

検査機構を備えた機器サービスネットワークの構築に関する研究

西山 裕之 溝口 文雄
東京理科大学理工学部

1 はじめに

近年、急速なネットワークインフラの整備に伴いコンピュータやインターネットの利用者が拡大している。このような背景に伴い、情報機器やロボットの汎用化も進み、カメラやマイクなどのセンサをコンピュータもしくはインターネットに接続することでテレビ会議などのサービスを可能にする他、エンターテインメントを目的とした各種の家庭用ロボットが発売されている。このような機器を多数用いることで複合的なネットワークサービスも実現可能であるが、実装レベルにおいて各機器は個別に機能している他、各機器の接続状況や稼働状況は動的に変化するため、統合的な管理は困難である。

このような問題に対し、各機器を制御するソフトウェアのコンポーネント化や情報共有を行なうための接続アーキテクチャの設計等が行われているが [1]、規格外の制御ソフトウェアをサービスネットワークに組み込む場合、部分的な再構築を必要とする。また、機器間の統合方法として、各機器を制御するソフトウェア上に仲介用の通信ソフトウェアを用意する方法も存在する [2]。本方法の利点は、既に現存する制御ソフトウェアをそのまま利用してサービスネットワークを構築できることである。しかしながら、各制御ソフトウェアの起動や終了処理は基本的に手動で行う必要があるため、使用する機器数や動的に増減する機器（バッテリーを使用する機器等）の利用に応じて、物理的な管理が困難になる。

このような背景のもと、本研究では、計算機の内部状態を把握するメカニズムをサービスネットワークの構築に用いることで、各機器の制御ソフトウェアの状態に応じたサービスの実現や、制御ソフトウェアそのものの管理を可能にする。具体的には、各機器の稼働状況や通信状況を動的に把握するためのモニタリング機能を応用することで、機器のサービスネットワークの構築時における起動時の稼働検査を可能にする。また、利用中における動的な検証を可能にすることにより、頑健な統合管理を実現する。

2 計算機の内部モニタ

本研究では、計算機の内部をモニタリングする方法として、ネットワークセキュリティを目的として開発された次の2つのツール [3] を応用する。

2.1 プロセスモニタ

計算機内部で稼働しているプロセスを動的に認識するためのモニタツールである。モニタ対象として、プ

Design of the Development System for Device Service Network based on Checking Process

Hiroiyuki Nishiyama and Fumio Mizoguchi

Faculty of Science and Technology, Tokyo University of Science

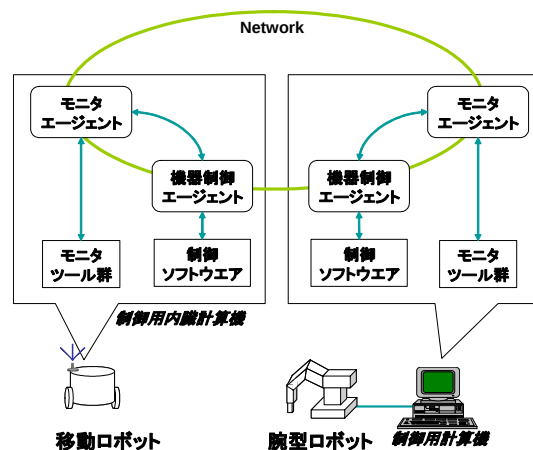


図 1: 各エージェントによる機器の統合

ロセス ID, 親プロセス ID, ユーザ情報, モジュール名等を把握可能である。また、各プロセスの CPU やメモリの使用量を把握する他、起動中のプロセスの削除や特定のモジュールを起動する機能も有する。本モニタを用いることにより、対象となる制御ソフトウェアが起動しているかを確認可能であるとともに、必要に応じて、制御ソフトウェアの起動や削除も可能である。

2.2 通信モニタ

ネットワークを介して計算機に送受信されるメッセージに対するパケットレベルの解析や、計算機内で開いているポートの認識を行うモニタツールである。モニタ対象として、開いているポート番号とプロトコルの他、送受信されたメッセージの送受信先の IP アドレスやポート番号を認識可能である。本モニタを用いることで、対象となる制御ソフトウェアの特定のポートが開いているかを事前に確認できる。また、送受信状況を確認することにより、制御ソフトウェアに対する指示が正常に行われているかの把握も行える。

3 機器間の統合

本研究では、機器間の統合方法として、図1のように既存の制御ソフトウェアの上に仲介用の通信ソフトウェア（以下機器制御エージェント）を用意するとともに、プロセスモニタと通信モニタから情報を収集するソフトウェア（以下、モニタエージェント）を用意することで、機器間の協調処理や稼働状況等の検証処理を実現する。

機器の制御ソフトウェアは、シリアルもしくは USB ケーブルを介して、接続されている機器を制御および情報収集する能力を持ち、ソケット通信を介して情報機器エージェントからの指示を受けるためのサーバと

しての機能を有するものとする（サーバとしての機能は、機器制御に関する最小限の機能を用意すればよいものとする）。

3.1 機器制御エージェント

機器制御エージェントは、機器サービスネットワークを構築するために他のエージェントとの通信機能を有する他に、対象となる制御ソフトウェアとの通信プロトコルを備えている。機器制御エージェントのモデルは次のようになる。

