as the technology which is necessary to meet these requirements, it considers about the dynamic software update, the cooperation of the nodes, the dynamic service composition. The proposed system aims at providing an appropriate service which satisfies user requirements and environmental condition.

Keyword ubiquitous computing, dynamic software update, embedded system

1. はじめに

ユビキタスコンピューティング環境においては, PC や PDA, 普及が著しく高機能化が進む携帯電話, 家電, あるいは組込み機器やセンサや等, 極めて多数の CPU (以下, ノードと呼ぶ) がネットワークに接続され, 互いに情報を送受しながら多様なサービスを提供していく. このようなノードに搭載されるソフトウェアも極めて多様な要求条件を満足する必要がある. 一方, これらの要求条件は常に変化していくことが予想される.

したがって, ノードに必要となるソフトウェアをあらかじめ全て作り込んでおくことは困難である. この条件の下で, サービスの要求条件に対応していくためには, ソフトウェア自身が, 環境条件やユーザからの

サービス要求条件の変化を検知し, 機能を適応させていく方式が必要になる.

またユビキタスコンピューティング環境のノードには, センサや IC カード等, コンピュータとしては十分なリソースをもたないものが多数利用されるようになることが予想される. 本稿ではこのようなノードを「マイクロノード」(以降, M ノード)と呼ぶ.

M ノード上で実行されるソフトウェアの単位を「マイクロソフトウェアコンポーネント」(以降, MSC)と呼ぶことにする. ここでマイクロとは M ノード用ということで, ソフトウェアの規模が小さいことを必ずしも意味しない.

M ノード単体では, 必要とされるサービス要求条件を満足させることができない場合が生じるが, M ノー

ドを複数組み合わせることで、条件を満足させるようにさせることが期待できる。Mノードの集合により特定の機能(サービス)を実現するものを、「マイクロノードグループ」(以降、MNグループ)と呼ぶことにする。

MNグループに属するMノードは変化する。すなわち時間の経過により、MNグループに新たなMノードが追加されたり、MNグループから離脱することがある。本稿では、Mノードは同時に一つのMNグループにしか属さないものとする。

Mノードを接続するネットワークを「マイクロネットワーク」(以降、Mネットワーク)と呼ぶことにする。MNグループは、一つ以上のMネットワークで接続される。

一つのMネットワークには、他のMネットワークあるいは基幹ネットワークへのゲートウェイ機能をもつMノードが存在するものとする。これを「マイクロノードマネージャ」(以降、MNマネージャ)と呼ぶ。ここでいう「公衆ネットワーク」はインターネットを想定する。

MNグループが実現する機能(サービス)を「アドホックサービス」(以降、サービス)と呼ぶことにする。本稿ではサービスを、Mノードへの入力を、そのノード上あるいは適切なMノードに転送して処理し、その結果を必要があれば転送した上で適切なMノードか

ら出力する処理、と定義する。またこれらの処理を行うプログラムを入出力情報処理プログラムと呼ぶ。これはアプリケーションに相当する。

公衆ネットワーク側に、MノードとMNグループの情報を管理するノードが存在する。これをマイクロノードグループマネージャ(以降、MNGマネージャ)と呼ぶ。詳細は後述するが、MNマネージャはMSCを管理し、必要に応じてそれをMノードに送る。

本稿では、上記の構成要素からなるシステムを構成、管理、制御するソフトウェア(OS)の基本技術について提案する。これには各ノード相互の通信機能を含む。図1に想定するシステムの構成の概要を示す。

2. 関連研究

提案システムを実現するソフトウェアに必要な機能としては、(a)Mノードの制御、(b)Mノードの連携協調動作、が必要となる。(a)については、センサなどのための小規模のOS[1]の提案があるが、OS自体の動的な機能変更は含まれていない。また条件に応じて、OSの機能を変化させるためには、OS内部のパラメータを変更することを主とすることが通常である。センサネットワーク[2]におけるセンサが本研究におけるMノードに対応するが、センサが動的に機能を変化させる技術や、センサ単体では実現できない新たなサービスを提供する技術、ノード(センサ)が移動させる

