Javaアプレットを用いたネットワーク型ロボットインタフェース

平松 薫[†] 森 啓[†,☆]
納谷 太[†] 大里延康[†,☆☆]

本論文では，Webブラウザ上のJavaアプレットをユーザインタフェースとし，ネットワーク経由でのロボット操作やセンサ状態の表示を可能にする仮想ロボットターミナルについて論ずる．仮想ロボットターミナルは，ロボットやセンサなどのハードウェア，ハードウェアとネットワークをブリッジするデーモンとユーザインタフェースの3層で構成する．既存のロボット操作環境は，ハードウェアへの依存が強く，拡張が困難なものがほとんどであるのに対し，仮想ロボットターミナルではJavaアプレットを利用することで開発，拡張を容易にし，Webブラウザを利用した可搬性の高いロボット操作環境を実現する．また，Javaアプレットがソケットを利用して，ハードウェアを制御するデーモンと直接通信することで，CGIプログラムなどによる中継を省き，通信の効率化を図っている．仮想ロボットターミナルの有効性を評価するために，データグローブ表示システムとロボットアーム操作システムを試作したところ，効率的な通信によりインタラクティブな状態表示が可能なことと，Javaアプレットにより自律性を持ったユーザインタフェースの構築が可能なことを確認できた．

A Robot Interface Using Java Applets

KAORU HIRAMATSU,[†] AKIRA MORI,[†,☆] FOTOSHI NAYA[†]
and NOBUYASU OSATO[†,☆☆]

This paper describes a network robot interface system called Virtual Robot Terminal using Java applets. This interface system consists of hardware modules such as robots and sensors and software modules such as daemons and Java applets. Java applets used in a Web browser act as hardware module user interfaces. The advantage of using Java applets is their portability. Java applets also assist system development. The daemons bridge hardware modules and the network. As a result, Java applets can access to robots and sensors. In order to improve communication throughput, daemons and Java applets interact directly with each other via TCP/IP sockets. This paper also describes and evaluates two experimental systems: a Data-glove visualizing system and a robot arm control system. The systems demonstrate capability for interactive robot operation systems. They also indicate that Java applets can construct an intelligent user interface system.

1. はじめに

近年，コンピュータネットワークは計算機や端末を相互に接続し，情報資源を共有する手段として，広く利用されるようになった．これをロボット技術と統合することは，ロボットに新たな機能や利用法を与える可能性を持つ[1]と同時に，ネットワーク[☆☆☆]と接続したコンピュータに対して，ロボットやセンサなどを通して実世界のオブジェクトへアクセスする手段を提供できることから，ネットワーク上で扱える情報量を飛躍的に増大させる可能性ももたらす．

そこで本論文では，ネットワーク上で情報を共有する手段の中で，特に広く普及したWorld Wide Web（WWW）に注目し，WWWとロボット技術の統合を図る．この統合のためのアーキテクチャを，仮想ロボットターミナルと呼ぶ．仮想ロボットターミナルでは，実世界のオブジェクトをWWW上の情報の一部として仮想的にリンクさせ，Webブラウザ上で動作するJavaアプレットから選択利用する．Javaアプレット

† NTTコミュニケーション科学研究所
NTT Communication Science Laboratories
☆ 現在，NTT関西法人営業本部
Presently with NTT Kansai Business Communication Headquarters
☆☆ 現在，日本大学工学部情報工学科
Presently with Department of Computer Science, College of Engineering, Nihon University

☆☆☆ なお，本論文でいうネットワークとは，コンピュータを相互に接続して情報資源を共有し相互利用するコンピュータネットワークのことであり，特に一定の区域内で利用するローカルエリアネットワークを指す．

はデーモンを経由して，ロボットやセンサなどのハードウェアにアクセスする．デーモンはハードウェアとネットワークの間をブリッジし，Javaアプレットと直接通信することで，効率的に動作する操作環境を実現する．また，Javaアプレットを利用することで，端末の種類によらない操作環境を実現する．

以下，ロボットシステムにおける操作環境の現状について検討し，それに基づいた仮想ロボットターミナルの設計とアーキテクチャの有効性を確かめるために試作したシステムについて論じる．

2. 背　景

現在普及しているロボットは，ソフトウェアとハードウェアの密接な関係によって統合され，全体が1つのシステムとして動作するものが多い．

たとえば，市販されている工業用ロボットでは，ティーチペンダントなどシステムに付属した操作端末を利用するのが一般的であり，ティーチペンダント上でロボットの動作をプログラミングし，その動作を繰り返し行わせる．このような操作端末はシステムごとに特化したものであり，それ自体がシステムの一部として組込まれているため，端末や操作方法の変更は容易ではない．

また，研究開発の段階ではあるが，遠隔手術環境[2),3)]に見られるように，バーチャルリアリティの技術を応用し，データグローブやヘッドマウントディスプレイなど，多くの入出力機器を利用することによって，直感的で使いやすい操作環境を実現しているシステムもある．しかし，これらのシステムも，多くの入出力機器を統合し，特定のソフトウェアを利用して操作を行うため，その操作方法の変更やシステムの拡張は困難である．

