

Extended Protocol

(1) Registration

クライアントのレジストレーションを行う。

- ・ビューワのタイプ、ビューワのニックネームを伝達すると、クライアントID(CID)が付与され、返される。
- ・レジストレーションを行わないクライアントは、拡張ビューワとは見なされず、以降の通信で拡張プロトコルは使われない。

CLIENT		SERVER	Comments
<div style="border: 1px solid black; padding: 2px;"> <b>VV_REGISTER</b>  length  TYPE  NAME  <b>HEADER_NULL</b> </div>	→		(TYPE=1:バージョン番号) (NAME=STRING,任意の識別子,lengthまで)
	←	<div style="border: 1px solid black; padding: 2px;"> <b>VV_ACKNOWLEDGE</b>  length(=8)  STATUS  CID  <b>HEADER_NULL</b> </div>	(0:成功,1:NAMEの衝突,2:クライアント数あふれ,3:その他のエラー) (失敗の場合0,それ以外の場合、 1以上の整数)

(2) Query

クライアントからサーバに他のクライアントに関する情報を問い合わせる。以下の2種類の問い合わせが可能である。

- 「他の接続クライアントの一覧の問い合わせ」(QTYPE=1)
- 「接続クライアントの情報に関する問い合わせ」(QTYPE=2)

CLIENT		SERVER	Comments
<div style="border: 1px solid black; padding: 2px;"> <b>VV_QUERY</b>  length(=4)  QTYPE(=1)  <b>HEADER_NULL</b> </div>	→		QTYPE=1:接続クライアント一覧
	←	<div style="border: 1px solid black; padding: 2px;"> <b>VV_ACKNOWLEDGE</b>  length  STATUS  CID(1)  CID(2)  CID(3)  ...  <b>HEADER_NULL</b> </div>	QTYPE=1の場合 (0:成功) (length-1)繰返し
<div style="border: 1px solid black; padding: 2px;"> <b>VV_QUERY</b>  length(=8)  QTYPE(=2)  CID  <b>HEADER_NULL</b> </div>	→		QTYPE=2:接続クライアントのTYPE,NAME
	←	<div style="border: 1px solid black; padding: 2px;"> <b>VV_ACKNOWLEDGE</b>  length  STATUS  TYPE  NAME  <b>HEADER_NULL</b> </div>	(0:成功,1:該当クライアントなし) (NAME=STRING,任意の識別子,lengthまで)

(3) Message

クライアントから他クライアントに対してメッセージを伝達する。クライアントのIDを指定する場合と、ブロードキャストの場合がある。

- ・ブロードキャストの場合にも、メッセージ発出本にはメッセージは伝達されない。
- ・クライアントID指定の場合には、成功したかどうかを示すVV\_ACKNOWLEDGEが返される。ブロードキャストの場合には何も返されない。

CLIENT		SERVER	Comments
<div style="border: 1px solid black; padding: 2px;"> <b>VV_SEND</b>  length  SOURCE-CID  TARGET-CID  Container  <small>(Depends on Container type)</small>  <div style="border: 1px solid black; padding: 2px;"> <b>HEADER_NULL</b> </div> </div>	→		発信元CID 宛先CID,0=BROADCAST
	←	<div style="border: 1px solid black; padding: 2px;"> <b>VV_ACKNOWLEDGE</b>  length(=4)  STATUS  <div style="border: 1px solid black; padding: 2px;"> <b>HEADER_NULL</b> </div> </div>	(CID=自分の場合, BROADCASTの場合返答不要)  (0:成功,1:該当CIDなし)

(4) Attention

サーバが中継したメッセージを伝達する。

- ・自分宛のメッセージの場合にはVV\_ACKNOWLEDGEメッセージを返送する必要がある。ブロードキャストメッセージの場合には、返送は不要

CLIENT		SERVER	Comments
	←	<div style="border: 1px solid black; padding: 2px;"> <b>VV_ATTENTION</b>  length  SOURCE-CID  TARGET-CID  Container  <small>(Depends on Container type)</small>  <div style="border: 1px solid black; padding: 2px;"> <b>HEADER_NULL</b> </div> </div>	発信元CID 宛先CID,0=BROADCAST
<div style="border: 1px solid black; padding: 2px;"> <b>VV_ACKNOWLEDGE</b>  length(=4)  STATUS  <div style="border: 1px solid black; padding: 2px;"> <b>HEADER_NULL</b> </div> </div>	→		(CID=自分の場合, BROADCASTの場合返答不要)  (0:成功)

