

RCML 2.0: ネットワーク環境における遠隔ロボット操作システムの開発

関口 大陸*¹ 川上 直樹*¹ 舘 暲*¹

RCML 2.0: Development of a Networked Telerobotic System

Dairoku Sekiguchi*¹ Naoki Kawakami*¹ and Susumu Tachi*¹

Abstract – The concept of R-Cubed (Real-time Remote Robotics: R³) aims to provide a way to telexist anywhere in the world by controlling remote robots over the network. RCML (R-Cubed Manipulation Language) is considered to be a bottom-up approach of the R-Cubed concept. The design of an RCML system utilizes existing infrastructures and devices such as the Internet and PC and, users of the system will be able to use it easily and intuitively.

RCML is a language for describing the interface for controlling remote robots in an R-Cubed concept. This paper proposes a new design of the RCML system called RCML 2.0. In RCML 2.0, we introduced a language RXID 2.0 for defining Graphical User Interface (GUI), which is used for controlling the remote robot into the system. Both RCML 2.0 and RXID 2.0 are XML-based languages. By using XML, expandability and flexibility in implementation are introduced to the RCML system. RXID 2.0 has mechanism for a one-way link to an RCML data structure, and this mechanism provides for the complete separation of the control of the robot and the user interface. We also show the experimental implementation of the RCML 2.0 system which supports PDA and celler phone as a client in addition to ordinary desktop PC client.

Keywords : teleoperation, internet, network, Self-Descriptive Remote Memory Model, RCML, XML

1. はじめに

近年、ヒューマノイドやペットロボット、介護ロボット等の人間協調・共存型ロボットが人気を博している。その一因として、具体的な研究成果の発表や [1] [2], さまざまな新しい製品の発売などがあげられる。人間協調・共存型ロボットという名前自身が表しているように、ロボットが活躍するフィールドも、工場のラインなど、ロボットが旧来多く使用されてきた場所から、我々が普段生活している空間など、より身近な場所へシフトしつつある。

大学の研究者や通商産業省（現経済産業省）などによりまとめられたアールキューブ構想 [3] [4] も、近年見られるロボットのパーソナル化への流れの中で、ロボットを社会の中で広く利用できるようにするための一つのアプローチとして捉えることが出来る。アールキューブ (R³) の名前は、Real-time Remote Robotics (実時間遠隔制御ロボット技術) の頭文字からつけられており、R-Cubed と表記する。アールキューブ構想は、家庭やオフィス、公共の場所など、我々の生活する

社会の各所に、自由にアクセスできるロボットを配置し、電話をかけるかのような気軽さで、遠隔のロボットをあたかも自分の分身であるかのように扱う、すなわち遠隔にテレグジスト (Telexist) 可能とすることを目指す構想である。アールキューブ構想では、テレグジスタンス [5] (Telexistence) を中核となる技術として用いることにより、高臨場感操作を実現する。

アールキューブ構想では、これまで特別な設備が必要とされるハイエンドなシステムの研究が主として行われてきた [6] (図 1 上)。本研究では、既存のデバイスやネットワークインフラを生かしたローエンドなシステムを主な研究対象とする (図 1 下)。ローエンドのデバイスを生かした実装は、アールキューブ操作言語 (RCML: R-Cubed Manipulation Language) を中心とした RCML システムとして構成され、アールキューブ構想におけるボトムアップのアプローチとして位置づけられる。すでに存在する PC などの既存デバイスやインターネットなどの既存ネットワークインフラを活用することにより、誰もが、自由に、何時でも何処でも、気軽に使用できる、パーソナルなロボットネットワークシステムの構築を目指している。RCML システムは、アールキューブ構想におけるボトムアップ

*1: 東京大学大学院 情報理工学系研究科

*1: Graduate School of Information Science and Technology, The University of Tokyo

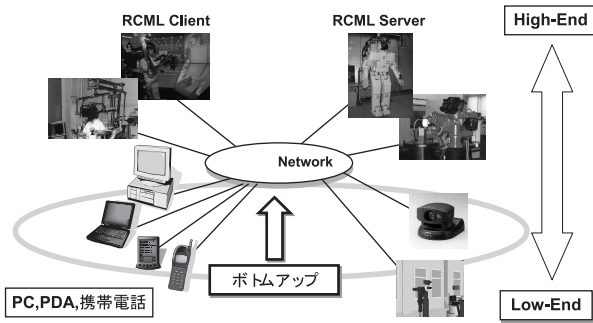


図1 アールキューブ構想
Fig.1 The R-Cubed concept

アプローチとして、今後、家庭内での使用が予想される遠隔監視カメラやペットロボットなど、さまざまなパーソナルロボットを主たる遠隔制御対象とする。

本論文では、今回新たに設計を行った RCML 2.0 システムに関して述べる。RCML 2.0 システムでは、特に、クライアント側のデバイスとして、PC に限らず PDA (Personal Digital Assistant) や携帯電話への対応を視野に入れ、システムが使用できる範囲の拡大を試みた。

2. RCML 1.0 システム

まず、RCML 2.0 システムの前身となった RCML 1.0 システムについて簡単に説明する。RCML 1.0 システムは、遠隔ロボットの記述言語である RCML 1.0 と通信プロトコル RCTP/1.0 (R-Cubed Transfer Protocol) からなる [7] [8] [9] [10]。RCML 1.0 は、VRML97 (Virtual Reality Modeling Language)[11] に対して遠隔ロボットの制御情報の記述に必要な仕様が拡張された言語である。RCML 1.0 の開発を通じて、幾つかの改良すべき点があることも明らかになった。一つは、VRML97 を基本とする言語仕様としたために、実装が基本的に既存の VRML ブラウザを Java を用いて拡張する方法に限られてしまい、システム実装の自由度およびパフォーマンスに制約が生じた。

