

# RCML: アールキューブ操作言語の開発

鄧惟中<sup>\*1</sup> 関口大陸<sup>\*1</sup> 川上直樹<sup>\*1</sup> 柳田康幸<sup>\*1</sup> 奴久妻章<sup>\*2</sup> 舘暲<sup>\*1</sup>

## Development of R-cubed Manipulation Language

Wei-Chung TENG<sup>\*1</sup>, Dairoku SEKIGUCHI<sup>\*1</sup>,  
Naoki KAWAKAMI<sup>\*1</sup>, Yasuyuki YANAGIDA<sup>\*1</sup>, Akira NUKUZUMA<sup>\*2</sup>, and Susumu TACHI<sup>\*1</sup>

**Abstract** - R-Cubed ( $R^3$ : Real-time Remote Robotics) is the concept and technology for people to telexist anywhere in the world by controlling remote robots as his avatars in real time base through the network. An R-Cubed system is supposed to allow people to control remote robots from terminals installed on homes, offices, or any public booths connected to Internet or other dedicated networks. In this paper some efforts devoted to the design of R-Cubed systems are shown, including the specification of RCML (R-Cubed Manipulation Language) that describes remote environments and robot characteristics, and RCTP (R-Cubed Transfer Protocol) that is in charge of network communication between users and robots in real time base. Finally an experimental implementation example and evaluation to the behavior of this system is reported.

**Key Words:** Telexistence, R-Cubed, RCML, RCTP, VRML

### 1. はじめに

アールキューブとはネットワーク環境を利用して感覚センサを備えた遠隔ロボットを臨場感を持ちつつ制御することであたかも自分が遠隔環境に実在するかのように感じ、あるいは行動することを目指している研究開発構想とそのための基礎研究である。Real-time Remote Robotics (実時間遠隔制御ロボット技術)の頭文字をとって  $R^3$  と表記し、これをアールキューブと呼んでいる[1]。

アールキューブ構想では、自宅や会社、公共施設などに設けられた操作端末からネットワークを介して職場、公共空間、危険地帯などにある遠隔ロボットを制御することにより、時間や空間などの物理的制限を超越し、遠隔環境に自由に存在し、活動することを可能にする(図1)。現在のネットワークの利用形態は情報メディアを扱うことが殆どで、つまり計算機内部のいわゆるバーチャル空間で閉じており、我々が実際に存在する実世界とは分断されている。アールキューブのようなネットワークロボティクス技術はロボットを制御し、実世界に対して物理的な作用を及ぼすことにより、このような閉じた空間から開放し、より積極的に現実空間同士を、あるいは現実空間とバーチャル空

間とを結合することを目指している。

アールキューブネットワークの実現へ向けて、われわれは HTML や VRML のような手軽さで利用できる標準的な手法の開発および提案を行ってきた[2][3]。本論文では、遠隔ロボット制御および感覚情報提示に関する遠隔環境、制御情報、感覚情報などの標準的な記述言語である RCML (R-Cubed Manipulation Language、アールキューブ操作言語)、またそのための通信手法 RCTP (R-Cubed Transfer Protocol、アールキューブ通信プロトコル)を設計し、その実装を行う。

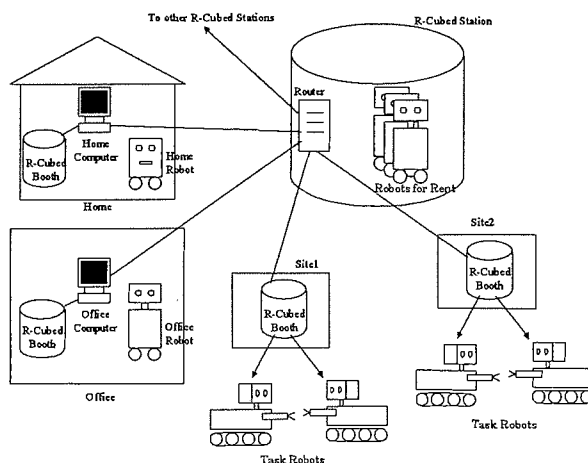


図1 アールキューブ実現イメージ

Fig. 1 Concept of R-Cubed network

\*1: 東京大学 大学院工学系研究科

\*2: ミノルタ (株) 高槻研究所

\*1: School of Engineering, The University of Tokyo

\*2: Takatsuki Laboratory, Minolta Co. Ltd.