これらのシステムでは，ロボットと操作環境が一体のものであり，操作端末もその一部として開発されているため，ロボットと入出力機器の接続関係が固定的で，システムに特化したソフトウェアで全体を操作する必要がある．このため，システム構成の変更が難しく，ソフトウェアの交換にも多くの手続きが必要となり，操作方法の改良やシステムの拡張は非常に困難なものになっている．

これに対しWWWを利用したシステムでは，端末上のソフトウェアをWebサーバからダウンロードして利用するため，操作方法の変更のほとんどはWebサーバ上で行うことができる．

たとえば，Desktop Teleoperation[4)]や群ロボットの遠隔操作[5)]では，不特定多数のユーザが利用可能なWWW上に操作環境を実現し，ロボットシステムと操作端末の分離を図っている．これらのシステムでは，Webブラウザを操作端末として利用し，ネットワーク経由でデータの送受信を行い，ロボットを遠隔操作する．

しかし，WWWを利用したロボットシステムの多くは，一体型のシステムと比較すると反応が遅く，バッチ処理的なロボット操作しか実現されていない．したがって，WWWを利用したシステムの可変性と，一体型システムの反応性の良さを両立することが課題となる．

3. 仮想ロボットターミナル

前章で述べた課題を解決するための一手法として，本論文では3層構造を導入した仮想ロボットターミナルを提案し，WWWとロボット技術の統合を図る．仮想ロボットターミナルでは，Webブラウザ上で動作するJavaアプレットをユーザインタフェースとして利用することで，インタラクティブな操作環境をコンピュータネットワーク上に実現する．

JavaアプレットをWWWのリンク中に複数用意することで，Webブラウザ上で操作方法の選択が可能となる．たとえば，ロボットシステムの試作段階において，新しく開発した操作方法を複数切り替えて比較したり，熟練度に応じた操作環境を複数用意し，作業者に応じて切り替えたりできる．また，必要に応じてJavaアプレットをダウンロードし，端末上で利用するため，操作環境の管理がWebサーバホスト上で一括してできる利点もある．

Javaアプレットをユーザインタフェースに採用した理由は，直感的なロボット操作環境を実現する表現能力だけではなく，端末の種類によらず動作する可搬性と，通信などを自律的に行う能動的な処理能力を持っているからである．Javaアプレットは，多くの汎用Webブラウザ上で動作可能である．したがって，これまでのシステムのように専用の入出力機器からだけでなく，ネットワークに接続した携帯型ペンコンピュータなどからも，ロボットの操作が可能となる．またJavaアプレットは，能動的な通信が可能なので，反応の遅いCGIプログラム[6)]を経由せずに，ソケットを利用して直接通信することができる．そして，受動的にデータを表示するだけでなく，自律的なデータ処理が可能なことから，システムを構成する各モジュール間の通信速度とJavaアプレットの動作速度を考慮したうえでの機能分散も行うことができる．

ロボットを構成するセンサやモータは，機能ごとに

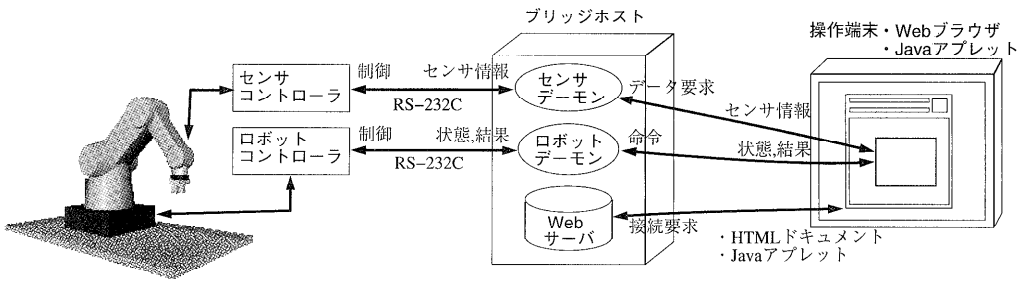


図1 仮想ロボットターミナルの構成例
Fig. 1 Diagram of virtual robot terminal.

仮想的に分散させた形でネットワーク上に公開し，それぞれ独立にアクセス可能にする．このような構成をとることで，モータやセンサなど各モジュールごとに調整を行う開発段階での利用と，複数のモジュールからの情報を統合して取り扱うロボット操作環境での利用の両方に対応することができる．

3.1 構　　成

仮想ロボットターミナルは，ロボットやセンサなどのハードウェアとその制御を行うコントローラ，ブリッジホスト（パソコンまたはワークステーション），およびネットワークに接続した操作端末で構成する（図1）．ブリッジホストは，各種コントローラとネットワークの双方と接続し，Webサーバとデーモンが常駐する．

デーモンは，各種コントローラの制御とコントローラとユーザインタフェースであるJavaアプレットの間のブリッジを行う．ブリッジホスト上のデーモンが，Javaアプレットとソケットを利用して直接接続・通信することで，作業者からの命令に対するレスポンスの改善を図り，インタラクティブな操作環境を実現する．