その他に、制御情報とユーザインタフェース情報の記述を分けたほうが良いことも明らかになった。RCML 1.0 では、入力に使用する GUI の種類など、ユーザインタフェースに関する情報と、遠隔ロボットの制御に必要な情報、さらには VRML 本来のモデリング情報が一つのファイルに記述されている。遠隔ロボットの制御に必要な情報は、各ロボットに固有のものであるために、一度記述されると、それほど頻繁に変更されるものではない。しかし、ユーザインタフェースの記述は制御情報より頻繁に書き換えられる可能性が高い。事実、RCML 1.0 の実験で使用されたシステムは、遠隔のロボットを制御するインタフェースとして、入力がボタンもしくはスクロールバーなど、幾つかの異

なったデザインのインタフェースが用意されていた。RCML 1.0 の仕様では、複数のユーザインタフェースを一つのロボットに対して用意すると、ユーザインタフェース以外の制御情報などが複数ファイルに重複して存在することになり、効率が悪いだけでなく、ファイル管理が複雑になる。加えて、性質の異なる制御情報及びユーザインタフェース情報、モデリング情報の各記述が一つのファイルに混在するために、注意して記述しないとファイルの可読性が悪化する。

3. RCML 2.0 システム設計

3.1 設計指針

我々は、RCML 1.0 システムの開発で明らかになった改良すべき点などを踏まえ、RCML 2.0 システムの設計 [12] にあたって、以下に示す設計指針をまず設定した。

- 多様な使用形態への対応
- 遠隔ロボット制御に最適な設計
- フレームワークとして機能

多様な使用形態への対応により、デスクトップ上の PC だけではなく、PDA や携帯電話などの様々な携帯機器もクライアントして使用できるようにし、何時でも何処でも使えるシステムの実現を目指す。さらに、遠隔ロボットの制御が主目的のシステムであるため、遠隔ロボット制御に最適な設計が要求される。特に、通信レイテンシの低減など、通信のリアルタイム性に関する考慮を欠かすことが出来ない。RCML 2.0 システムは、前記 2 つの設計指針を満たした上で、ネットワーク環境における遠隔ロボット操作システム構築のためのフレームワークとして機能しなければならない。その場限りの実装ではなく、システム構築のためのフレームワークとして、様々なロボットやデバイスへの柔軟な対応、スケーラビリティやシステム間の相互接続性の確保などが必要とされる。

3.2 既存の実装方法との比較

ネットワークを経由した遠隔ロボット制御の試みは、これまでも多くなされてきている。ここで、過去に行われた代表的な実装方法を概観し、RCML 2.0 の設計指針を満たすにはどのような設計にすればよいか考察する。

これまでに発表されたネットワークを経由の遠隔ロボット制御システムの多くは、CGI による実装 [13] もしくは Java アプレットによる実装 [14] [15] [16] をとっている。CGI による実装は、ユーザがロボットに対して指示を与えるたびに、クライアント側の Web ブラウザでリロードが発生し、通常の Web ブラウザを使用している限りにおいては、連続的に指令を送信することは難しく、遠隔ロボット制御の観点から見ると最

適な設計がされているとは言い難い。一方, Java アプレットによる実装では, 連続的なデータの送受信や, CGI では実現が難しいインタラクティブなユーザインタフェースが可能などの利点がある。しかし, 実行速度及びメモリの使用量の観点から見ると, C/C++ 等で記述されたネイティブのアプリケーションと比べると不利である。

より汎用的で洗練されたシステムを構築する目的で, CORBA [17] (Common Object Request Broker Architecture) や DCOM [18] (Distributed Component Object Model) などの既存の分散オブジェクトフレームワークを用いた実装も試みられている [19] [20]。すでに確立された技術である既存のフレームワークを用いることにより, 汎用性や信頼性の確保などが期待される。しかし, プログラム時もしくは運用時にネットワークをシステムの表面から積極的に隠蔽する分散オブジェクトフレームワークは, ノード間の結合が疎で常にネットワークの存在を意識しなければならないインターネットとの不適合性が指摘され, SOAP [21] (Simple Object Access Protocol) のようなインターネットの特性に合わせたメッセージ交換のフレームワークが近年提案されてきている。SOAP は, インターネットへの適応という観点からは非常に成功しているが, ノード間で単発的に発生するメッセージ交換に主眼をおいているため, 遠隔ロボット制御に必要とされる連続的なデータの送受信などはあまり考慮されていない。

このように, 既存の実装方法の延長線上で考える限りでは, いずれも一長一短があり, 設計指針を満たすシステムを構築することは困難である。そこで, 設計指針を満たすような新たな設計モデルから考えることにした。

3.3 Self-Descriptive Remote Memory モデル

RCML 2.0 システムの開発では, 設計指針を満たす新たな設計モデルとして Self-Descriptive Remote Memory モデル (以降, SDRM モデル) を提案し, SDRM モデルをもとにシステムの設計を行った。SDRM モデルは, イントロスペクションを実現する Self-Description (自己記述) とリモートメモリアccessモデルとの組み合わせからなる。リモートメモリアccessモデルとは, ネットワークを介してアクセス可能なメモリを構成する方法である (図 2)。リモートメモリアccessモデルでは, リモートのメモリへのアクセスは, リモートメモリに対する READ と WRITE の 2 種類の通信プリミティブですべて行われる。リモートメモリアccessモデルは, 分散共有メモリモデルとは異なりアクセス対象となる側のみにメモ