ロボットをインターネットや ISDN 回線を通じて遠隔操作する研究が近年活発に行われている。しかし、そのための標準の言語やプロトコルの開発には至っておらず、ユーザインターフェースや通信内容はまちまちであり、解決すべき課題もたくさんある。これらの遠隔操作システムは主に WWW ブラウザと CGI(Common Gateway Interface)を用いたアプローチ、または VRML(Virtual Reality Modeling Language)[4][5]と Java を取り入れたアプローチにより実現されることが多い。以下に両者の特徴と問題点を述べる。

## 2.1 WWW ブラウザと CGI を用いたアプローチ

WWW 環境を利用した遠隔ロボット制御システムの多くはこのアプローチをとっている。WWW ブラウザをユーザインターフェースとして、その通信手段を CGI で実装するこのタイプのシステムは、CMU の Xavier などの例がある[6][7][8]。このようなシステムでは、広く普及している WWW ブラウザをそのまま利用できることや、実装が簡単である利点がある反面、次のような制限を持つ：

- (1) HTML コンテンツと CGI スクリプトは構造上お互い独立しているため、ロボット操作の記述と実行の間に何ら統一性が保証されておらず、汎用的なロボット操作に関して整合性を保つことは困難である。
- (2) 通信プロトコルは HTTP を用いたため、ロボット側からの応答は HTML 文書で返さなくてはならない。これでは通信データに余分なオーバーヘッドが生じるし、制御実行の状態に応ずる通信モードの設定といった機能も提供されない。
- (3) ユーザインターフェースは HTML がサポートするボタンなどの汎用 GUI(Common Graphic User Interface)に限定される。よって、3次元位置センサーなどロボット制御に適した入力装置の導入が困難である。

実際このタイプの操作システムで、例えば移動ロボットを操作する場合は、実行コマンドは前後進や回転といった非常に単純な指令に制限されているのが殆どである。また、ロボットから遠隔環境に作用を及ぼすことがなく、システム全体として時間遅れがあってもいいような応用に対して、このようなアプローチをとる場合が多い。

WWW ブラウザや HTTP 自体は本来、文書や図など2次元の静的なメディアを閲覧することを想定して設計されたものである。しかしながら、臨場感を持ちつつ遠隔ロボットを自在に動かすためには、センサからの大量な情報を処理し、それを踏まえて制御コマンドを実時間でロボットに入力する必要がある。それを実現す

るためには、実時間性および高度な応答性を持つ通信プロトコルの導入や効果的なユーザインターフェースを実現する操作体系が問われる。

## 2.2 VRML と Java を用いたアプローチ

上記のシステムが遭遇した問題点のうち、ユーザインターフェースや通信プロトコルの問題に関する解決策として、VRML と Java を取り入れたシステムも提案されている[9][10]。このタイプの操作システムでは、Java の通信機能を用いて、HTTP の代わりに他のアプリケーションプロトコルの導入により、通信データの形式を自由に設計できるようになるほか、VRML ブラウザと Java に基づいた操作システムでは汎用 GUI に限らず物理入出力装置の利用も考えられる。しかし、3次元バーチャル空間へのアクセスに国際標準の VRML 言語を用いた反面、実空間へのアクセス手法が標準化されていないのが現状である。

上で述べた問題点を解決するため、われわれは汎用性のあるユーザインターフェース構成を持ち、なおかつバーチャル空間と実空間へ統合した手段でアクセスできるシステム：RCML システムを設計する。さらに、RCML システムの概念に基づいた記述言語 RCML 及び通信プロトコル RCTP の提案をする[3][11]。以下、それぞれの詳細を説明する。

## 3. RCML システムの詳細

RCML システムの機能レベルでの構成を図2に示す。本構成では、ロボット側をサーバ、ユーザ側をクライアントとして定義し、またサーバとクライアントは1対1で接続されることを前提としている。

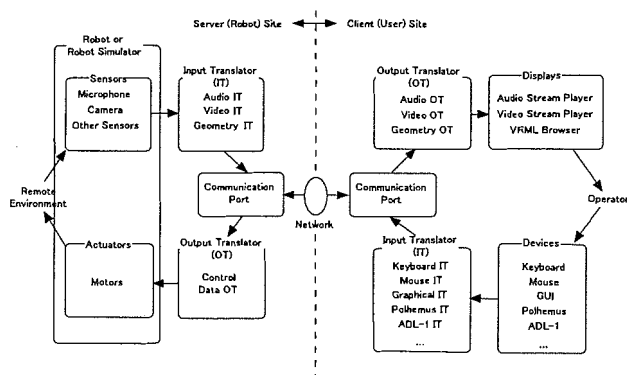


図2 RCML システム構成図

Fig. 2 Block dialog of RCML system

### 3.1 オブジェクト

本構成において、情報の入出力を行う機能上のユニットをオブジェクトと呼ぶ。オブジェクトはその機能から次の3種に分類される。

- (1) System Object
- (2) Input Object
- (3) Output Object

System Object には Server Object と Client Object の 2 種類があり、それぞれサーバとクライアントの計算機のハードウェアおよび RCML システムを実現するソフトウェアを指す。Input Object 及び Output Object は計算機内の情報空間と実世界との入出力を行う機能上のユニットである。RCML システムにおける入力・出力は常に計算機内の情報空間を基準として考える。すなわち、カメラは入力装置であり、画像ディスプレイは出力装置である。また、オブジェクトは実際の装置自体を単位とするのではなく、あくまで機能上のユニットを単位として考える。例えば力覚フィードバック付きの 3 次元位置計測装置は位置を入力する Input Object であり、同時に力覚を出力する Output Object であり、実体は 1 つの物体であっても同時に 2 つのオブジェクトとして扱われる。

