

重安哲也[†] 三浦賢司^{††} 浦上美佐子^{†††} 松野浩嗣^{††}
[†] 山口大学大学院理工学研究科 ^{††} 山口大学理学部 ^{†††} 大島商船高等専門学校

1 はじめに

近年、制御機器等にプロセッサを組み込んだ機器の高機能化が進み、さらにネットワーク接続機能を備えるようになってきている。これらは、ネットワークを介して処理を行うことでデータの集中管理を行うためである。組み込み機器に限らず、現在の分散システムの通信は、リモートプロシージャコール(以下、RPC)を基本としている。RPCでは、ネットワークに接続されている各端末に対して処理を行う場合、常にネットワーク回線の確保が求められる。また、RPCベースの通信ネットワークでは、呼出中のプログラムは返値を受け取るまで処理が停止し、分散システムの持つ並列性を十分に生かすことができない。

一方、モバイルエージェント技術というものがある。モバイルエージェントとは、コンピュータ間の自律的移動能力をもつプログラムであり、移動先においても計算機リソースを利用することができる[1]。モバイルエージェントシステムは、ネットワークを介した処理を行うため、多くの場合ネットワーク機能が高く評価されているJavaを使用して設計される。

我々は、モバイルエージェントの概念をJavaを組み込んだ機器(以下、Java組み込み機器)のネットワークに採用することで、現在の一点集中のネットワークシステムよりも柔軟で、利用効率の良い処理分散システムが実現できると考えた。本稿では、Java組み込み機器ネットワークを考察し、モバイルエージェント技術を用いることの有効性を説明する。そして、今回Java組み込み機器のモデルとしてTINIを取り上げて開発したTINI用のモバイルエージェント(TINI Agent)の設計と実装について述べる。

2 Java 組み込み機器ネットワーク

我々は、Java組み込み機器ネットワークを構築するにあたり、現在の組み込み機器としての機能を損なわないために以下の3つの項目に留意した。

- 1 高リアルタイム性
制御用システムであるため、リアルタイムに処理を行えなければならない
- 2 ネットワーク協調性
効率的にネットワーク資源を利用できなければならない
- 3 低コスト・コンパクト
ネットワーク機能の追加によるコストの上昇とサイズ増加は極力避けなければならない

Design and Implementation of Mobile Agent System for Embedded Java Processor

Tetsuya Shigeyasu[†], Kenji Miura^{††}, Misako Urakami^{†††}, and Hiroshi Matsuno^{†††}

Graduate School of Science and Engineering, Yamaguchi University
^{††} Faculty of Science, Yamaguchi University
^{†††} Ohsima National College of Maritime Technology

このような、Java組み込み機器ネットワークシステムにモバイルエージェント技術を導入することで、

- a 移動先でローカル処理を実現
高リアルタイム性の維持
- b 負荷分散が可能
自律的に実行場所を選択でき、効率的にネットワーク資源を利用できる。(ネットワーク協調性の実現)
- c TCP/IP で実現可能
汎用のTCP/IPを利用して実現することができる(低コスト・コンパクト性の維持)

といった効果が期待できる。

3 情報家電用モバイルエージェントシステムの設計

3.1 ネットワークモデル

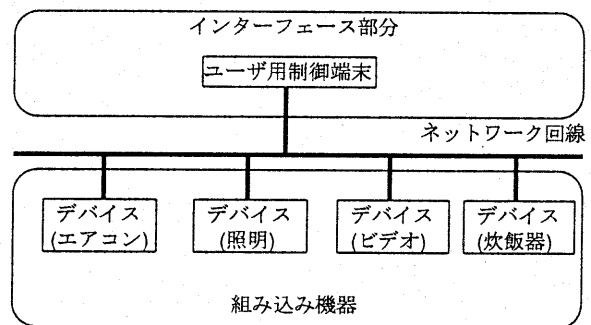


図1: ネットワークモデル

Java組み込み機器ネットワークのモデルとして、既に提案されているJTRONの適用例[2]を用いる(図1)。ネットワークに接続される構成要素は、ユーザからの要求を受け付けるインターフェース用端末と、そうでない個々の家電製品に搭載されている組み込み機器の2種類に大別できる。インターフェース用端末は、ネットワークに最低1つは存在することになり、組み込み機器は制御するデバイスの数だけ存在することになる。

3.2 モバイルエージェントのネットワークモデル

図1のネットワークモデルに対し、モバイルエージェントシステムを導入する(図2)。モバイルエージェントの実行環境は各端末に1つずつ搭載する。制御用端末から要求を受けたエージェントは、その要求にしたがって、組み込み機器を巡回し処理を行う。