デーモンはC言語で実装し，SolarisやPOSIXのスレッドライブラリを利用してマルチスレッド化を図っている．デーモンプロセス内には，Javaアプレットとの接続ごとに生成される通信用スレッドと，ブリッジホストに接続したハードウェアと通信し，それを制御するための制御用スレッドが，並行して動作し，共有メモリを介して互いに通信しあうことでブリッジ機能を実現する．

デーモンは，Javaアプレットとの接続ごとに通信用スレッドをプロセス内に複数生成することで，マルチユーザ，マルチクライアント環境に対応する．デーモンとJavaアプレットの間の通信およびスレッド間通信に関する部分には，制御対象によらずほぼ共通のモジュールを利用することができる．また，制御用スレッドは，従来の操作環境で利用した通信・制御のモジュールを，マルチスレッドでの動作に対応するように拡張することで実現可能である．

操作端末には，ネットワークと接続可能な汎用のコンピュータを利用する．Javaアプレットは，この操作端末上で動作する汎用Webブラウザ上で利用する．ロボットの操作端末としては，作業者が一定の位置から操作する固定型端末と，必要に応じて作業者が移動しながら操作する携帯型端末の2種類が考えられるが，端末の処理能力や接続可能な入出力機器に差があるため，それぞれに適した操作方法を用意する必要がある．特に作業者が移動しながらロボットを操作する携帯型端末では，端末自体の処理能力などに制限があるため，その動作速度と利用する状況に応じて，情報の表示方法やロボットの操作方法を変更できることが望まれる．仮想ロボットターミナルでは，このような変更をダウンロードするJavaアプレットを変更するだけで実現できる．

3.2 動作メカニズム

仮想ロボットターミナルでは，操作端末上でWWW中の情報を閲覧する場合と同様に，Webブラウザ上でリンクをたどることで，操作環境を選択・利用することができる．この手順の詳細を以下に示す．

(1) ブリッジホスト上のWebサーバから，利用できる操作方法やデータ表示方法へのリンクのリスト（図2）を端末上のWebブラウザにダウンロードする．

(2) 提供されたリストから希望の方法を選択すると，Webサーバからその方法に対応したHTMLドキュメント（図3）とJavaアプレットがWebブラウザへ転送される．HTMLドキュメントには，利用するデーモンが常駐しているホスト名と通信に利用するポート番号が含まれる．

(3) Javaアプレットが起動すると，提供されたホスト名とポート番号に基づき，ブリッジホスト上のデーモンとソケットを利用して直接接続する．

(4) Javaアプレットはデーモンと通信し，デーモ

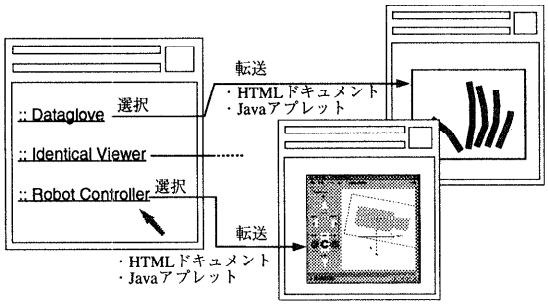


図2 初期画面からのリンク
Fig. 2 Initial screen.

```
<BODY>
<APPLET code="UserInterface.class"
                    width=300 height=240>
<PARAM name="host" value="bridge_host">
<PARAM name="port" value="6655">
</APPLET>
</BODY>
```

図3 HTMLドキュメントの一部
Fig. 3 HTML document sample.

ンから送られてくるロボットやセンサの状態を表示する．また，ユーザからの命令を受付け，デーモンへと送信する．

デーモンとJavaアプレットは，ソケットを利用して直接接続し，send/receiveの対を基本とした簡易なプロトコルを利用することで，通信に必要な手続きを最小限におさえ，効率化を図っている．この方式は，従来のCGIプログラムを利用したシステムと比較し，以下の利点を持つ．

- Webサーバに対する操作端末からのリクエストは，Javaアプレットをダウンロードするときのみで済む．したがって，本アーキテクチャではWebサーバに負荷が集中せず，Webサーバがボトルネックとなって発生する通信の遅延をおさえることができる．
- Javaアプレットとデーモンの間をパケットが直接行き来するので，通信スループットが高く，パケット中のデータ型の自由度も大きい．

4. 応用事例

仮想ロボットターミナルのアーキテクチャを評価するため，データグローブの状態を表示するシステムと，ロボットアームを操作するシステムを試作した．本章では，各システムの構成とその動作について論じる．

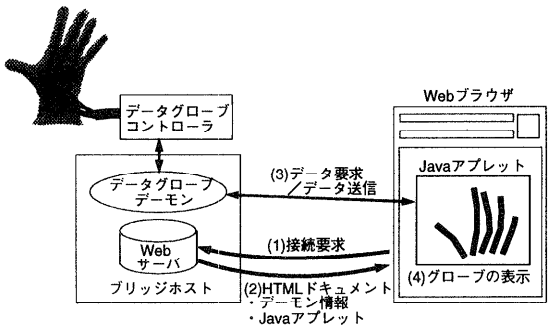


図4 データグローブ状態表示システムの構成
Fig. 4 Diagram of a dataglove visualizing system.