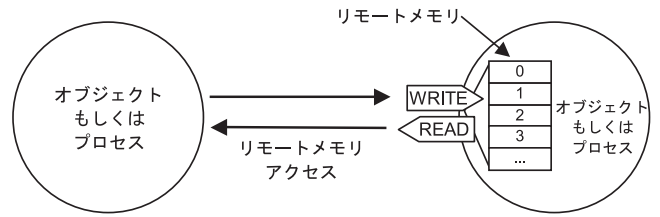


図 2 リモートメモリアccessモデル
Fig. 2 Remote Memory Access Model

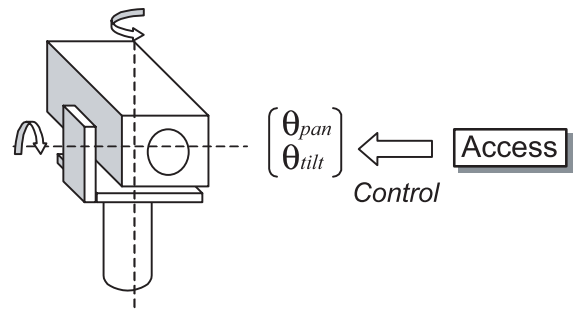


図 3 パン・チルトカメラによる例
Fig. 3 Example by Pan/Tilt Camera

リが存在するため, 複雑なプロトコルを用いてノード間のコンシステンスを保つ必要もない。このように, リモートメモリアccessモデルは通信方法が非常に単純化されており, 常にノード間でメッセージ交換は最低限可能であることが保証され, 疎結合なネットワークであるインターネットでの使用に十分対応可能な方法である。

さらに, SDRM モデルでは, リモートメモリに対するイントロスペクションを導入し, 相互接続性の向上を試みる。イントロスペクションとは, 自らを省みる性質, すなわち, プログラム実行中に自身についての情報を取得することができる性質を指し, オブジェクト指向においても重要な概念である。イントロスペクションを備えるシステムは, 未知のインタフェースに遭遇した時でも, イントロスペクションを使用して相手のインタフェースについての情報を動的に取得し, 対応することが可能となる。SDRM モデルでは, リモートメモリアccessモデルにより交換したメッセージ内容を Self-Description より取得することになる。

3.4 SDRM モデルと遠隔ロボット制御

ここで, SDRM モデルの遠隔ロボット制御への適用方法について説明する。リモートメモリアccessモデルを遠隔ロボット制御に適用した場合, 遠隔ロボットの制御はリモートメモリの読み書きへとすべて抽象化されることになる。この場合, ロボットの各軸に対応した変数をリモートメモリとして割り当てる。図 3 にパン・チルトカメラによる簡単な例を示す。

図 3 で示したパン・チルトカメラの遠隔制御を行う

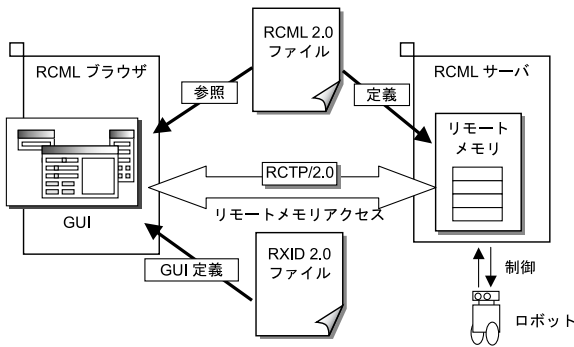


図 4 RCML 2.0 システム構成
Fig. 4 Configuration of RCML 2.0 System

場合、各軸に対応する変数の書き換えを行うことになる。例えば、パン軸を回転させるときは、パン軸に相当するリモートメモリ上の変数を外部から書き換える。このような適用方法は、実際のロボットのハードウェア構成とよく対応がとれているため、大変に理解しやすい。また、トレイグジスタンスのような、コマンドベースではなく、ロボットの各軸に対して直接制御指令を送るようなシステムに対しても、無理なく適合する。さらに、連続的な制御指令を効率よく伝達するための方法として、ストリームによるリモートメモリアクセスなどの概念も導入しやすい。一方で、コマンドベースのロボット制御に対応するために、リモートメモリの書き換えをトリガーとしたコマンド入力、すなわちメソッド呼び出しのセマンティクスを実現することも可能である。そのためには、リモートメモリアクセスにおいて、幾つかの変数の集合を一度に書き換えられることを保証する方法を用意しておく必要がある。

Self-Description には、リモートメモリのデータ構造に関する情報や通信に関する情報など、リモートメモリアクセスでのイントロスペクションに必要な情報の他に、ロボットのスペックや設置してある場所、ロボットへのアクセス権に関する情報など、遠隔ロボット制御に必要な付加的な情報も記述する。

3.5 全体設計

図4にRCML 2.0 システムの構成を示す。RCML 2.0 システムでは、RCML 2.0 と RCTP/2.0 の組み合わせで、SDRM モデルを実現する。ここで、RXID (RCML Extensible Interface Definition) 2.0 とは、RCML 2.0 システムで新たに取り入れられた GUI 定義用の言語になる。

4. RCML 2.0 システム仕様概説

4.1 RCML 2.0

RCML 2.0 は、各データに関する情報やリモートメモリ上に構築されるデータ構造などを定義するための記述言語として位置付けられる [23]。これらの情報を、

表 1 RCML 2.0 の要素
Table 1 Elements of RCML 2.0

| 要素 | 説明 |
|----------|---|
| <rcml> | RCML のルート要素。RCML サイトについての情報を宣言する。 |
| <group> | RCML のデータ構造において、データの集合体を宣言する。 |
| <access> | RCML クライアントが <data> ノードへアクセスするときの方法を宣言する。 |
| <data> | RCML データ構造において、アクセス対象である <data> ノードを宣言する。 |
| <link> | 親要素に関するリンクを宣言するための要素。 |
| <meta> | 親要素に関するメタデータを記述するための要素。 |

RCML 2.0 によって定められた形式でファイルに記述することにより、イントロスペクションを実現する。RCML 2.0 では、情報の記述に XML [22] (Extensible Markup Language) を用い、表 1 に示す 6 つの要素が定義されている。

表 1 において、<rcml> 要素から <data> 要素までの 4 つの要素を使用して、RCML データ構造と呼ばれるリモートメモリ上のデータの木構造を定義する。<group> 要素は、データの集合体や階層構造などを定義するための要素である。また、<group> 要素では、例えばロボットの管理者のみがアクセス可能なリモートメモリ領域の指定など、データのアクセス権に関する情報も同時に定義することが出来る。<access> 要素は、リモートメモリアクセスの通信に関する情報を記述するためのノードである。RCTP/2.0 で定義されている event 型もしくは stream 型などの通信方法の種類や、そのときのデータ送り出し間隔や守られるべきレイテンシ、帯域やタイムアウト値などを定義する。<data> 要素は、データを定義するための要素である。<data> 要素の各属性で、データの型や値の取りうる範囲などを宣言する。<link> 要素は、親要素に関する追加的な情報へのリンクを宣言するための要素である。<meta> 要素はデータの単位や意味など、データに対する付加的な情報 (メタデータ) を記述するための要素である。