Input Object/Output Object は対象とする入出力情報からさらに次の 5 種類に細分される。

- (1) Video Input/Output Object
- (2) Audio Input/Output Object
- (3) Control Input/Output Object
- (4) Text Input/Output Object
- (5) Binary Input/Output Object

これらのオブジェクトはそれぞれ画像、音声情報、制御情報、文字情報並びに 2 値情報の入出力に用いられる。また、原則としてサーバ側のすべてのオブジェクトはクライアント側のすべてのオブジェクトと 1 対 1 の対応づけがなされる。

### 3.2 入出力装置

計算機内の情報空間と実空間とのインターフェースとなる実際の入出力装置はその実装手段から次の 3 種の実装に分類される。

- (1) ハードウェア実装
- (2) GUI 実装
- (3) バーチャル実装

ハードウェア実装はマウスやジョイスティックなどの実際のデバイスを利用する実装であり、GUI 実装はボタンのクリックやスクロールバーのドラッグなどいわゆる GUI 的な手段を用いて実装することを示す。また、バーチャル実装とは実世界との入出力を行わない計算機内部でのバーチャル的、シミュレーション的な実装である。RCML の特徴としては各実装法を区別することなくシームレスに扱う。

### 3.3 トランスレータ

トランスレータとは特定のデバイスやディスプレイの性質に依存する情報とネットワーク上に流れる標準形式の情報の変換を行うユニットである。例えば、リンク式位置・姿勢計測装置などの 3 次元入力デバイスにより計測した情報（例：関節角）は Input Translator により標準的な空間記述形式（デカルト座標）に変換され、ネットワークを通してサーバ側へ送られる。その情報はロボット側で Output Translator によりマニピュレータの構造に依存した制御指令値へと変換される。感覚情報側の例を挙げると、カメラから得られたビデオ信号はサーバ側の Input Translator（この場合は Video Stream Encoder）により標準形式のビデオストリームへと変換され、ネットワークに送出される。ユーザ側で受信されたビデオストリームは適切な形に変換されてディスプレイへと出力される。トランスレータを通して RCML システムに入出力装置への汎用性を持たすことができると考えられる。

RCML システムでは、入出力装置にその装置に対応するトランスレータを合わせて一つの Input Object/Output Object を構成すると考える。

### 3.4 実現イメージ

RCML による実現イメージを図 3 に示す。VRML ブラウジングの場合と同様に、WWW サーバ上に RCML を用いて記述したロボット記述ファイルを用意し、このファイルを WWW ブラウザがダウンロードして WWW ブラウザにプラグインされている RCML ブラウザを起動する。RCML ブラウザは記述ファイルの内容を読み込み、遠隔環境の 3 次元モデル、ロボットの位置、自由度配置などの情報、諸元、制御パラメータ並びに入手可能遠隔情報の種類などを解析する。RCML ブラウザは解析結果を元にこれらの情報を提示する。ユーザが接続ボタンを押すと、ブラウザは RCTP プロトコルを用いて遠隔ロボットサーバに接続要求を行う。接続が確立された後、ブラウザは制御のためのユーザインターフェースや画像などの遠隔情報表示ウィンドウを生成し、リアルタイムに遠隔ロボットと通信を行い、ユーザの操作を遠隔

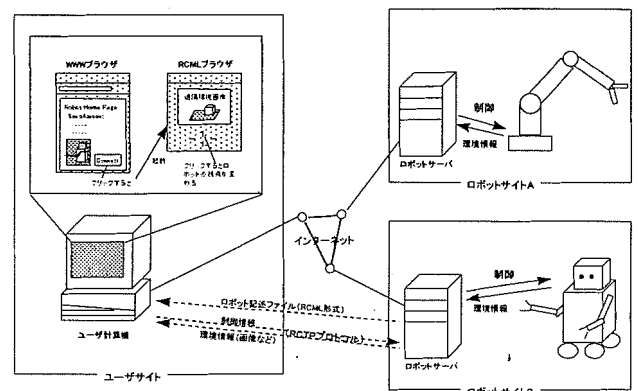


図 3 RCML による実現イメージ

Fig. 3 How an RCML system works

ロボットに伝達するとともにロボットから入手する情報を提示する。

#### 4. RCML 記述言語

RCML は、ロボットが存在する遠隔環境およびロボットの特性（自由度配置、制御可能パラメータ、センサ情報）などを記述するための言語である[11]。RCML により記述された情報はロボットを制御する際、ユーザー側に遠隔環境のモデルやロボット操作インターフェースについて提示する。以下、RCML 言語について説明する。