Container

番号	メッセージ	意味	フォーマット	長さ
0	CONTAINER_NULL	containerの終わり	CONTAINER_NULL	4
1	CONTAINER_DECLAR	クライアントの宣言	CONTAINER_DECLARE len message	8+メッセージ長 byte[]
2	CONTAINER_CURPOS	カーソル位置の指定	CONTAINER_CURPOS pos_x pos_y	12
3	CONTAINER_SCREEN	画面の中心位置,スケールの指定	CONTAINER_SCREEN center_x center_y scale	16
4	CONTAINER_DRPOLY	書き込みポリゴンの指定	CONTAINER_DRPOLY #point pos_x1 pos_y1 ..	8+ポイント数*8
5	CONTAINER_PUTICN	アイコンの設置	CONTAINER_PUTICN icon_id pos_x pos_y	16
6	CONTAINER_MESSAGE	任意文字列の送信	CONTAINER_MESSAGE len message	8+メッセージ長 byte[]
7	CONTAINER_HLIGHT	オブジェクトのハイライト	CONTAINER_HLIGHT OID FLAG len message	16+メッセージ長 byte[]



# リアルタイム災害情報共有に向けた RoboCupRescue システムの拡張

桑田喜隆<sup>1</sup>, 神成淳司<sup>2</sup>

An extension of RoboCupRescue system towards the real-time sharing of disaster information

Yoshitaka KUWATA and Atsushi SHINJO

**Abstract:** We propose an extended architecture and a communication protocol for RoboCupRescue simulation system, with which make GIS-based viewers communicate each other. We built a prototype viewer named “Collaborative Viewer, (CV)” based on the protocol. As the extended communication protocol is a superset of standard protocol, CV can make use of all of components developed in RoboCupRescue simulation project.

**Key Words:** RoboCupRescue Project, Communication Protocol, Geographic Information System (GIS)

## 1. はじめに

RoboCupRescue シミュレーションプロジェクト<sup>1)</sup>は複数の災害シミュレータ及び救助活動を行うエージェントを含む統合シミュレータを提供することを目標の一つとしている。作成されたシステムの実災害への応用として、以下の形態が可能であると考えられる。

- (1) 事前の災害対応計画の立案検証
- (2) 災害対応時の情報収集・意思決定の支援
- (3) 災害救助ロボットの制御戦略の検討および検証

筆者等は(2)の応用を想定して地理情報システム(GIS)を基礎とした情報共有と、収集した情報を用いたシミュレーションによる意思決定支援に関する検討<sup>2)</sup>を行っている。

本稿では RoboCupRescue シミュレータを拡張することで、災害現場などと実時間で GIS 上の情報を交換する仕組みを提案する。

## 2. 拡張アーキテクチャの設計

既存のソフトウェアの有効活用の観点から、既存のシステムの変更は最小限になるようなアーキテクチャを採用した。拡張アーキテクチャを Fig. 1 に示す。

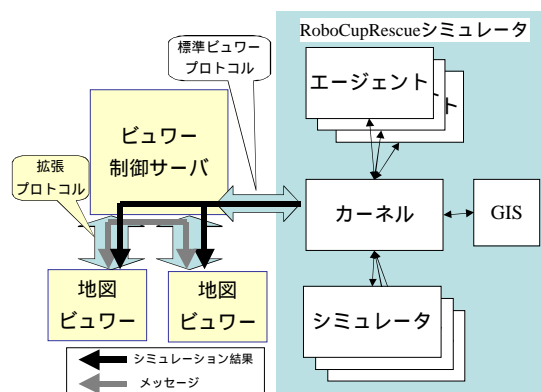


Fig. 1 Extended RoboCupRescue Architecture

災害情報等の情報共有に、地図を基礎とした複数の地図ビューアーを利用することとし、地図ビューアー間の通信制御にビューアー制御サーバを導入した。ビューアー制御サ

ーバを中心としたクライアント-サーバ型のシステム構成となっている。ビューアー制御サーバはカーネルにレスキューシステムのビューアーの一つとして接続する。これにより、シミュレーション結果がカーネルよりビューアー制御サーバを経由して各地図ビューアーに配信される。

現在の RoboCupRescue シミュレータにはビューアーを通じてカーネルを制御する機能が実装されていない。このため、収集した情報を基にしたシミュレーション用データの変更および、シミュレーションの進行制御を行うことは出来ない。しかしながら、同じ設定に基づくシミュレーション結果を共有しながら情報の伝達を行うことが可能であり、特にシミュレーション結果を示しながら、現場へ指示を出す等の利用が期待される。

## 3. 通信プロトコル

アーキテクチャの拡張に伴い、地図ビューアーとビューアー制御サーバ間の通信プロトコルの設計を行った。拡張プロトコルは以下の特徴を有する。

- (1) 標準ビューアープロトコルに重畳可能  
拡張プロトコルは標準プロトコルの拡張として実装されているため、拡張ビューアーと既存ビューアーの混在が可能である。
- (2) 通信プロトコル層から通信内容が独立  
通信内容は上位層で独自に実装が可能。このため、異なる仕様の拡張ビューアーが同じ通信プロトコル上に混在可能である。
- (3) クライアントの状態をサーバで管理  
サーバ側でクライアントの状態を管理することで、最適な情報の提供が可能となる。
- (4) ブロードキャストのサポート  
サーバ経由で複数のクライアントに同時にメッセージを送ることが可能である。

### 3.1 状態遷移

拡張プロトコル上でのビューアーの通信状態の遷移を Fig.2 に示す。通信状態は以下の3状態を持つ。

- (1) 初期状態

<sup>1</sup> (株) NTTデータ 公共システム事業本部

<sup>2</sup> 国際情報科学芸術アカデミー

ビューアー起動直後の状態 .