4.2 RCTP/2.0

RCTP/2.0 は、RCML 2.0 システムにおいて、リモートアクセスを実現するためのプロトコルとして位置付けられる [24]。RCTP/2.0 は、HTTP/1.1 をもとにしたアプリケーションレベルのプロトコルであり、リモートメモリにアクセスする方法を提供し、さらに帯域などの制御に関する機能、アクセス権を管理する機能などを提供する。RCTP/2.0 では、コントロールコネクションとデータストリームの 2 種類の通信方法

表2 RCTP/2.0 のメソッド
Table 2 Methods of RCTP/2.0

| メソッド | 説明 |
|---------|--------------------|
| CONNECT | RCTP セッションを開始する。 |
| ACQUIRE | アクセス権を取得する。 |
| RELEASE | 取得したアクセス権を解放する。 |
| READ | <data> ノードの値を取得する。 |
| WRITE | <data> ノードの値を設定する。 |
| SETUP | ストリームの設定を行う。 |
| GO | ストリームの開始を指示する。 |
| PAUSE | ストリームの一時停止を指示する。 |
| STOP | ストリームの停止を指示する。 |
| BYE | セッションを終了する。 |

が定義されている。コントロールコネクションでは、主に、アクセス権の確保やデータストリームの制御を行う。コントロールコネクションは、コネクション・オリエンテッドな通信方法を用い、クライアントは、まずコントロールコネクションをサーバと確立する。

一方、データストリームでは、ロボットの遠隔操作に必要な制御データの連続的な通信を行う。データストリームは、リアルタイム性が要求されるロボットの制御データなどの転送に用いるため、コントロールコネクションとは異なり、パケットの再送制御などが行われぬコネクションレスな通信方法を用いる。

RCTP/2.0によるロボットの制御は、RCML 2.0 ファイルで定義されているデータ(リモートメモリ)へのアクセスとして抽象化されているが、RCTP/2.0ではデータへのアクセス方法として、READ/WRITE のメソッドを用いた通常のデータアクセスの他に、event 型アクセスと stream 型アクセスを定義する。event 型アクセスは、通常のデータアクセスと同じようにコントロールコネクションを使用するが、いくつかのデータをまとめてアクセスできる点とサーバ側でデータの書き換えがおこった際にイベントが発生する所が異なる。通常のデータアクセス及び event 型アクセスは、サーバ・クライアント間のデータ書き換えの確実性に重点をおいている。一方、stream 型アクセスは、コネクションレスな通信方法を用いたデータストリームを使用し、データ書き換えの確実性よりはレイテンシの低減に重点を置いている。

(1) コントロールコネクション

コントロールコネクションは、HTTP/1.1と同様に、コネクション・オリエンテッドな通信方法を用いたリクエストとレスポンスからなるプロトコルになっている。メッセージの構造も HTTP/1.1とは基本的に差異はなく、表2に示す10のメソッドが定義されている。

(2) データストリーム

データストリームは、コントロールコネクションとは異なり、パケット単位で通信が行われるコネクションレスな通信方法が用いられる。データストリームが

用いられるのは、stream 型アクセスのときである。リアルタイム性を確保するために、パケットの欠落などによるデータの再送制御を行わない。データストリームのパケットには、payload と呼ばれるデータ転送の最小単位を複数格納することができる。同時刻に転送が発生したいくつかの payload を一つにまとめることにより、データストリームのパケット数を減らすことが可能である。RCTP/2.0では、実データを転送する data-stream payload の他に、フロー制御を行う flow-control payload が定義されている。

4.3 RXID 2.0

RXID 2.0は、RCML 2.0とRCTP/2.0の組み合わせにより実現される SDRM モデル上で、GUI を構築するための言語として位置付けられる [25]。RCML 1.0 システムの開発時にも明らかになったように、GUI 部分は他のロボット遠隔制御部分などに比べるとより頻繁に変更が行われる可能性が高い。従って、GUI 部分をシステムの他の部分に出来るだけ影響を与えずに、変更出来るようにするのが望ましい。そのために、GUI 定義の外部化を実現するのが RXID 2.0 になる。

RXID 2.0 で定義されるユーザインタフェースの各要素と、RCML 2.0 で定義される RCML データ構造は、RXID 2.0 ファイルから RCML 2.0 ファイルへの一方向リンクで結び付けられる。一方向リンクとすることで、RCML 2.0 ファイルの独立性が高められ、RXID 2.0 ファイルに対する変更や、新たな RXID 2.0 ファイルの作成を行った場合でも、もとなる RCML 2.0 ファイルを変更する必要がない。一方向リンクを活用することにより、一つのロボットに対して複数のユーザインタフェースを同時に定義する事などが可能であり、多様な使用形態への対応が容易になっている。

RXID 2.0 は、RCML 2.0 と同じように XML を元にした言語仕様となっている。RXID 2.0 では、表3に示す要素群が定義されている。<window> 要素は常に <rxid> 要素の子要素であり、一つの <window> 要素が、RCML ブラウザによって表示される画面上の一つのウインドウに対応する。<window> 要素のそれぞれの属性により、表示される位置や大きさ、タイトルなどを設定する。<session> 要素は、<window> 要素の子要素として、関連づけられる RCML ファイルの URL を指定する。<access> 要素は、<session> 要素の子要素として、指定された RCML ファイル内の <access> 要素へのアクセス方法を定義する。<window> 要素は、RCML ブラウザがウインドウ内に表示するボタンやスクロールバーなどの各 GUI 要素を配置するための子要素をさらに有する。