##### 4.1 設計基準

RCML 言語の設計にあたって、以下の要求を満たすように配慮した：

- (1) 一貫性：RCML ではバーチャル空間を扱う方法と実空間を扱う方法が一貫性をもっていなければならない。
- (2) 互換性：RCML は現存の言語標準とできるだけ互換性を持つ。
- (3) 統一性：RCML ではすべての入出力装置を均一した手法で記述し、情報のやり取りをしなければならない。
- (4) 汎用性：RCML の使用は特定した作業プラットフォームやオペレーティングシステムに限られてはならない。

##### 4.2 設計方針

現在インタラクティブな 3 次元空間を記述する言語として VRML97[5]が既に国際標準となっている。この

標準化されているバーチャル空間へのアクセス方法とシームレスな形で実空間へのアクセス手段を提供するため、我々は VRML97 をベースにして RCML の設計を行った。ジオメトリなどの記述に関しては VRML97 を用い、実世界の記述など拡張機能は VRML97 の PROTO を用いて表現する手段を採用した。RCML 拡張機能を解釈しない VRML ブラウザでも正しく読み込むことができ、そのブラウザが単に RCML についての拡張機能を見捨てるだけである。エンドユーザの視点からみると、RCML ブラウザを用いるアプリケーションと VRML ブラウザを用いるアプリケーションの相違点はユーザが現実環境とバーチャル環境のどちらにアクセスするのかという一点のみである。

##### 4.3 RCML ファイルの構造

RCML ファイルには記述の対象が二つに分類できる。一つは VRML による遠隔環境の 3 次元モデルの記述で、もう一つは RCML Robot 拡張ノードに囲まれたロボットの制御情報および感覚情報についての記述である。RCML Robot 拡張ノードの内部構成を表 1 にまとめる：

###### 4.3.1 RCML\_Robot ノード

すべての RCML 記述情報はこのルートノードに含まれる。RCML\_Robot ノードの下に、オブジェクトの種類ごとに対応するノードが存在する。例えば、Video Input Object に関する記述は RCML\_VideoInput ノードに収まる。

RCML\_Robot ノードの使用例として、ビデオカメラ（画像入力装置）を持つロボットを RCML ファイルで記述する場合は、Video Input Object の他に制御情報に関する記述の Control Object を含めて、RCML\_Robot は以下のように記述される。

```
RCML_Robot {
  RCML_Server { . . . }
  RCML_Client { . . . }
  Rcm1_ControlInput { . . . }
  Rcm1_ControlOutput { . . . }
  RCML_VideoInput { . . . }
}
```

###### 4.3.2 レベル 2 ノード

RCML\_Server ノードを例にして説明する。このノードの内容が下記のように定義されている：

```
PROTO RCML_Server [
  field SFString language "RCML/1.0"
  field SFString os ""
  field SFString serversoft ""
```

表 1 RCML におけるノード構造  
Table 1. Node Structure in RCML

Level 1 Node		RCML_Robot
Level 2 Node	System Object Node	RCML_Server RCML_Client
	Input Object Node	RCML_VideoInput RCML_AudioInput RCML_ControlInput RCML_TextInput RCML_BinaryInput
	Output Object Node	RCML_VideoOutput RCML_AudioOutput RCML_ControlOutput RCML_TextOutput RCML_BinaryOutput
Level 3 Node	RCML_Control Input Node	RCML_ControlInputData
	RCML_Control Output Node	RCML_ControlOutputData

```

] {
  Script {
    field SFString language IS language
    field SFString os IS os
    field SFString serversoft IS serversoft
    url ""
  }
}
    
```

ノードの中にはサーバ側の使用言語、オペレーティングシステム並びにサーバソフトウェアなどの情報が記述される。また、url フィールドには RCTP サーバの URL を示す。

RCML システムにおけるオブジェクトが対称的な構成をしているため、RCML\_Client ノードは RCML\_Server と同じ定義をされている。また、Output Object ノードの内容は Input Object ノードと同じく定義されている。

### 4.3.3 レベル3ノード

オブジェクトの中で制御入出力情報を扱う Control Object だけは複数存在することがある。例えば首振りカメラ及び多自由度アームを備えたロボットの場合、カメラとアームの制御はお互い独立しているため、RCML ではそれぞれ一つの Control Output Object として扱うことになる。RCML ファイルはこの状況に対応するため、グローバルな制御パラメータを RCML\_ControlInput/Output ノードに預け、その下をさらに細分し、Control Object ごとに一つの RCML\_ControlInput/OutputData ノードを置く構成をとった。