(2) 標準ビューアー状態

サーバに接続を行った状態 . この状態で , 標準プロトコルのみが使われる .

(3) 拡張ビューアー状態

サーバに拡張ビューアーとして登録処理を行った状態 . 標準プロトコルに加えて , 拡張プロトコルが使われる .

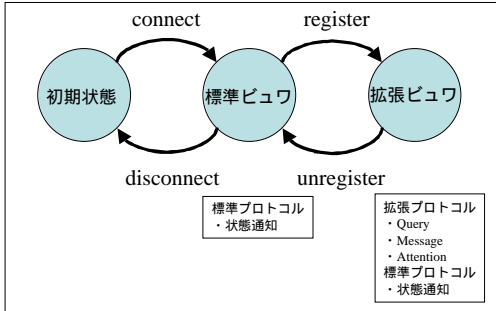


Fig. 2 State diagram and valid messages

3.2 制御メッセージ

拡張プロトコルでは以下の制御メッセージが使われる .

- (1) Register / Unregister : 状態を遷移する .
- (2) Query : 他のビューアーの状態を問い合わせる .
- (3) Message : 他のビューアーに通信を送る .
- (4) Attention : 他のビューアーからメッセージを受け取る .

4. Collaborative Viewer での実装

提案したアーキテクチャの検証のために , 拡張アーキテクチャおよび拡張プロトコルを組み込んだビューアー (Collaborative Viewer, CV)を開発した . CV は複数のビューアー間の単純な画面共有を目的としているため , コンテンツは単純なものに限った . CV では , カーソルの位置共有 , 画面の同期 , 手書き情報の共有機能を実現した . 画面イメージを Fig3 に示す .

実装したコンテンツ形式および内容を Table1 に示す .

Table 1. Message format of Collaborative Viewer

Message (Mnemonic)	意味	情報種別
V2_DECLAR	クライアント情報の登録	クライアント種別, ニックネーム等
V2_CURPOS	カーソル位置の指定	座標値
V2_SCREEN	画面の中心位置, スケールの指定	座標値, スケール
V2_DRPOLY	手書き情報位置	座標値列
V2_PUTICN	伝達位置情報	座標値 (アイコンなど)
V2_MESSAGE	メッセージ送信	文字列

5. まとめと今後の課題

地図ビューアー同士で通信が可能な拡張アーキテクチャおよび拡張プロトコル仕様を提案した . 提案した仕様に基づき既存のビューアーを拡張して Collaborative Viewerとして実装した . 拡張ビューアーとビューアー制御サーバはクライアントサーバ形式であるため , トラフィック量が増加した場合サーバに負荷が集中することが予想される . また実用化に向けては通信制御の効率化 , ユーザ管理 , セキュリティ対策などの実装が必要になる .

なお , 試作したビューアーは地図を基礎としたユーザインタフェースの評価実験<sup>3)</sup>にも利用された .

参考文献

- 1) 田所論, 北野宏明監修: ロボカップレスキュー緊急大規模災害救助への挑戦, 共立出版(2000)
- 2) Yoshitaka Kuwata et al.: Design of RoboCup-Rescue Viewers – Toward a Real World Emergency System, The 4<sup>th</sup> Intl. Workshop on RoboCup (2000)
- 3) 桑田喜隆, 神成淳司, 吉田茂樹, 大谷尚通, 井上潮: 現場活動支援のための地理情報に基づく実時間情報共有システム, 情報処理学会 GN 研究会

ビューアー制御サーバ



地図ビューアー ( 1 )

地図ビューアー ( 2 )

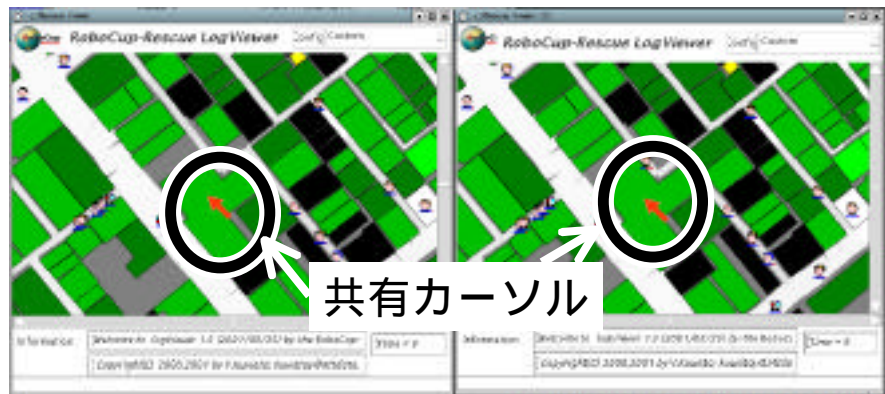


Fig. 3 Screen Image of Collaborative Viewer