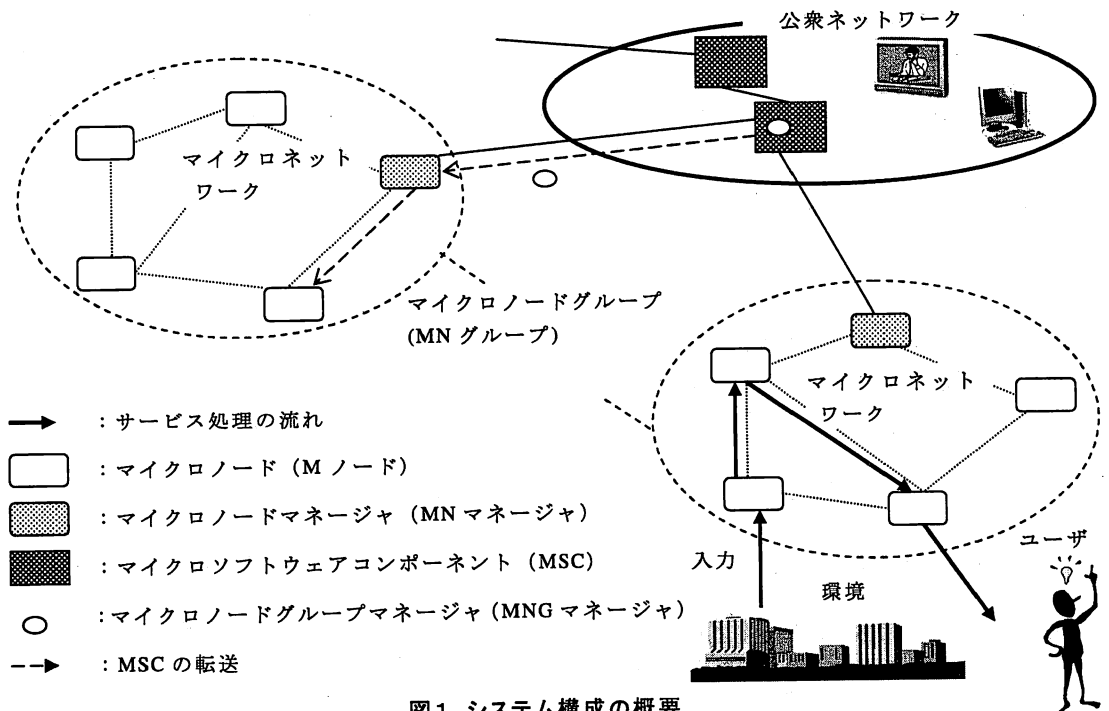


図1. システム構成の概要

技術に関する研究はあまり進んでいない。

本提案では、ソフトウェアの更新により OS の機能自体も変更していくことを狙いとする。このため、必要なソフトウェアをネットワーク経由で転送して M ノードの機能を更新する機能と、ネットワーク経由で他の M ノードのソフトウェアを実行する機能を提供する。また、ソフトウェア更新を用いて M ノードの機能分担を変更することにより、MN グループの機能を条件に応じて自律的に適応させていく機構を提供することを目標とする。

(b)については、情報機器を動的にネットワークに組み込む技術[3, 4]の開発、サービスや機能を自律的に構築していく創発型システムの研究[5, 6]、プログラムを転送することにより情報機器の連携を行う研究[7]、が行われているが、これらは十分な機能をもったコンピュータ上で実現されることを前提とする。本研究では、単体で十分な機能を提供できない M ノードが協調し、自律的にノードの連携関係を適応させ、ノードが相互に機能を補完しながら、サービスを提供することを狙いとする。

3. システム構成と応用例

提案システムの基本的な構成とその応用例について述べる。

3.1. システム構成

(1) M ノードの機能

M ノードは、M ノード間の通信機能と、その他の情報入出力機能の一つ以上もつ。ここで、情報の入力元と出力先は環境とユーザである。例えば、環境からの入力を担当する M ノードはセンサであり、環境への出力を担当する M ノードは環境に対して動作を行うアクチュエータである。また、ユーザからの入力を担当する M ノードの例としてキーボードであり、ユーザへの出力を担当する M ノードの例として各種のディスプレイがある。