表4に、GUI 要素を配置するために、RXID 2.0 で定義されている要素の一覧を示す。表4において、'R' は、

表 3 RXID 2.0 の要素
Table 3 Elements of RXID 2.0

| 要素 | 説明 |
|-----------------|-------------------------------|
| <rxid> | RXID のルート要素 . |
| <window> | ウィンドウを表示させるための要素 . |
| <session> | RCTP のセッションを定義するための要素 . |
| <access> | RCML データ構造へのアクセス方法を定義するための要素 |
| Widget elements | 画面に表示される Widget を定義するための要素群 . |

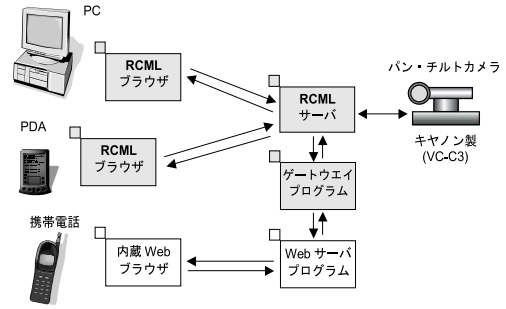


図 5 実験システム概要

Fig. 5 An outline of the experimental system

対応する GUI 要素が, RCML データ構造から値を読みとり可能であることを表している. 例えば, <slider> 要素で配置されるスライダバーは, スライダーノブの表示すべき位置を RCML データ構造から取得し, 得られた値をもとに, 画面を更新する. 一方, 'W' は, GUI 要素が, RCML データ構造に対して値を書き込むことが可能であることを表している. <button> 要素や <checkbox> 要素, <slider> 要素等は, RCML データ構造の値を変更可能な GUI 要素である. RCML データ構造に対して, 書き込みも読み込みも行わない GUI 要素も存在する. 例えば, <box> や <label> などが, RCML データ構造とは関係を持たず, 読みとりも書き込みもおこなわない, いわば静的な GUI 要素である. RCML データ構造に対して, 読みとりもしくは書き込みを行う GUI 要素は, すべて 'dataPath' 属性を有している. 'dataPath' 属性には, アクセスを行う RCML データ構造内の <data> ノードへのパスが記述され, RXID ファイルから RCML データ構造への一方向リンクとして機能する.

5. 評価実験

我々は, 新たなシステム設計の評価の為に, RCML 2.0 システムの試作も行った. 今回は, 試作したシステムが様々なクライアント実行環境に対してシームレスに対応できることを示すために, 通常の PC に加えて, PDA および携帯電話においても実験を行った. 遠隔制御の対象としては, いずれもパン・チルトカメラ (キヤノン社製 VC-C3) を用いた. 構築したシステムの概要を図 5 に示す.

5.1 PC 環境での評価

試作を行ったシステムは, サーバ側でロボットの制御を行う RCML サーバと, クライアント側で使用される RCML ブラウザからなる. いずれのプログラムも, 開発は Windows2000 上の C++ で行われており, XML プロセッサには XML for C++ Parser v2.3.1 を用いている. RCML サーバは, クライアントからの接続を受け付けるサーバプロセスと, 各ロボットごとに用意されるロボットモジュール・プロセスからなる.

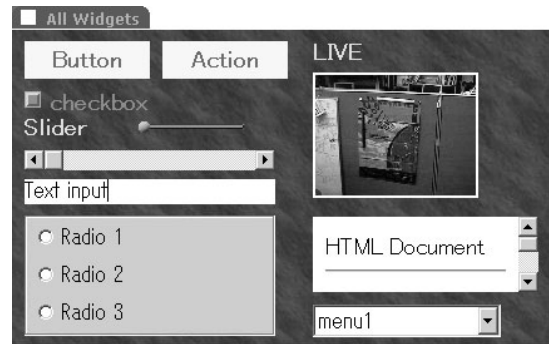


図 6 全 GUI 要素表示

Fig. 6 Displaying all widgets

サーバプロセスは, クライアントからの RCTP/2.0 接続の処理およびアクセス権の管理などを担当し, ロボットモジュール・プロセスは, 各ロボット固有の処理を行う. RCML ブラウザは, 単体のアプリケーションとして実装されており, 通常の Web ブラウザと同じように, URL を入力することにより目的とする RCML サーバに接続する.

今回, 試作を行ったシステムにおいて, 制御対象専用のモジュールは, ロボットモジュール・プロセスだけであり, 他のモジュールはすべて汎用的に使用できるように設計されている. したがって, 別のロボットに対応する場合においても, 新たに用意する必要があり, ロボットモジュール・プロセスと RCML 2.0 および RXID 2.0 の各ファイルのみである.

RCML ブラウザは, 読み込んだ RXID ファイルに従って動的に GUI を生成する. 図 6 に, 試作した RCML ブラウザで使用可能な全 GUI 要素を表示した例を示す.

図 7 は, 実際に RCML サーバへ接続し, パン・チルトカメラの制御を行った際の画面表示である. 画面には, 遠隔のパン・チルトカメラによるライブ映像と操作のためのスクロールバーが表示されている. 今回は, ライブビデオの転送手段として, Microsoft 社のビデオ会議ソフトウェアである NetMeeting を使用した. また, RCML 1.0 で, その実効性が確認された, VRML ビューも, VRML ブラウザと連携をはかるこ

表 4 GUI 定義要素
Table 4 Widget Elements

| 要素 | 説明 | R | W |
|----------------|----------------------------------|---|---|
| <box> | 四角い枠を表示する要素 . | × | × |
| <label> | ラベルを表示する要素 . 表示される文字列は静的である | × | × |
| <text> | テキストを表示する要素 . 表示される文字列は動的に変更可能 . | | × |
| <button> | ボタンを表示する要素 . | × | |
| <checkbox> | チェックボックスを表示する要素 . | | |
| <radioGroup> | ラジオボタンの集まりを表示する要素 . | | |
| <scrollbar> | スクロールバーを表示する要素 . | | |
| <slider> | スライダーバーを表示する要素 . | | |
| <edit> | 編集可能なテキストを表示する要素 . | | |
| <popUpMenu> | ポップアップメニューを表示する要素 . | | |
| <liveVideo> | ライブビデオを表示する要素 . | × | × |
| <html> | HTML ビューアーを表示する要素 . | × | × |
| <actionButton> | 動作を割り当て可能なボタンを表示する要素 . | × | × |