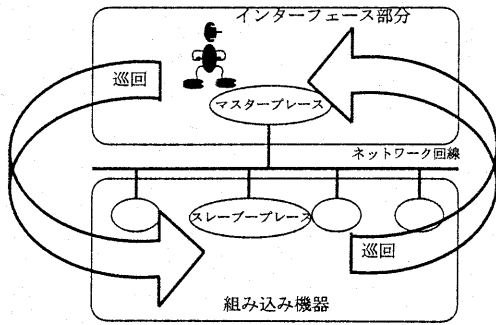


図 2: モバイルエージェントのネットワークモデル

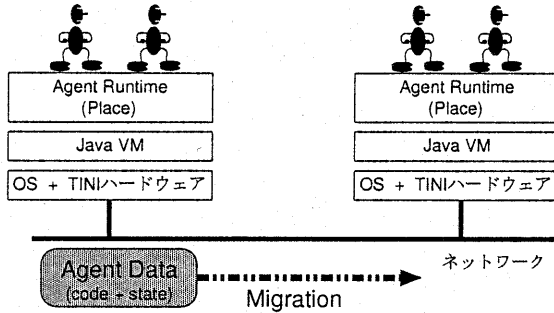


図 3: モバイルエージェントシステム

3.3 システムの設計

本研究では、Java 組み込み機器の TINI[3] をモデルとして設計する。TINI は SIMM サイズの非常にコンパクトなワンボードマイコンで、リアルタイム OS を搭載している。そして、1-Wire デバイスと呼ばれる電子デバイスを接続することで測定機器の制御用としての使用もできる。また、Ethernet もサポートしており、インターネットへの接続も可能である。

3.3.1 TINI 用モバイルエージェントシステム

TINIAgent(図 3) は OS と TINI のハードウェア上に搭載される JVM(Java Virtual Machine) の上で動作する。情報家電は計算機としてのリソースは限られたものになるため、TINIAgent も最低限度の以下 3 点に機能をしばって開発を行った。

- 1 エージェントの実行・管理
- 2 エージェントの転送
- 3 エージェント間通信

エージェントの実行・管理 エージェントはプレースにより普通の Java プログラムと同様に実行される。この際、エージェントが実行不能状態に陥ったとき(エージェントのフリーズ)はプレースによりそのエージェントは排除される。エージェントが実行できる機能は JVM におけるクラスローダを再定義することで使用できるクラスの制限を行い、悪意あるエージェントからシステムを守ることができる。

エージェントの転送 本システムでは、エージェントが遠隔地に転送される場合、エージェントの内部状態

とエージェントの実行コードを 1 つの AgentData と呼ぶオブジェクトにして転送を行う。

i エージェントの実行の中断
Java のシリアライゼーションを利用する。転送要求が発生したエージェントの内部状態は、バイトデータにシリアライズ(保存)される。このときシリアライズされるのは、Java の仕様によりエージェントの状態に限られる。

ii エージェントの実行再開
エージェントは転送後、再び実行再開するために内部状態がデシリアライズ(復元)される。本システムでは、Java のダイナミッククラスローディングを利用してエージェントを復元する。JVM 標準のクラスローダでは、ファイルシステムしか参照できないが、これをメモリを参照できるように再定義することで、転送された実行コードからクラスを生成し、エージェントを実行可能状態にする。

エージェント間通信 エージェントがネットワーク上に複数存在する場合、お互いに協調することで、より分散システムの並列性が増す。このため本システムでは、プレースを介してエージェント間通信が行えるよう設計した。

3.3.2 TINI 搭載 JVM の機能制限

TINI 搭載される JVM は一般の PC 上の JVM と異なり以下の制限がある。

- シリアライゼーションの未サポート
- ダイナミッククラスローディングの未サポート

しかし、TINI 用の次期 JVM リリースにより、上記 2 つの正式サポートが表明されている [4]。TINIAgent はその他の機能において TINI で動作することを確認しており、JVM リリースがあればすぐに本システムを TINI 上に搭載することが可能である。

4 おわりに

本稿では、Java 組み込み機器ネットワークにモバイルエージェントシステムを導入することを提案し、実際に TINI をモデルとして取り上げてシステムの設計を行った。今後は、TINI 用の JVM の次期リリースを待って TINIAgent を実際に TINI に搭載するとともに、TINI のネットワークを構成しての TINIAgent の評価を行う方針である。

参考文献

- [1] 佐藤一郎, モバイルエージェントの動向, 人工知能学会, vol.14, no.4, pp.598-605, 1999.
- [2] TRON プロジェクト, JTRON 仕様書, <http://tron.um.u-tokyo.ac.jp/TRON/JTRON/>
- [3] Dllas Semiconductor 社, Introducing TINI, <http://www.ibutton.com/TINI/>
- [4] Mailing list ARChives at PCC <http://marc.theaimsgroup.com/>