(2) M ノード間の通信

M ノード間のネットワークは無線アドホックネットワークを想定する。M ノードは多種多様なものが使用されることが想定される。したがって様々なものがあるアドホックネットワークプロトコル[8]のうち、特定のものに限定することは困難である。提案システムでは、必要に応じて複数のプロトコルを使い分けることとする。

(3) MN マネージャ

MN マネージャも M ノードの一つであり、M ノードと同じプロトコルを用いて通信可能である。前述のように MN マネージャは、M ネットワーク間のゲートウェイ機能をもつ。これに加え、M ネットワークに属す

る M ノードの構成情報も管理する。したがって、他の M ノードに比べて多くのリソースを必要とする。実際の環境では PC や PDA あるいは高機能な携帯電話が MN マネージャの役割を担うものとする。

MN マネージャは移動可能である。新たに M ネットワークに加わったとき、その M ネットワークに既に別の MN マネージャがあれば、それが管理する構成情報加わった MN マネージャに伝える。

(4) MNG マネージャ

MNG マネージャは、M ネットワークと MN マネージャのシステム構成情報を管理すると共に、ソフトウェアの動的更新に用いる MSC を生成・管理する。システム構成情報には M ネットワークの識別子や MN ノードのアドレスがある。MN マネージャと MNG マネージャ間の通信は、公衆ネットワークすなわちインターネットを利用するものとする。

MNG マネージャは必要に応じて動的に MSC を生成する。実行される M ノードの構成情報に応じて、MSC はあらかじめコンパイルされて実行形式になっているものとする。MNG マネージャのリソース節約のため MSC の転送時に動的にコンパイルする方式は検討課題である。

3.2. 応用例

ユビキタスコンピューティング環境では、環境からの情報の取得が主要な技術である。提案システムはその一つであるセンサネットワーク等への適用を狙いとする。想定している応用例について述べる。

(1) アドホックロボットシステム

移動するロボットがあるとき、移動する経路上で利用可能なセンサ等を組み合わせ、さらに移動ロボットを高機能化することが可能である。たとえば経路上のセンサからの視覚情報を利用することで、ロボットに組み込まれたセンサからの情報だけはカバーできない部分を補い、ロボットがより適切な行動をとることができる。また、環境に作用するアクチュエータがあれば、ロボットだけでは実行できない操作、例えば経路上にあるドアの開閉も可能となる。本方式は、移動ロボットの代わりに車椅子を想定すれば障害者支援システム、自動車を想定すれば高度交通システムにも応用できる。

(2) 行動記録システム

今後、個人の活動を記録する要求が高まると予想される。従来のインターネットを経由して記録を収集することも可能であるが、記録する情報と個人の対応づけ、あるいはそれに関連したプライバシー保護の点で問題が生じる[9]。提案システムでは、サービスを提供する M ノードが属している M ネットワークの情報をを用いることにより、位置情報を新たに求める必要はな

い。またアドホックネットワークを用いることで、情報を公衆ネットワークに流す必要がなくなり、プライバシー保護もより容易になる。

上記は入力のみに着目したサービスであるが、出力も考慮するとガイダンスシステムにも適用できる。例えばガイダンス用装置（方向を示すランプや音声情報を流すスピーカー）が設置された環境を想定すると、ユーザが持つ MN マネージャの移動に応じて、ユーザの要求条件に応じた情報をガイダンス用装置から出力することが可能となる。

4. 必要となる技術

提案システムの実行環境の実現に必要な技術について述べる。まず、動的な機能変更をサポートする M ノード用制御ソフトウェアが必要となる。これを「マイクロ OS」(以降、MINOS)と呼ぶ。もう一つは、MN グループを自律的に構築し、所属する M ノードを連携させ、サービスを動的に構成する方式である。