RCML ノードの構造について簡単に紹介したが、ノードそれぞれの構文及び定義は RCML 1.0 の仕様[12]を参照していただきたい。

## 5. RCTP 通信プロトコル

RCML システムを実現するためには、RCML 言語とともにサーバとクライアント間の通信プロトコルが必要である。そこで RCTP の設計及び提案を行う。

### 5.1 設計方針

RCTP に関しては、以下のことを念頭において設計を行う：

- (1) 既存の通信プロトコルと最大限の親和性を持つように配慮する。
- (2) ロボット制御信号の通信処理に専念し、動画や音声などマルチメディアの送信には独自の規格を開発せず、現存のプロトコルを用いる。

- (3) 実時間制御に対応しうる通信手段として、制御動作を行っている間に余分なオーバーヘッドをできるだけ減らす工夫をする。

### 5.2 フェーズ

われわれはユーザが遠隔ロボットサーバに接続し、ロボット操作を行ってから、接続が切断するまでの動作を一回の RCTP コネクションと定義する。前述した実現イメージのシナリオから分析すると、全体は三つの段階に細分できる：

- (1) ユーザが WWW サーバから RCML ファイルを入手し、RCML ブラウザによりこの RCML ファイルを解析する。
- (2) RCTP サーバに接続を要求し、RCML ファイルで記述した情報を元に利用可能なデバイスとディスプレイを割り当てる。
- (3) 制御を行うための接続を切断するまでの遠隔操作。

1 段階目は HTTP コネクション[13][14]で実現でき、そして 2 と 3 段階目は RCTP コネクションに属する。RCTP 規格ではフェーズという言葉を用い、上記の 2 段階目をネゴシエーションフェーズと、それに 3 段階目をライブセッションフェーズと呼ぶ。HTTP コネクションと RCTP コネクションの関係は図 4 に示す。

#### 5.2.1 HTTP コネクション

RCML ファイルはロボットの姿勢や遠隔環境の変化に合わせ、内容が動的に変わるため、ロボットサーバに接続する前には HTTP コネクションを通し RCML ファイルをダウンロードしなければならない。また、トランスレータモジュールも必要に応じダウンロードすることがある。

#### 5.2.2 ネゴシエーションフェーズ

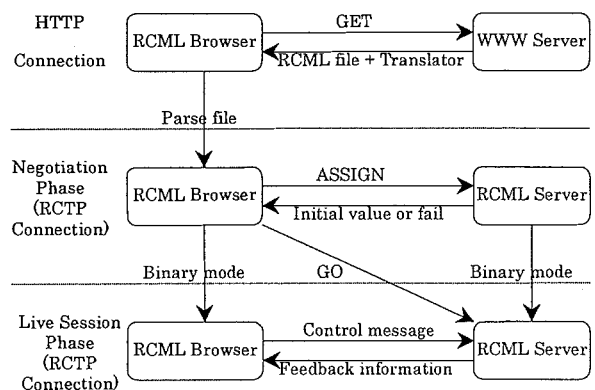


図 4 RCML システムにおける制御実行のシナリオ

Fig. 4 Scenario of one controlling session

RCMP サーバが空いている状態で、クライアントの RCML ブラウザが接続を要求することで、両方がネゴシエーションフェーズに入る。このフェーズでは RCML ブラウザがオブジェクト単位でロボットの制御権を要求し、成功の場合ではサーバが許可を下すとともにオブジェクトの ID 及び初期値を発行する。通信のやり取りは全てテキストモードで完成できると思われ、この通信規格は HTTP/1.0[13] をベースに上位互換の形で定義する。

二つの RCMP 拡張メソッドは ASSIGN 及び GO である。ASSIGN メソッドは割り当てを行い、GO メソッドでは次のフェーズへと移行するが、この際クライアントの請求およびサーバからの返答はすべて HTTP 規格に従う。なお、具体的な例を下記の例に示す：

```
(Client) ASSIGN head RCMP/1.0
(Server) 201 OK
        ID: 1
        Current Value: 30.0 10.0 5.0
(Client) ASSIGN arm RCMP/1.0
        ... 以下省略
(Client) GO RCMP/1.0
(Server) 201 OK
```

### 5.2.4 ライブセッションフェーズ

少なくとも一つのオブジェクトの制御請求が成功して、GO メソッドにサーバが許可する場合、両方はライブセッションフェーズに移行する。このフェーズの通信はバイナリモードで行われ、そのデータ形式を図5に示す。複数コマンド・データの多重送信を行い、通信のオーバーヘッドを回避するとともに、各 Input Translator/Output Translator が独自に通信手段を持たずにすむような構成をとった。また、ロボット制御情報を転送する際、下記の規格によりパケットが作成される：