$cAgent(ID, Task, Negotiation, Rule, Info)$

- *ID* ... 識別コード
- *Task* ... 遂行すべきタスク
- *Negotiation* ... 他エージェントとの交渉および通信状態
- *Rule* ... 自分の状態，受信したメッセージ内容に応じた行動ルール
- *Info* ... 制御ソフトウェアに関する情報

機器制御エージェントは、他の機器制御エージェントやユーザからの依頼タスクを受けるエージェントである。また、協調作業が必要となる際、他のエージェントとの間でタスク依頼の交渉を行う [2]。

また、機器制御エージェントは、生成時にモニタエージェントの生成を行うことで、接続する制御ソフトウェアの存在確認や稼働状況の確認、そして、他の機器制御エージェント間の通信履歴の確認を依頼する。

3.2 モニタエージェント

モニタエージェントはプロセスモニタおよび通信モニタを用いることにより、生成元エージェントおよび依頼されたモジュール（プロセス）の稼働状況や通信状況を確認し報告する機能や、モジュールを起動および終了させる機能を持つ。モニタエージェントのモデルは次のようになる。

$mAgent(ID, Task, Info)$

- *ID* ... 識別コード
- *Task* ... 機器制御エージェントから依頼された監視タスク
- *Info* ... 監視対象となるモジュール（プロセス）やポート番号の情報

モニタエージェントは、機器制御エージェントから生成され、機器制御エージェントが接続する制御ソフトウェアの稼働状況や、機器制御エージェントの送受信状況を把握する。これにより、機器制御エージェントが制御ソフトウェアの利用を行う前に、モジュール名やポート番号の開放状況を把握することで、その存在確認を実施する。また、モジュール（プロセス）の稼働状況の把握による異常検知や、モジュールの終了および再起動も行う。これにより、機器サービスネットワーク構築時における制御ソフトウェアの起動ミスや異常時の再起動など、頑健なサービスシステムの実現を可能にする。その他、機器制御エージェントが他のエージェントと通信する場合、その通信状況を監視する。本機能は、機器サービスネットワークの利用制限や不正利用の検知を可能にする。

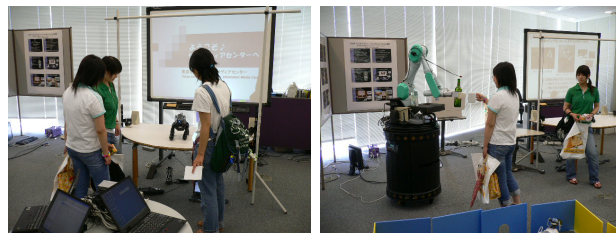


図 2: オープンキャンパスにおけるサービスの様子。ペットロボットによる説明サービスの様子（左図）とペットロボットの説明を伴うドリンクサービスの様子（右図）。

4 実装実験

本研究では、機器制御エージェント群およびモニタエージェント群を用いることで、5 台のロボット、カメラやマイク等の 4 つのセンサ系、音声発話のためのスピーカーやプリンタなど、計 20 種類の機器（およびサーバマシン）からなるサービスシステムを実装した。本サービスシステムは、本大学の見学者を対象とした移動ロボットと腕型ロボットによるドリンクサービス、見学時の写真撮影サービス、写真の印刷および配達サービスからなり、見学者は、小型ペットロボットからの説明サービスを受けながら、これらの一連のサービスを受けることができる。

本システムで使用した計 20 種類の制御ソフトウェアは、各サービスシステムの開発者により、異なる時期に開発されたものであり、制御するための通信プロトコルのみを備えている。本実装では、各制御ソフトウェアの改良や再構築を行うことなく、機器制御エージェントとモニタエージェント間の協調により、全サービスシステムの統合を実現した。本システムは、年間数回の見学会の他、図 2 のオープンキャンパスでは、1 日に 30 回以上のデモンストレーションを実現している。

5 おわりに

本研究では、複数のロボットやセンサ系からなるサービスネットワークを構築するために、各機器の稼働状況や通信状況を動的に把握するためのモニタリング機構を組み込むことにより、各機器の制御ソフトウェアの稼働検査を可能にした。また、サービス実施中における動的な検証を可能にすることにより、頑健なシステムの構築を実現した。

参考文献

- [1] 松坂 要佐，於久 健太郎，小林 哲則：多機能ロボット開発のための情報共有アーキテクチャの設計と実装，電子情報通信学会論文誌 VOL.J86-DI, No.5, pp. 318-329, 2003.
- [2] F. Mizoguchi, H. Nishiyama, H. Ohwada and H. Hiraishi: Smart Office Robot Collaboration based on Multi-agent Programming, *Artificial Intelligence*, Vol.114, pp.57-94, 1999.
- [3] H. Nishiyama and F. Mizoguchi: Design and Implementation of Security System Based on Immune System, *Software Security - Theories and Systems Lecture Notes in Computer Science*, No.2609, pp. 234-248, 2003.