4.1 データグローブ表示システム

4.1.1 構　　成

このシステムでは，リアルタイムに変化するデータグローブの状態をJavaアプレット上に簡単なイメージとして表示する．このシステムで利用したデータグローブ☆は，手袋の各指に2個ずつ計10個のセンサで指の角度を検出する．表示システムの構成を図4に示す．

データグローブコントローラとブリッジホストの間は，RS-232Cでシリアル接続する．またブリッジホストと操作端末は，それぞれネットワークと接続する☆☆．

4.1.2 動作メカニズム

まず，ブリッジホスト上にデータグローブデーモンを常駐させる．データグローブデーモンは，データグローブコントローラの制御と，コントローラから連続的に出力されるグローブの各指の角度データのバッファリングを行う．データグローブデーモンは，Javaアプレットからのデータ要求に応じて，その時点でバッファリングしている最新のデータを返信する．

なお，ブリッジホストとコントローラの間のシリアル接続の速度は38,400 bpsである．データグローブデーモンは，コントローラからのデータ出力に従って，16.7 msごとにバッファの内容を更新する．

データグローブ表示システムは，3.2節で述べた手順で利用する．操作端末上のJavaアプレットがデータ要求パケットを送信すると，データグローブデーモンはバッファにある10個のセンサデータ（0から255の値）をバイト列にし，応答パケットとしてJavaアプレットに返信する．データを受け取ったJavaアプレットは，受信した10個のセンサデータから15個の

☆ VPL Research Inc.
☆☆ ブリッジホストにはSun Ultra30，操作端末には三菱電機AMiTY VPを利用し，10BASEのイーサネットを利用して双方を接続した．

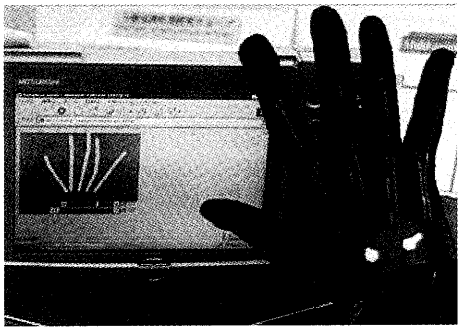


図5 ペンコンピュータ上で動作するJavaアプレット
Fig. 5 Java applet on a pen computer.

オブジェクトで構成する手のイメージを作成し，Webブラウザ上に表示する（図5）．

4.2 ロボットアーム操作システム

4.2.1 構　　成

ロボットアームは，ティーチペンダントから操作するのが一般的である．しかし，ティーチペンダントには組み込まれている機能が多く，ロボットを中心に固定された座標系で命令する必要もあったため，操作に不慣れなユーザにとって使い方が難しいものが多かった．そこで本節では，仮想ロボットターミナルのアーキテクチャに基づき，ロボットコントローラと位置センサをJavaアプレット上で統合し，作業者の位置に基づいた直感的操作のできるティーチペンダントを操作端末上に実現する．

本システムでは，位置センサを取りつけたペンコンピュータを操作端末として利用する．位置センサでロボットの持つ座標系の原点から見たペンコンピュータの位置を測定し，その位置情報から作業者のいる方向を求める．そしてこの方向を利用し，作業者からの見た目による上下左右の命令を，ロボットの持つ固定座標系における命令に変換する．この変換を実装したJavaアプレットを利用することで，作業者は任意の位置から見た上下左右でロボットアームを操作できるようになる．

たとえば，ロボットアームの側面（図6，位置A）に作業者がいる場合には，側面から見た上下左右で，また，正面（図6，位置B）からは正面から見た上下左右でロボットアームの操作が可能になる．この動作命令の変換は，Javaアプレットで行う．また，Javaアプレットに組み込む機能を絞ることによって，ユーザインタフェースを簡単なものにする．

仮想ロボットターミナルでは，この操作方法をロボットデーモンと位置センサデーモンの2種類のデーモンとJavaアプレットで実現する．ロボットアーム操作

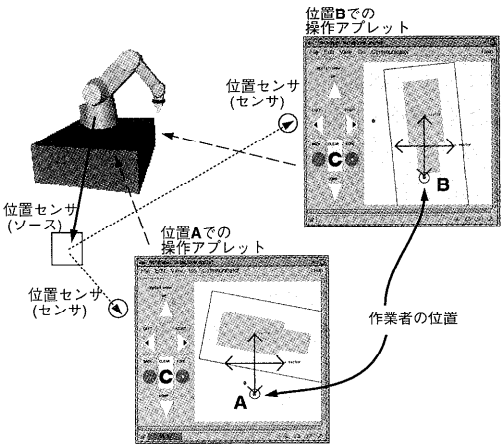


図6 位置を考慮したロボットアーム操作法
Fig. 6 A robot arm controlling method based on a user's viewpoint.