MINOS に対する要求条件として、組み込み用リアルタイム OS が持つようなタスク制御、メモリ管理、割込管理等の最低限のプログラム実行制御機能（カーネル）が必要である。これに加えて、M ネットワーク用通信機能やソフトウェア更新機能も必要となる。M ノードは十分なリソースを持たないので、プログラム実行制御については、仮想記憶を用いずに実メモリ空間で実現することとし、カーネルと入出力情報処理プログラムは一体化して動作する。なお、本稿では主に動的ソフトウェア更新機能について考察する。図 2 に MINOS の構成の概要について示す。

4.1. 動的ソフトウェア更新

MINOS が提供するソフトウェア更新機能について以下に述べる。なお更新には、(新規)追加も含まれるものとする。

(1) 基本方式

更新対象は、デバイスドライバ、プロトコル処理、

入出力情報処理プログラムである。タスクのスケジューラ自体も含むカーネル部の置き換えは今後の検討課題である。

MN マネージャがサービスに対する要求を解析し、M ノードの構成情報から、ソフトウェアを更新する M ノードと更新すべきソフトウェア、すなわち MSC を決定する。MSC の実体と MSC のバージョン情報は MNG マネージャが管理する。

M ノードのソフトウェアの実行は、タスクにより実現されるサーバと、テーブル形式による駆動を前提とする。ソフトウェアの更新は、テーブルの書き換えおよびタスクの再起動により実現する。

サービス実行中に、サービスを中断せずにソフトウェアを更新できること[10, 11]が望ましいが、MN グループを利用するユーザは限定的であり、サービスの動的構築に主眼をおくため、本稿ではサービス実行時の、ソフトウェア更新は対象としない。MN グループの適用範囲の拡大のため、実行時の更新についても検討を進める。

(2) 回復処理

M ノード上では複数の MSC が混在する。これらのバージョンの差違により、動作不良を起こす可能性が考えられる。このため、ソフトウェア更新後に不具合が生じた際に元の状態に復帰させる必要がある。MINOS は、MSC のバージョン情報と M ノード構成情報を用いて、ソフトウェア更新後も更新前の MSC と状態を保存し、元の状態に随時復帰される機能を提供する。

(3) 更新のスケジューリング

前述のように、M ネットワークは無線アドホックネットワークを想定している。M ノード間の通信速度は必ずしも大きくない。無線の出力不足によりマルチホップで通信する場合もある。また、各ノードで処理できるプロトコルの差違により、MNG マネージャから、対象となる M ノードに MSC を直接転送することがで

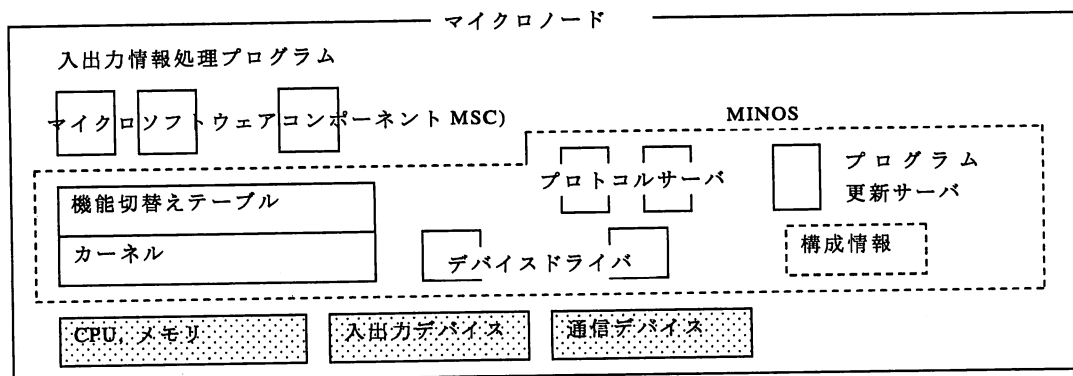


図2. MINOS の構成概要

きない場合がある。

したがって、MSC の転送をまとめて実施できない場合は、転送処理を分割し、ネットワークの通信速度に応じて転送処理をスケジューリングすることにより効率的に更新を行うことが必要となる。これは MNG マネージャと MN マネージャ間で、M ノードの構成情報（更新状態に関する情報を含む）と M ネットワークのシステム情報を相互にやりとりしながらスケジューリングを行う。