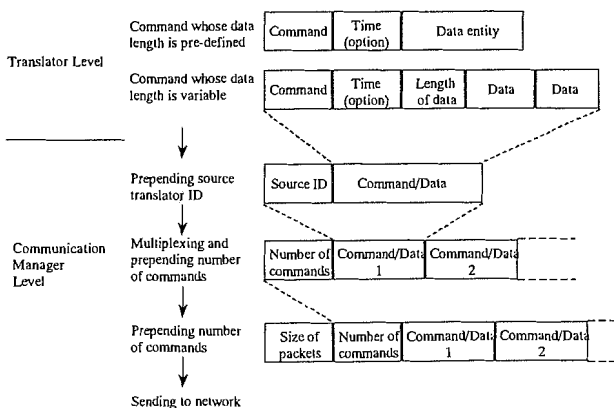


図5 RCMP のライブセッションにおけるデータ形式  
Fig. 5 Format of Data Packet

```
Control = ObjectID [TimeStamp] [DataMask] ControlData
ObjectID = OCTET
TimeStamp = 4OCTET
DataMask = 1*OCTET
ControlData = 1*OCTET
```

OCTET は 8bit からなるデータのユニットであり、 $n \times \text{OCTET}$  とは  $n$  以上の OCTET の列であることを表す。ここで ObjectID は各オブジェクトに対し RCMP サーバが割り当てた ID 番号。ライブセッションフェーズに入る時点をもととして、TimeStamp は接続した時間の長さを表す。また、DataMask は各ビットに対応するコントロール情報が送信されるか否かを示す。一パラメータを一ビットで数えて、必要なだけのバイト数が DataMask の長さになる。例えば、パラメータ数が 14 の場合、 $14 / 8$  の切り上げで DataMask が 2 バイトになる。最後に ControlData は一オブジェクトを制御するコマンドである。

TimeStamp と DataMask の使用は RCML の制御入力オブジェクトノードで記述されている。DataMask を使う場合に、ユーザが変更したパラメータのみ数値を出すのに伴い、DataMask の中でそれらのパラメータに対応するビットだけを 1 にする。例えば、10 個のパラメータを持つオブジェクトにユーザが 2 つめと 7 つめのパラメータだけに指令を下した時、実際にサーバに出したコマンドは以下ようになる：

```
Control = ObjectID 0x0042 ControlData2
ControlData7
```

この手段によって、制御情報の通信量を有効に抑えることが考えられる。なお、Control に DataMask が入らない場合には、常に全てのパラメータを通信プロトコルによって送ることになる。

## 6. 試験システムの実装

RCML システム設計の有効性を検証するため、我々は首振りカメラ (Sony 社製 EVI-D30) を載せた自律移動ロボットを制御対象として、実装システムの開発を行った (図6) [15]。

サーバ側には一台のノート PC (Pentium 100MHz, メモリ 32MB) でロボット及びカメラの操縦を行う。運動指令として、ロボットには回転及び前進、カメラには上下、左右及びズームの 3 自由度を実装した。また、ロボットの位置センサからの位置情報及びカメラからのビデオストリームは感覚情報としてクライアント側へ送り込む。

クライアント側でも PC (Pentium MMX 200MHz, メモリ 32MB) を用いて操作する。入力装置は汎用 GUI を実装した他、Joystick や 6 自由度の 3 次元位置計測装

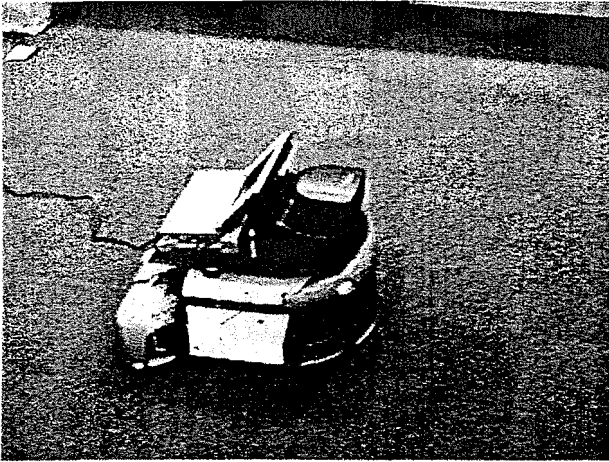


図6 実装システムにおけるサーバの様子  
Fig. 6 Photo of the controlling robot

置 ADL-1 を付けている。RCML ファイルの VRML モデル部分は Sony の Community Place Browser で表現し、ビデオストリームの伝送は CUSeeMe といった既存のソフトを利用するため、開発を容易に行えた。