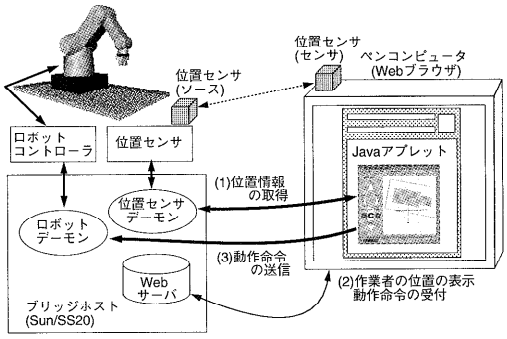


図7 ロボットアーム操作環境の構成
Fig. 7 Diagram of a robot arm operating environment.

システムの構成を図7に示す☆．

(1) ロボットデーモン

ロボットデーモンはロボットコントローラの制御と，コントローラとJavaアプレットの間のブリッジを行う．

(2) 位置センサデーモン

位置センサは，ソースで生成した磁界をセンサで測定することによって位置を測定する（図7）．位置センサの有効範囲は，ソースを中心とした半径1.5 mの範囲である．位置センサデーモンは，ロボットデーモンと同様に，位置センサを制御し，ソースから見たセンサの位置データのバッファリングを行う．そして，あらかじめ計測しておいたロボットアームの

☆ ロボットアームにはCRS ROBOTICS A465，位置センサにはPOLHEMUS ISOTRACKを利用，それ以外のブリッジホスト，操作端末およびネットワークは4.2節のシステムと同様のものを利用した．

座標原点から見たソースの位置と，計測した位置データからペンコンピュータの位置を求め，Javaアプレットに提供する．

4.2.2 動作メカニズム

ロボットアーム操作システムは，3.2節で述べた手順で利用する．ロボットコントローラと位置センサは，ブリッジホストとそれぞれシリアル接続する．シリアル接続の速度は，双方とも38,400 bpsである．

まず，本システムで利用する2種類のデーモンをブリッジホスト上に常駐させる．位置センサデーモンは，データグローブデーモンと同様に，位置センサからの連続的なデータをバッファリングし，Javaアプレットからの要求に応じて，最新のデータを返信する．ロボットデーモンは，Javaアプレットからの命令に従い，コントローラへ制御命令を送信する．

Javaアプレットは，Webブラウザに転送された後，ブリッジホストに常駐している2つのデーモンと接続する．本システムでは，位置情報に基づいた命令の変換をJavaアプレット上で行う．Javaアプレットは，まずソケットを通じて位置センサデーモンに対してデータ要求を行い，受け取ったデータに基づいて作業者の位置を示した地図を作成し，画面に表示する（図7(1)）．作業者は，画面上で上方から見た自分の位置と，その位置における見た目の上下左右の命令の方向を確認しながら，ロボットアームに命令を出すことができる（図6）．上下左右の命令は，Javaアプレットが位置情報に基づいてロボットの座標系における命令へと変換し，ロボットデーモンに送信する（図7(2)）．命令を受信したロボットデーモンが，その命令をロボットコントローラへ送信すると，命令に従ってロボットが動作する（図7(3)）．

5. 評　　価

本章では，提案したアーキテクチャに基づいた2つの応用事例の評価について論じる．仮想ロボットターミナルの動作速度の測定には，デーモンとJavaアプレットが1対1対応であるデータグローブ表示システムを利用し，デーモンの反応時間と，ユーザインタフェースであるJavaアプレットの表示更新周期を測定した．ロボットアーム操作システムに関しては，ロボットアーム付属のティーチペンダントとの操作感の比較と，本論文で実装した操作方法に関する評価を行った．

測定は，ペンコンピュータとブリッジホストを同一のハブに接続し，外部ネットワークから切り離して行った．Javaアプレットの開発にはJDK（Java Development Kit）1.0.2を利用，操作端末のOSにはWindows95を利用し，付属のWebブラウザInternetExplorer3.0上でJavaアプレットを動作させた．なお，外部のネットワークと接続した通常のネットワーク状態においても，ほぼ同様の速度で動作することを確認している．

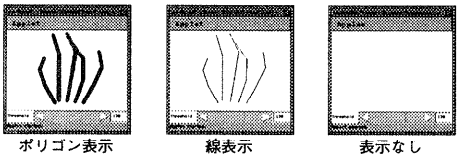


図8　表示オブジェクトの異なる3種類のJavaアプレット
Fig. 8　Three Java applets (Polygon, Line and no object).