図 7 パン・チルトカメラ制御時の画面
Fig. 7 Control panel of the pan/tilt camera

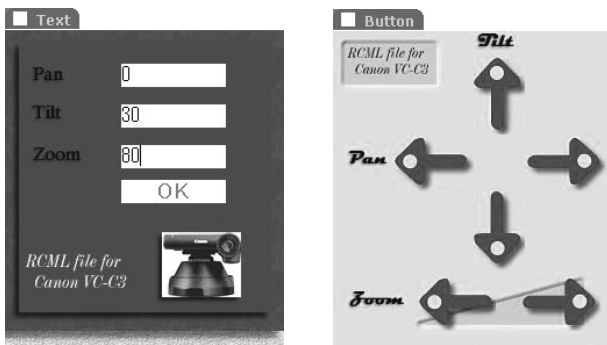


図 8 異なるデザインの制御用パネル
Fig. 8 Various designs of control panel

とにより実現している .

図 8 は、同じパン・チルトカメラに対して、異なるデザインの制御用パネルを表示した例である . ここで、一つのパン・チルトカメラに対して複数のユーザインタフェースを表示する例を示したが、変更を行ったのは RXID ファイルのみであり、RCML ブラウザおよび RCML サーバのプログラムの変更は一切行っていない . 従って、個々のユーザインタフェースの切り替えも、RCML ブラウザに入力する URL を変更するだけでよい .

(1) 性能測定

試作したシステムの性能測定として、RCTP/2.0 のコントロールコネクション及びデータストリームにおける、クライアントからサーバの <data> ノードの書き換えに要する時間の測定を行った . サーバ側には変更を加えず、クライアント側に測定用のプログラムを作成した . いずれの測定においても、純粋なシステムのオーバーヘッドを検証する意味で、ロボットの制御に要した時間は含まれていない .

TCP 接続であるコントロールコネクションは、通常の手順でセッションを開始した後に、クライアント側より、特定の <data> ノードに対して write メソッドを複数回発行し、write メソッドのリクエストを発行してからレスポンスを受信するまでの個々の時間を測定した . 測定に使用した <data> ノードはパン・チルトカメラのパン軸に相当し、変数長は int (4byte) である . 測定時に発行されたリクエストのメッセージ長は 111byte、レスポンスのメッセージ長は 48byte であった .

一方、UDP 接続のデータストリームは、クライアント側より、<data> ノードの書き換えを意味するパケットを送信し、書き換えが終了したことを意味するパケットを受け取るまでの時間を測定した . 測定に使用した <data> ノードは、コントロールコネクションの測定と同じノードを使用し、測定時に発行された UDP パケットのデータ長は 24byte であった .

測定は、それぞれ 1 万回連続して行った . 結果を、表 5 に示す . Local とは、測定用のプログラムをサーバ側の同一マシン上で動作させた時の測定値であり、ネットワークでの通信に要する時間を含まないプログラムの処理時間を表す . LAN で示される項目は、100Mbps の Ethernet のネットワークにサーバ及びクライアントのみを接続した時の測定値である . それぞれの測定結果において、最大値が平均値及び標準偏差に比べると比較的大きな値をとっている . 特に、Local において

表 5 性能測定

Table 5 Performance of the System

| | コネクション | 平均 | 標準偏差 | 最小 | 最大 |
|-------|--------------|--------|---------|--------|-------|
| Local | コントロールコネクション | 0.44ms | 0.031ms | 0.41ms | 1.6ms |
| | データストリーム | 0.32ms | 0.026ms | 0.30ms | 1.5ms |
| LAN | コントロールコネクション | 0.73ms | 0.31ms | 0.46ms | 2.0ms |
| | データストリーム | 0.64ms | 0.31ms | 0.39ms | 1.9ms |

サーバ: Windows 2000, PentiumIII 700MHz, 256MB

クライアント: Windows 2000, PentiumIII 800MHz, 256MB

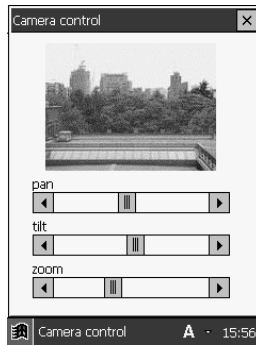


図 9 PDA での表示

Fig. 9 The control panel displayed by PDA

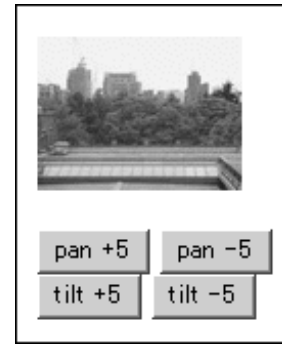


図 10 携帯電話に表示される画面

Fig. 10 The control panel displayed by celler phone

その傾向が顕著に見られる。実際の Local の測定データを見ても、連続して 1 万回行った測定の内、1ms 以上かかった例が数回発生している。これは、定期的に行われるシステムプロセスなどの影響によるものと思われる。極力測定対象以外のプロセスが実行されていない状態で測定を行ったが、影響を完全に排除することは出来なかった。今回のシステムで使用したパン・チルトカメラの実際に制御にかかる時間は、パン軸回転動作指令として 10 度を入力したときに約 320ms であり、システムのオーバーヘッドは、制御に要する時間に比べて十分に小さい結果が得られている。

5.2 PDA 環境での評価

PDA での評価は、Java で記述された RCML ブラウザを用意し、Windows CE 3.0 (ヒューレット・パッカード社製, Pocket PC, SH-3 133MHz, 32MB) における PersonalJava 実行環境上で行った。PC 環境と同じ RCML サーバに接続し、パン・チルトカメラの制御パネルを表示した画面を図 9 に示す。