(4) セキュリティ

M ノードは十分なリソースを持たないため、汎用 OS のようにセキュリティ機能をもつことはできない。セキュリティを保証するための機能は、MNG マネージャと MN マネージャが実施する。例えば、これらのマネージャは、M ノードの構成情報から M ノードに転送して実行可能な MSC の可能性を判定する。これにより、M ノードのメモリに格納できない MSC や、競合関係にある MSC（同時に実行できない）を組み込むことを防止する。

4.2. マイクロノードの連携

M ノードを連携させてサービスを提供するためには、M ノードを識別するためアドレス、サービス構成の要求条件を満たす M ノードの発見、M ノード間の通信と制御、等の機構が必要となる。以下にそれらについて示す。

(1) M ノードのアドレス

M ノードを識別するアドレスは、MN マネージャが割当・管理する。M ネットワークには固有のアドレスが付与されるものとする。MN マネージャは、M ノードが属する M ネットワーク内で一意に定まるアドレスを付与する。一度付与されたアドレスは M ノードが記憶し、別の M ネットワークに移動しない限り変更しない。

M ネットワークに新たに加わった MN マネージャは、可能な通信媒体とプロトコルをすべて用いて、M ノードに対して応答する要求をブロードキャストする。M ノードは、それに対して、応答メッセージを返すものとする。応答メッセージには、属している M ネットワークの番号とノード番号が含まれる。

(2) M ノードの検出

M ノードは、それぞれ構成情報を持ち、それを適切なタイミングで放送する。タイミングは、M ノードが使用可能な電力等に依存して決められる。MN マネージャが M ノードから送られる構成情報を受信して自身にキャッシュとして持つ。本稿では、M ノードは同時に複数のマイクロネットワークに属さないものとする。同時に複数のマイクロネットワークに属する場合の問題点の明確化については今後の課題である。

(3) M ノード間の通信と制御

提案システムでは、M ノード間通信方式として、隣接ノード間の通信方法とパケット形式だけを定める。ルーティング等の機能は、既存のプロトコル[8]の機能を流用する。提案システムの機能の階層性に対応したプロトコルが必要になる。これらについて以下に述べる。

M ノード間のプロトコルとして必要な機能は、MN マネージャへのルーティングが必要である。これは既存プロトコルを適用する。M ノードと MN マネージャ間のプロトコルに必要な機能は、構成管理情報の通知、サービス構成に必要な連携すべき M ノードの情報を通知、連携に必要なインターフェースの通知等である。MN マネージャ間のプロトコルの機能として必要なものは、構成管理情報の引き継ぎ等である。MN マネージャと MNG マネージャのプロトコルの機能として必要なものは、M ネットワークの構成管理情報の引き継ぎ等である。

4.3. サービス構成

(1) 構成情報

ソフトウェア更新あるいはサービスの動的な構成のために、M ノードの構成情報が必要となる。サービス構成に必要な構成情報は、M ノードが提供する機能と、機能が扱う入出力種別に分類される。機能はさらに、入力、出力、変換に分類される。入出力種別としては、センサ入力の物理情報、ユーザの入力するテキスト情報、画面による選択枝の情報等がある。また M ノードに固有の属性もある。

具体例としては、性能（CPU 速度、メモリ量）、メモリマップ情報、空きメモリ情報、利用可能なデバイスの機能、M ノードに格納されているソフトウェアの種別（すなわちどのような機能を提供できるか）、等がある。また利用可能な通信メディアや、通信可能なプロトコルに関する情報も含まれる。

また M ネットワークのシステム構成情報としては、利用可能なプロトコルの背雨別、データ転送時の遅延時間がある。これらの情報は MN マネージャが管理し、MNG マネージャにも送られる。

(2) M ノードの組み合わせ

提案システムでは、複数の M ノードで一つのサービス（機能）を実現する。MN マネージャ自身がキャッシュしている M ノードの構成情報を用いて、要求されるサービスを実現するために必要な M ノードを発見する。