5.1 デーモンの反応時間

デーモンの反応時間は，ブリッジホスト上の`tcpdump`コマンドを利用して測定を行った．デーモンが利用するポートの受信パケットと送信パケットのタイムスタンプを記録し，その時間差を反応時間とした．測定した1000回の反応時間の平均は0.135 msであった．ただし，この測定値はブリッジホスト上での反応時間であり，Javaアプレットから見た反応時間には，ネットワークによる遅延が追加されることになる．

5.2 Javaアプレットの表示更新周期

Javaアプレットの表示更新周期は，Javaアプレットが通信と表示を逐次的に繰り返すことを利用し，ブリッジホストに到着するデータ要求パケットの周期で測定した．測定には反応時間の測定と同様に，ブリッジホスト上の`tcpdump`コマンドを利用した．デーモンが受け取るデータ要求パケットのタイムスタンプを記録し，その間隔1000回分の平均をJavaアプレットの表示更新周期とした．なお，各Javaアプレットは横300 × 縦200のPanelに表示を行い，オフスクリーンを利用してダブルバッファリングを行っている．

まずこの測定では，表示するグラフィックオブジェクトの種類が，どの程度動作速度に影響するか調べるため，データグローブをポリゴンで表示するJavaアプレット（線データから六角形を作成しfillPolygonメソッドで表示），線で表示するJavaアプレット（線データをdrawLineメソッドで表示），表示はまったく行わずに通信とそのデータ処理のみを行うJavaアプレットの3種類を用意し，比較を行った（**図8**）．3種類のJavaアプレットの表示更新周期の測定結果を**図9**に示す．

ポリゴン表示のJavaアプレットは，表示更新周期が28.1 msであった．したがって，Javaアプレットを利用したデータ表示システムでも，30 Hz以上の表示

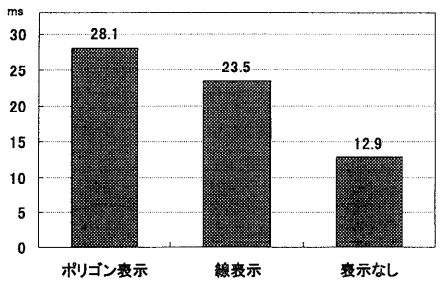


図9 Javaアプレットの表示更新周期（1）
Fig. 9 Refresh rate of the Java applets (1).

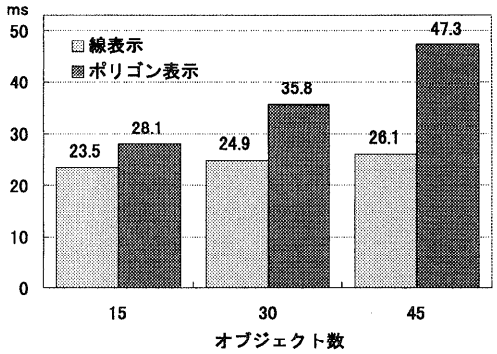


図10 Javaアプレットの表示更新周期（2）
Fig. 10 Refresh rate of the Java applets (2).

更新が可能なことが分かった．実際のペンコンピュータの液晶画面上でも，グローブの動きにほぼ追随したオブジェクトの表示をしており，リアルタイムでセンサデータを観察する目的では十分な速度といえる．

線表示のJavaアプレットの表示更新周期は23.5 msと，ポリゴン表示の場合より4.6 ms短くなった．また，表示をまったく行わずに通信とそのデータ処理のみを行うJavaアプレットでは，動作周期が12.9 msであった．オブジェクトを表示する2つのJavaアプレットにおいて，通信とそのデータ処理に必要な時間が同じと仮定すると，表示を行うJavaアプレットでは，その処理時間のうち約50%が表示処理に使われていたことになる．

次に，オブジェクト数が表示更新周期に与える影響を測定した．この測定では，基本のJavaアプレットが描画する15のオブジェクトを繰り返し描画させることで，表示するオブジェクト数を変化させた．測定には図9の測定と同様に，線表示のJavaアプレットとポリゴン表示のJavaアプレットを用意し，それぞれ1000回分の表示更新周期を測定し，その平均を計算した．

測定結果を図10に示す．表示オブジェクトの種類によらず，数が増加すると，ほぼ線形に表示更新周期も長くなった．また，オブジェクト数による表示更新周期の変化は，オブジェクトの種類によって影響されることも分かった．したがって，Javaアプレット上で操作環境を作成する際には，操作環境や状況に応じて表示オブジェクトを選択し，必要に応じてオブジェクト数を絞ることが，Javaアプレットの表示速度の改善するうえで重要となる．

5.3 ロボットアーム操作システム

ロボットアーム操作システムでは，作業者から見た上下左右でロボットの操作を可能にした．本システムでは，Javaアプレット上のボタンを押してから実際にロボットアームが動き出すまでに若干の遅延が感じられるケースもあったが，位置センサの有効範囲内の任意の位置から，上下左右の指示で違和感なくロボットアームの操作ができた．

また利用実験時に，この操作方法はロボットアームに物をつかませるなどの細かい作業には向かないとの指摘も受けている．この問題に関しては，複数の操作方法を用意し，それらを切り換えて利用することで解決できると考えている．今後は，本論文で実装した操作環境に加え，従来型のロボット中心の座標系を利用した操作環境や，センサ情報の表示に特化したJavaアプレットなどを作成し，操作環境の充実を図っていきたい．