作成した PDA 環境用の RCML ブラウザは、XML プロセッサ (JAXP Ver.1.0) を備え、PC 上での RCML ブラウザとほぼ同等の機能を有する。PC 環境の時と同じように、別の URL を入力することにより、異なるユーザインタフェースを表示することが可能である。使用した PDA の画面は、240x320 と小さい上に、画面の縦横比も通常の PC とは異なり縦長であるため、画面のサイズに合わせた RXID ファイルをそれぞれ用意することで、PDA に最適なユーザインタフェースを実現した。PC 環境での測定と同条件で、コント

ロールコネクションにおいて write メソッドのリクエストを発行してからレスポンスを受信するまでの時間を測定したところ、約 63ms であった。PDA より、実際にパン・チルトカメラの制御を行った場合においても、PC 環境とほぼ同じようにカメラを操作することが可能であった。今回は、開発の容易性から、Java を選択したが、個々の PDA にあわせてネイティブのアプリケーションとすることで、さらなるパフォーマンスの向上が可能であると考えられる。

5.3 携帯電話環境での評価

クライアントが携帯電話の場合を想定し、図 5 に示す構成のシステムを構築した。構築したシステムは、HTML もしくはそのサブセットとなる規格を表示可能な Web ブラウザを内蔵した携帯電話を対象とする。図 5 に示すように、RCML サーバと最終のクライアントである携帯電話との間に、ユーザインタフェース画面として動的に HTML ファイルを生成するゲートウェイプログラムを配置する。ゲートウェイプログラムは、設定された RCML ファイルおよび RXID ファイルにあわせて HTML ファイルを動的に生成するため、RCML ブラウザと同じく、後は設定を変えるのみで汎用的に使用することが出来る。また、携帯電話は事業者ごとに表示可能な HTML の書式が異なることがある。そういった場合でも、各事業者に対応した HTML を生成するゲートウェイプログラムを用意するだけでよい。

ゲートウェイプログラムは、WebObjects4.5 を用い

て作成した．携帯電話上の Web ブラウザで，他の実験と同じパンチルトカメラが接続された RCML サーバにゲートウェイプログラム経由で接続した際に表示される画面を図 10 に示す．Web ブラウザからのアクセスになるため，操作を連続的に行うことは出来ず，基本的にコマンド単位の操作となる．通信速度が 9600bps の携帯電話を用いて，一回操作を行って次の操作が可能となるまでの時間を計ってみたところ，画像を同時に転送すると約 11 秒，画像を転送しない状態で 2.7 秒であった．一回の操作毎に転送される HTML ファイルなどのデータ量が約 7.2KByte で，その中で画像データが 96% (6.9KByte) を占めている．使用した携帯電話の通信速度が 9600bps と遅いため，転送データ量の影響を直接受ける結果となった．速度が要求される操作は難しいが，遠隔の機器の運転状態を数値などで確認を行ったり，簡単なパラメータの設定など，用途を選べば十分に実用可能であると考えられる．また，通信速度等の向上が図られている第三代携帯電話等を用いることにより，さらなる応用範囲の拡大が期待される．

5.4 考察

PDA 環境での評価実験で使用した RCML ブラウザは，Java で記述されているため，PersonalJava 以上の実行環境であれば，別のクライアント上でも実行することが出来る．そこで，Java による実装のパフォーマンスを測定する意味で，Java で記述された RCML ブラウザを，PC 上で前記 PC 環境の Local と同一条件で実行したところ，コントロールコネクションにおいて write メソッドのリクエストを発行してからレスポンスを受信するまでの時間が，C++ で記述された PC 環境用の RCML ブラウザが 0.44ms であったのに対して約 1.4ms であった．また，携帯電話環境で使用したゲートウェイプログラムに PC 環境の Web ブラウザを用いて接続した場合，毎回の操作に 1 秒程度の時間を要した．このことより，RCML 2.0 のシステムは，実装を CGI もしくは Java に限ったシステムに比べ，それぞれのクライアント環境に合わせて，最適なパフォーマンスを発揮可能な実装を柔軟に選ぶことが出来るため，パフォーマンスの観点から見ても有利であると言える．今回の性能測定で使用したパンチルトカメラは，2 軸からなる非常に簡単な機構しか有しておらず，システムの評価としてはさらなる検討が必要とされるが，RCML 2.0 システムが主たる対象としている監視カメラやペットロボットのようなパーソナルロボットに対して，ネットワーク経由で連続的に制御データを送信するために必要とされる基本的パフォーマンス及び機能を有していることは確認できた．

6. まとめ

本論文では，アールキューブ構想に基づいた遠隔ロボット制御技術である RCML 2.0 の設計及びその評価について述べた．RCML 1.0 で明らかになった改良すべき点をふまえつつ，RCML 2.0 では，アールキューブ構想におけるボトムアップアプローチとして，誰でも簡単に使い，気軽に利用可能なシステムを目指して開発を行った．RCML 2.0 の設計にあたっては，多様な使用形態への対応と遠隔ロボット制御に最適な設計がなされていること，なおかつフレームワークとして機能することを設計指針として掲げ，以下に示す新たな機軸をシステム設計に導入した．

- Self-Descriptive Remote Memory モデルの提案
- SDRM モデルに基づいた新たなシステム設計
- GUI 記述言語である RXID の導入

これにより，設計指針を満たすシステム設計を示すことが出来た．さらに，RCML 1.0 において懸案であった制御情報とユーザインタフェースの分離が成し遂げられ，ユーザインタフェースの定義において最大限の自由度を引き出す設計が実現した．評価実験においては，RXID ファイルの変更により，様々なユーザインタフェースが表示可能であること及び，PC 環境に限らず，PDA 環境や携帯電話などの様々なクライアント環境に対してもシームレスに対応可能であることを示した．