制御対象のロボットが移動するため、サーバ側とクライアント側の通信は無線 LAN(帯域幅 2Mbps)で行う。システム全体の構成を図7に示す。

実際制御実行の手順は、HTTP コネクションを省いて、前述した実現イメージの通りで実現した。ただし、クライアントがサーバに接続する際、RCML ブラウザは利用可能な汎用 GUI および物理デバイスをユーザに提示し、制御対象のオブジェクト毎に制御入力装置を選択してもらう。実装システムでは Joystick 及び ADL-1 の他、ロボット及びカメラにそれぞれ三つの GUI を提供する。ユーザがすべての制御入出力装置の対応を決めて、セットボタンを押したら初めてネゴシエーションフェーズが始まる。この仕組みでは、サーバが予めロボットの特性に適合する GUI などのトランスレータをいくつか用意し、ユーザは自分の操作環境に合わせた制御手段を選択できるようになる。また実装システムでは、制御入力オブジェクトのトランスレータとし

て書いた Java プログラムを VRML のノードに結び付けることにより、実空間及びバーチャル空間に発生するロボットの変化を同調させる方式で開発した。クライアントプログラムの様子は図8で示す。

制御の流れとして、クライアントから送信されるコマンドをサーバプログラムが解釈し、移動ロボットと首振りカメラを制御して、また制御結果を読み取ってからクライアントに送信する。クライアントでは、既に制御開始時に得られている VRML データを制御結果に応じて更新し、バーチャル環境としてクライアントに概観として提示し、サーバから送られて来る動画像で詳細な状況を知ることができる。

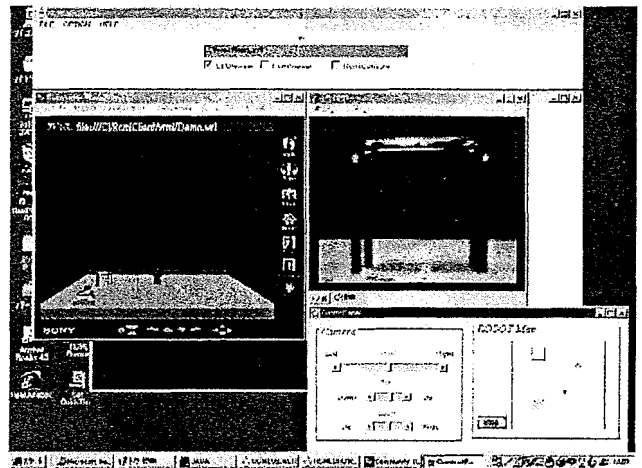


図8：RCML ブラウザの操作画面  
Fig. 8 Screen of operating RCML browser

## 7. 考察と検討

前章のシステムで簡単な障害物越し走行実験を行った。実験の概要としては、縦横5メートルの床に三つの障害物を置いて、遠隔環境を模擬したバーチャル空間及びカメラから撮ったビデオをユーザに提示しながら、ロボットを遠隔操縦で障害物を回避しながら目的の位置へ辿り着けるのである(図9)。すべてのユーザーインターフェースの組み合わせで操作を行ったが、ユーザが VR モデルを参照しながら操作した結果、全部障害物に衝突することなく目的地に到達したことが観察された。

また、操作の際には RCTP プロトコルで規定した多重送信を継続的にされているため、例えばカメラの向きを変えながらロボットを回転させるという同時操作やロボットが移動する際に緊急停止コマンドで前の指令を無効にして停止させるなど実時間制御に要する操作体系を実現した。

なお、今回の実装で明らかにされた問題点として、VRML 仕様で定義した Java クラスの機能制限により、3

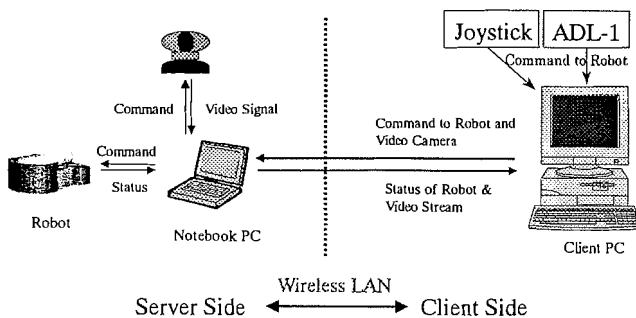


図7：実装システムの構成  
Fig. 7 Structure of Experimental Implementation

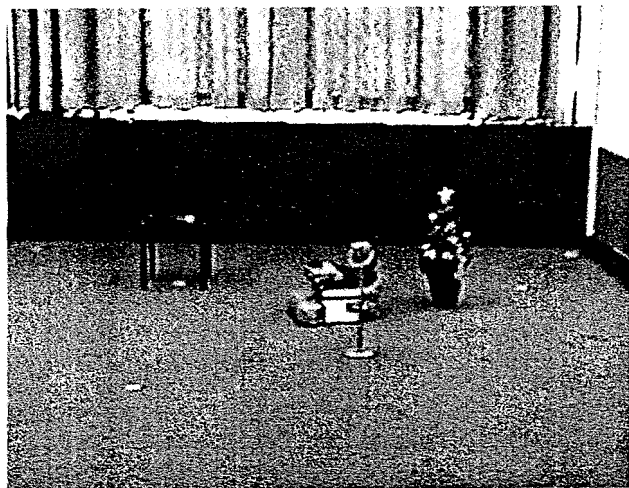


図9：障害物越し走行実験の様子  
Fig. 9 Photo of robot crossing obstacles

レベルにわたる RCML 拡張ノードの木構造の読み取りや設定は VRML ブラウザに結び付ける Java プログラムでは届かないところがあるため、RCML ファイルのパーシングを VRML ブラウザに任せることが出来ず、自前のパーサーを開発せざるを得なかった。また、RCML 言語の今後の拡張にしても VRML の文法に制限される懸念がある。