6. 考　　察

6.1 アーキテクチャの有効性

仮想ロボットターミナルでは，ロボットアーム操作システムで示したように，Javaアプレットの開発のみで，新たな操作方法の開発が可能である．たとえば，ロボットアーム操作システムで，位置センサからの情報の利用方法を変更したい場合には，Javaアプレット内のアルゴリズムのみを変更すればよい．従来のシステムと比較し，ソフトウェアの変更およびその交換が容易なことから，より効率的なシステム開発が可能となり，システム開発に必要な時間，コストとも大幅に削減できると考えている．

ブリッジホスト上のデーモンは，ロボットやセンサなどのハードウェアに対応して存在し，それぞれ独立した通信ポートをネットワークに公開する．したがって，必要な機能に合わせてブリッジホスト上のデーモンを選択し，Javaアプレットを開発することができる．たとえば，位置センサデーモンのみに接続し，位置表示機能のみを持つJavaアプレットや，データグローブデーモンに接続し，データグローブからのロボット

アームを操作可能にするJavaアプレットなどが，接続するデーモンとJavaアプレットの変更だけで実現できる．

4章のシステムでは，ロボットとセンサを利用してシステムを構築したが，仮想ロボットターミナルには特に利用できるデバイスに関しての制限はない．本アーキテクチャを利用することで，既存のネットワークと安価なパソコンから，実世界のオブジェクトもWWW上のリンクの1つとしてアクセス可能になるので，従来の電子情報しか扱えなかったコンピュータネットワークで扱える情報の範囲を簡単に拡大することができると考えている．

ただし問題点として，(1) 独自に設定した通信ポートを利用するため，firewallを越える接続が必要な遠隔操作への拡張が難しい，(2) JavaアプレットがWebブラウザのセキュリティ方針を継承するため，Javaアプレットの転送元であるブリッジホスト以外にあるデーモンとの接続が制限される，(3) JDKオリジナルの通信クラスを利用するため，通信の複雑さに応じてプログラムの記述量が増加する，ことがあげられる．これらの問題点については，今後検討していく．

6.2 機能分散

仮想ロボットターミナルでは，デーモンとJavaアプレットに処理を分散させることで，効率的処理を行うロボット操作環境を実現することができる．分散システムとして仮想ロボットターミナルを見た場合，システムの性能にはJavaアプレットなどのモジュール単体の処理能力だけではなく，利用できるネットワークの帯域幅とモジュール間の負荷配分が大きく影響する．特に，ユーザインタフェースであるJavaアプレットは，単に受動的にメッセージを表示するだけでなく，様々なデータ処理も可能なので，処理前後でのデータサイズの変化とネットワークの帯域幅を考慮にいれた機能分散を行うことができ，より効率的に動作するシステムを構成することが可能になる．

従来のWWWベースのシステムでは，サーバ側で作成したサイズの大きな画像データをクライアントであるWebブラウザに転送し表示していたため，通信量が多く，レスポンスも悪いものになっていた．これに対しデータグローブ表示システムでは，デーモンがデータグローブコントローラの制御と出力データのバッファリングを行い，デーモンが整形したデータを基に，Javaアプレットが手のイメージを作成し表示する．このように仮想ロボットターミナルでは，イメージ作成などに必要なJavaアプレットの処理時間とネットワークの帯域幅のバランスをとった機能分散が可能であり，全体としてレスポンス良く動作するシステムの構築を可能にしている．

6.3 遅延対策

仮想ロボットターミナルでは，デーモンやJavaアプレットをはじめ，複数のモジュールが互いに通信しあいながら動作する．このようなシステムの場合には，特定の通信路の速度から受ける影響ができるだけ全体に及ばないように実装することが重要となる．このため4章のシステムでは，デーモンをマルチスレッド化し，データのバッファリングを行うことで，効率化を図っている．効率良くデータをバッファリングすることで，デーモン，Javaアプレット，ロボットやセンサが非同期に動作することが可能となり，特定の通信路が原因で発生するボトルネックを回避することができる．

ただし，ネットワーク経由で通信するシステムの性格上，デッドラインのある命令の実行を保証することはできない．しかし，応用事例の測定環境では，Javaアプレット上のイメージが止まるような状況は発生しなかったことから，トラフィックの少ないローカルネットを利用するのであれば，ネットワークによる遅延はほぼ無視できると考えている．

6.4 マルチユーザ対応

仮想ロボットターミナルは，データグローブ表示システムで利用した観測の側面と，ロボット操作システムで利用した操作の側面を持つ．観測の側面では，デーモンが複数のJavaアプレットからの接続に対応することで，複数の端末から同時に同じデータを利用できる．しかし，操作の側面では，実世界のオブジェクトを扱い，その操作に応じて状況が変化するため，ユーザ間で排他処理を行う必要がある．

Desktop Teleoperationでは，パスワード認証による排他制御を実装している．アクセスしている多数のユーザのうち，実際にロボットを操作できるのは1人だけにするというポリシーで排他処理を行い，操作可能なユーザは5分ごとに切り替わるようになっている．操作可能なユーザ以外は，その動作を観察するだけとなる．このシステムでは，ユーザインタフェースがWebブラウザ上に存在しても，情報を表示するだけの受動的なクライアントであるので，Webサーバ上で排他処理を容易に行うことができる．