今後は，より多くの実システムへの適用を行いながら，さらに使いやすいシステムとなるよう開発を続けていく予定である．

参考文献

- [1] 広瀬真人, 竹中透, 五味洋, 小澤信明: 人間型ロボット, 日本ロボット学会誌, Vol. 15, No. 7, pp. 983-985 (1997).
- [2] 佐部浩太郎, 藤田雅弘: エンターテインメントロボットの商品化, 日本ロボット学会誌, Vol. 18, No. 2, pp. 185-187 (2000).
- [3] 通産省アールキューブ研究会 (編): アールキューブ, 日刊工業新聞社 (1996).
- [4] Susumu Tachi: Real-time Remote Robotics-Toward Networked Telexistence, *IEEE Computer Graphics and Applications*, Vol. 18, No. 6, pp. 6-9 (1998).
- [5] S. Tachi, K. Tanie, K. Komiyama and M. Kaneto: Tele-Existence (I): Design and Evaluation of a Visual Display with Sensation of Presence, *Proc. of the 5th Symposium on Theory and Practice of Robots and Manipulators*, pp. 245-254 (1984).
- [6] 館暉, 小森谷清, 澤田一哉, 井床利之, 井上幸三: 人間協調・共存型ロボットシステム研究開発プロジェクト, 日本ロボット学会誌, Vol. 19, No. 1, pp. 16-27 (2001).
- [7] Yasuyuki Yanagida, Naoki Kawakami and Susumu Tachi: Development of R-Cubed Manipulation Language - Access Real Worlds Over the Network, *Proc. 7th International Conference on Artificial*

[著者紹介]

- Reality and Tele-existence '97*, pp. 159-167 (1997).
- [8] Weichung Teng, Akira Nukuzuma, Naoki Kawakami, Yasuyuki Yanagida and Susumu Tachi: Development of R-Cubed Manipulation Language - The specification of RCML and RCTP, *Proc. 8th International Conference on Artificial Reality and Tele-existence '98*, pp. 152-162 (1998).
- [9] Weichung Teng, Dairoku Sekiguchi, Akira Nukuzuma, Naoki Kawakami, Yasuyuki Yanagida and Susumu Tachi: Development of R-Cubed Manipulation Language - Implementation and Evaluation of RCML System, *Proc. 9th International Conference on Artificial Reality and Tele-existence '99*, pp. 79-83 (1999).
- [10] 鄧惟中, 関口大陸, 川上直樹, 柳田康幸, 奴久妻章, 舘暉: RCML: アールキューブ操作言語の開発, Vol. 5, No. 2, pp. 881-890.
- [11] : The Web3D Consortium: ISO/IEC 14772-1:1997 Virtual Reality Modeling Language (VRML97). <http://www.web3d.org>.
- [12] Dairoku Sekiguchi, Weichung Teng, Yasuyuki Yanagida, Naoki Kawakami and Susumu Tachi: Development of R-Cubed Manipulation Language - The design of RCML 2.0 system, *Proc. 10th International Conference on Artificial Reality and Tele-existence 2000*, pp. 44-51 (2000).
- [13] R. Simmons: Xavier: An Autonomous Mobile Robot on the Web, *Proc. IROS'98 Workshop 'Robots on the Web'*, pp. 43-48 (1998).
- [14] Patrick Saucy and Francesco Mondada: KhepOn-TheWeb: One Year of Access to a Mobile Robot on the Internet, *Proc. IROS'98 Workshop 'Robots on the Web'*, pp. 23-30 (1998).
- [15] Matthew Stein: Painting on the World Wide Web: The PumaPaint Project, *Proc. IROS'98 Workshop 'Robots on the Web'*, pp. 37-42 (1998).
- [16] 原功, 他: ワールドワイドロボティクスの提案-その2:WWW ブラウザを用いた遠隔操作環境の構築, 第2回 ロボティクスシンポジウム予稿集, pp. 114-119 (1997).
- [17] : The Object Management Group: The Common Object Request Broker Architecture (CORBA) specification. <http://www.omg.org>.
- [18] : Microsoft corporation: Distributed Component Object Model (DCOM). <http://www.microsoft.com>.
- [19] Hirohisa Hirukawa and Isao Hara: The Web Top Robotics, *Proc. IROS'98 Workshop 'Robots on the Web'*, pp. 49-54 (1998).
- [20] M. Mizukawa, H. Matsuka, T. Koyama and A. Matsumoto: De-facto standard API for Open and Networked Industrial Robots, *Proc. 30th Int. Symp. on Robotics*, pp. 455-462.
- [21] : W3C:Simple Object Access Protocol (SOAP) 1.1. <http://www.w3c.org>.
- [22] : W3C: Extensible Markup Language (XML) 1.0. <http://www.w3c.org>.
- [23] 関口大陸: RCML 2.0 仕様書 <http://www.rcml.org>
- [24] 関口大陸: RCTP/2.0 仕様書 <http://www.rcml.org>
- [25] 関口大陸: RXID 2.0 仕様書 <http://www.rcml.org>

関口 大陸 (正会員)



2001年東京大学大学院工学系研究科博士課程修了, 博士(工学)。同年より科学技術振興事業団研究員。2002年東京大学大学院情報理工学系研究科 助手となり現在に至る。ネットワークロボティクスおよびテレグジスタンスに関する研究に従事。

川上 直樹 (正会員)



1999年東京大学大学院工学系研究科博士課程修了。博士(工学)。同年より東京大学大学院情報理工学系研究科 助手。2001年東京大学大学院情報理工学系研究科 講師となり現在に至る。テレグジスタンス及びネットワークロボティクスの研究に従事。

舘 暉 (正会員)



1973年東京大学大学院工学系研究科博士課程修了工学博士。同年東京大学工学部助手, 機械技術研究所等を経て, 1989年東京大学先端科学技術研究センター助教授, 1992年同教授, 1994年東京大学工学部教授, 2001年同大学大学院情報理工学系研究科システム情報学専攻教授。現在に至る。信号処理, 盲導犬ロボット, テレグジスタンス等に関する研究に従事。

(2002年4月10日受付)