RCTP の部分では、実装システムで用いる無線 LAN の帯域幅が狭いため、ビデオストリームを入れた場合ではロボットの制御遅延が平均 0.5 秒から 1.2 秒と長引くことが今回の実装で分かった。実時間制御を目指すプロトコルとしてはこれから信頼性重視あるいはリアルタイム性重視両方の通信方法の使い分けを実現すべきであると考えられる。

## 8. まとめ

アールキューブネットワークの実現へ向けて、RCML システムの機能構成から始め、アールキューブ操作言語 RCML とその通信プロトコル RCTP の仕様の設計、および仕様の有効性を検証するための遠隔操作システムの実装を行った。RCML システムは、VRML や HTTP など既存の標準をベースにし、手軽に構築そして利用できる標準的な手法を開発する設計思想を念頭に置いて開発された。またその思想に基づき、RCML および RCTP の設計開発も極力既存の標準と下位互換となるように行った。RCML システムのもう一つ重要な特徴では、オブジェクトという機能構成で入出力装置に汎用性をもつことである。今回の実装は設計通りに構築し、その汎用性の持つ標準的な手法の可能性を示した。

今後は、実装による経験をフィードバックし、仕様の改善を検討し、評価システムの構築及び評価実験の実施を行う予定である。また VRML の次期バージョンが

XML ベースと検討されていることからしても、RCML を XML ベースに移行することも検討すべきである。

## 参考文献

- [1] 通産省アールキューブ研究会：“アールキューブ”，日刊工業新聞社 (1996)
- [2] 舘：“バーチャルリアリティとロボティクス”，日本ロボット学会誌 Vol.15 No.4, pp.32-35 (1997)
- [3] Y. Yanagida, N. Kawakami, and S. Tachi, Development of R-Cubed Manipulation Language - Access Real Worlds Over the Network, The 7th International Conference on Artificial Reality and Tele-existence, pp.159-164 (1997)
- [4] M. Pesce: “VRML Browsing and Building Cyber space”, Macmillan Publishing (1995)
- [5] <http://www.vrml.org/technicalinfo/specifications/vrml97/index.htm>: “VRML97 Specification, ISO-IEC 14772-11997”, Web3D Consortium (1997)
- [6] R. Simmons: “Xavier: An Autonomous Mobile Robot on The Web,” Preprints of IROS’ 98 Workshop ‘Robots on the Web,’ pp.43-47 (1988)
- [7] P. Saucy and F. Mondada: “KhepOnTheWeb: One Year of Access to a Mobile Robot on the Internet,” Preprints of IROS’ 98 Workshop ‘Robots on the Web,’ pp.23-29 (1998)
- [8] S.B.Goldberg, et al: “DIGIMUSE: A interactive telerobotic system for remote viewing of three-dimensional art objects,” Preprints of IROS’ 98 Workshop ‘Robots on the Web,’ pp55-59 (1998)
- [9] 音田, 他: “ネットワークロボティクスの可能性—その 3: 6 自由度アームを用いた仮想遠隔環境操作システム—”, 第 14 回日本ロボット学会学術講演会予稿集, pp.813-814 (1996)
- [10] 原, 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)
- [11] W. C. Teng, A. Nukuzuma, N. Kawakami, Y. Yanagida, and S. Tachi: “Development of R-Cubed Manipulation Language - The Specification of RCML and RCTP -,” Proc. of the 8th International Conference on Artificial Reality and Tele-existence, pp.156-162 (1998).
- [12] <http://www.star.t.u-tokyo.ac.jp/projects/RCML>
- [13] Berners-Lee, T., Fielding, R. and H. Frystyk. “Hypertext Transfer Protocol -- HTTP/1.0,” RFC 1945, MIT/LCS, UC Irvine (1996)
- [14] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,



Berners-Lee, T., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, UC Irvine, Digital Equipment Corporation, M.I.T. (1997)

- [15] W. C. Teng, et al.: "Development of R-Cubed Manipulation Language -Implementation and Evaluation to a RCML System-", Proc. of the 9th International Conference on Artificial Reality and Telexistence, pp.156-162 (1999).

(2000年1月13日受付)