これに対し仮想ロボットターミナルは，通信経路を一括して管理する機能がないため，同様の方法をとるのは困難であり，現在のところ，このような排他処理は実装されていない．

仮想ロボットターミナルで排他処理を行うために

は，デーモンのレベルで排他処理を実装する必要があり，複数のデーモン間で協調して処理を行う必要があると考えている．また，操作環境の利用効率をあげるためには，利用可能なユーザの時間切替えだけではなく，命令ごとの切替えや，スケジューリング機能も必要と考えており，今後の課題として検討していく予定である．

7. おわりに

本論文では，ネットワーク経由でロボット操作やセンサ状態の表示を可能にする，仮想ロボットターミナルについて議論した．Javaアプレットをユーザインタフェースとして利用し，ブリッジホスト上のデーモンを介してロボットやセンサにアクセスすることで，実世界のオブジェクトをネットワーク上から扱えるようにした．Javaアプレットを利用することで，操作環境の可搬性を高め，端末の種類によらない動作を可能にした．また，Webブラウザ上での効率的なデータ表示を実現し，柔軟な機能分散も可能にした．

今後の課題としては，操作環境における排他処理の実装と通信に関連するプログラムの記述性を高めることがあげられる．また，位置センサを利用した直感的ロボットアーム操作方法についても，改良および検証を行っていきたい．

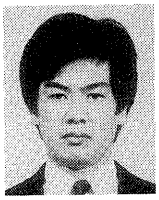
参考文献

1) 平井：ネットワークとテレロボティクス，日本ロボット学会誌，Vol.15, No.4, pp.516–519 (1997).
2) Mitsuishi, M., et al.: Development of an Inter-World Tele-Micro-Surgery System with Operational Environment Information Transmission Capability, *Proc. ICRA'95 Conference*, Nagoya, Japan, pp.3081–3088, IEEE (1995).
3) 大泉ほか：VRの現状と脳神経外科におけるVRシステムの展望，新医療，pp.42–45 (1995).
4) Goldberg, K., et al.: Desktop Teleoperation via the World Wide Web, *Proc. ICRA'95 Conference*, Nagoya, Japan, pp.654–659, IEEE (1995).
5) Suzuki, T., et al.: Telcoperation of Multiple Robots through the Internet, *Proc. IEEE Intl. W. Robot and Human Communication*, Nagoya, Japan, pp.84–89, IEEE (1996).
6) Orfari, R., et al.: *Client/Server Programming with Java & CORBA*, Wiley (1997). 並河ほか（訳）：Java & CORBA C/Sプログラミング，日経BP社 (1997).

(平成10年3月13日受付)
(平成10年10月2日採録)

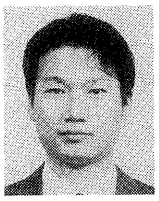
平松 薫（正会員）

昭和46年生．平成6年慶應義塾大学理工学部電気工学科卒業．平成8年同大学院理工学研究科計算機科学専攻修士課程修了．同年日本電信電話（株）入社．現在NTTコミュニケーション科学研究所に所属．コンピュータやロボット等の人工物のユーザインタフェース，デジタルシティに興味を持つ．日本認知科学会，人工知能学会各会員．



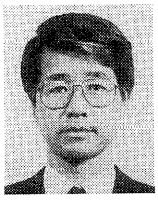
森 啓

昭和37年生．昭和61年東北大学工学部電子工学科卒業．昭和63年東北大学大学院工学研究科電子工学専攻博士前期課程修了．同年日本電信電話（株）入社．同研究所において，キャッシュメモリ管理プロトコル，マルチプロセッサ，分散オブジェクト指向に基づく制御システム等の研究開発に従事．平成10年よりNTT関西法人営業本部．電子情報通信学会，日本ロボット学会，日本生物物理学会各会員．



納谷 太

昭和45年生．平成4年慶応義塾大学理工学部電気工学科卒業．平成6年慶応義塾大学大学院理工学研究科計算機科学専攻修士課程修了．同年日本電信電話（株）に入社．以来，NTTコミュニケーション科学研究所にて，マルチエージェントシステム，分散協調型ロボット制御，ロボットビジョン等の研究開発に従事．現在に至る．人間と共存できるロボットシステムに興味を持つ．電子情報通信学会，日本ロボット学会，計測自動制御学会各会員．



大里 延康（正会員）

昭和26年生．昭和49年九州工業大学工学部電子工学科卒業．昭和51年東北大学大学院工学研究科電気及通信工学専攻博士前期課程修了．同年日本電信電話公社入社．同研究所において，Lispマシン，オブジェクト指向プログラミング，知能ロボット等の研究開発に従事．博士（工学，大阪大学）．平成10年より日本大学工学部情報工学科教授．分散協調型知能システムに興味を持つ．情報処理学会平成元年度論文賞受賞．電子情報通信学会，IEEE各会員．