必要な M ノードが発見されると、これらの M ノード間の関係とインターフェースを決定し、各ノードにそれらの情報を伝える。各 M ノードはこれらの情報に基づいて連携動作を開始する。

本稿では、一つの M ネットワークでサービスを構成するとしているが、M ネットワークをまたがるサービスを提供することで、さらに高度なサービスの構成が期待できる。MN マネージャは、自身が管理する M ネットワークに属する M ノードの中で、他の M ネットワークで利用可能なものを識別する。他 M ネットワークの MN マネージャからの M ノード利用の要求があった場合は、利用可能な M ノードの構成情報を伝える。

(3) 自律分散制御

要求条件により合致したサービス構成が迅速に行うためには、M ノードを自律分散方式で知的にかつ協調的に自己制御する技術が求められる。これらの技術は、既にエージェント技術として多くの提案がなされている[12]。本システムではこれらの技術のうち適切なものを応用していくことを想定しているが、M ノードではリソースが限られるため、既提案の機能を縮退させる必要がある。

5. まとめ

ユビキタスコンピューティング環境において、十分な機能を持たないノードを動的に組み合わせ、新たなサービスを構成するために必要なソフトウェアシステムについて述べた。提案システムの構成要素が実現しなければならぬ機能要求条件を明らかにし、これらの条件を満たすために必要な技術として、動的ソフトウェア更新、ノードの連携、サービスの動的構成について考察した。

今後、オープンソースとして公開されているカーネルをベースに機能追加・変更を行うことで、MINOS の開発を進める。本ソフトウェアの実行環境をすべて実デバイス上で構築するのは、それぞれのデバイス向け開発環境の構築、対応したソフトウェアの開発等、稼働が極めて大きくなる。このため仮想ハードウェア(ノード)を擬似するソフトウェアを用いることにより、MINOS の開発と動作検証を効率的に進めていく予定である。

文 献

- [1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," ACM SIGOPS Operating Systems Review, Vol. 34, No. 5, pp.93 - 104, Dec. 2000
- [2] 安藤 繁, 田村 陽介, 戸辺 義人, 南 正輝 編著, センサネットワーク技術, 東京電機大学出版局, 2005
- [3] UPnP, <http://www.upnp.org/>
- [4] Jini Technology Architectural Overview, <http://www.sun.com/software/jini/whitepapers/architecture.html>
- [5] 板生 知子, 松尾 真人, "適応型ネットワークサービス環境 DANSE," 信学論(B), vol. J82-B, no. 5, pp. 730-739, May 1999.
- [6] 南 正輝, 杉田 馨, 森川 博之, 青山 友紀, "ユビキタス環境に向けたインターネットアプリケーションプラットフォーム," 信学論(B), Vol.J85-B No.12, pp.2313-2330, Dec. 2002.
- [7] N. Kawaguchi, "VPcogma: A Light-Weight Cooperative Middleware for Ubiquitous Embedded Devices, International Workshop on Software Architectures for Self-Organization (SASO '05) with Pervasive 2005.
- [8] C. K., Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall, 2001.
- [9] 井上知洋, 中村隆幸, 中村元紀, "外部センサーを用いた行動履歴の個人蓄積システム," 情処研報, 2003-GN-051, pp.13-18, Mar. 2004.
- [10] K. Koyanagi, H. Sunaga, T. Yamada, and H. Ikeda, "Applicability Evaluation of Service Feature Enhancement Using Plug-in Modification Technique," IEICE Trans., Vol.E81-B, No.1, pp.58-65, Jan. 1998.
- [11] T. Yamada, H. Sunaga, S. Tanaka, S. Sharaisi, K. Koyanagi, "Evaluation of Partial File Modification for Java-Based Realtime Communication Systems," IEICE Trans., Vol.E88-B No.10 pp.3982-3991, Oct. 2005.
- [12] 西田 豊明, 木下 哲男., 北村 泰彦, 間瀬 健二, エージェント工学, オーム社, 2002