
logViewer 1.5.1および viewerServer1.1の新機能

平成14年3月9日

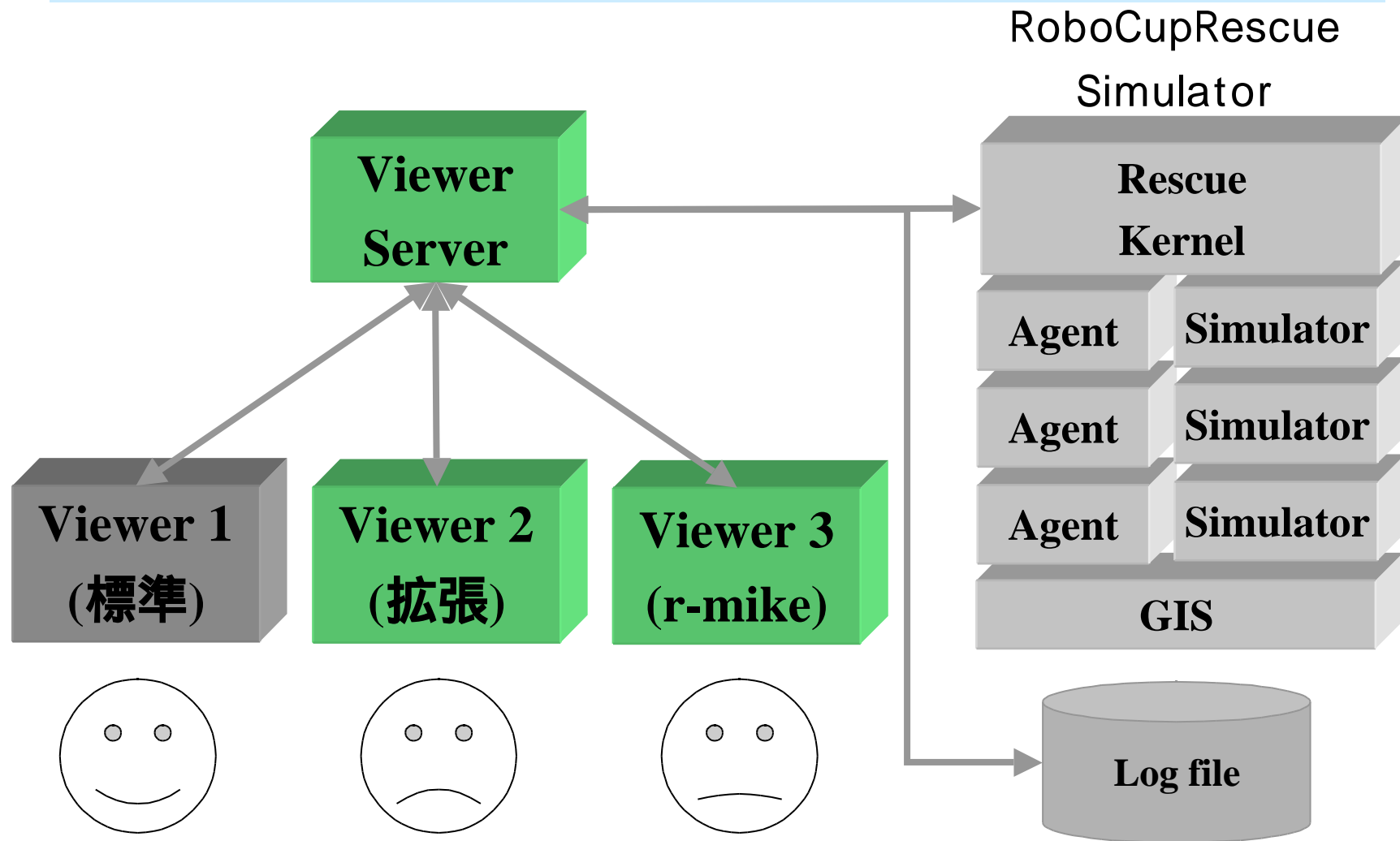
(株)NTTデータ

桑田 喜隆

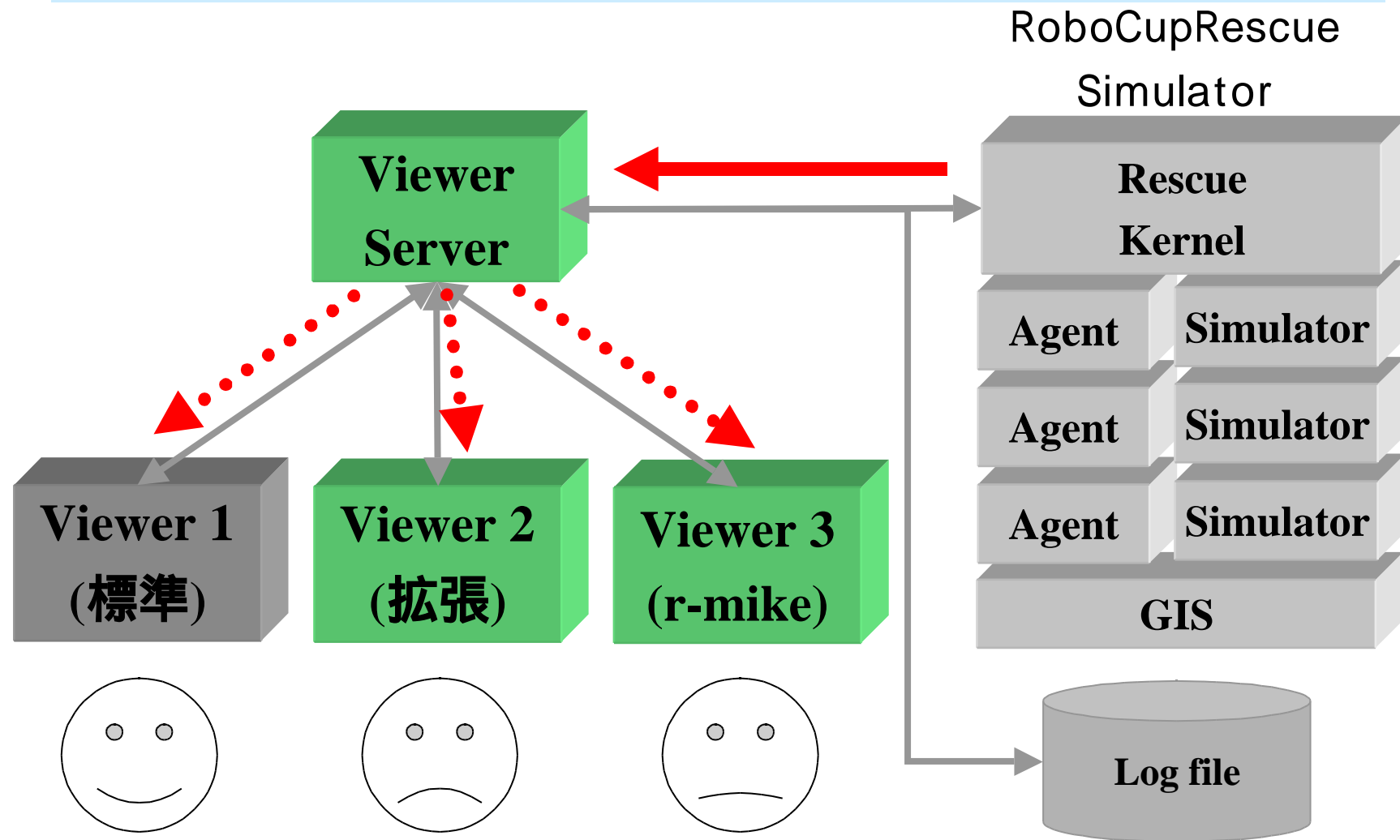
機能概要

- **Viewerのサブシステム化**
 - RescueKernel = サーバ(viewerServer) = クライアント(viewer)
 - 異種ビューワのサポート (標準, 拡張, Rescue-mike)
- **Viewer間通信のサポート (拡張logViewerモード)**
 - 手書き情報, 視点位置などの共有
 - 通信レイヤとコンテンツレイヤの分離
 - 既存プロトコルとの互換性を維持
- **viewerServerによる時刻管理機能**
 - Kernel-Server通信とServer-viewer通信の分離/非同期化
 - 両通信機能を実現するため処理をMulti-thread化
 - Viewer毎に異なる時刻の状況を管理
 - Viewer間の時間同期のためのマスタークロックの導入: Server Time
- **プログラム管理・配付方法の変更**
 - 1パッケージ化 (同じJARファイルで配付)
 - ソースコードの一部共有化

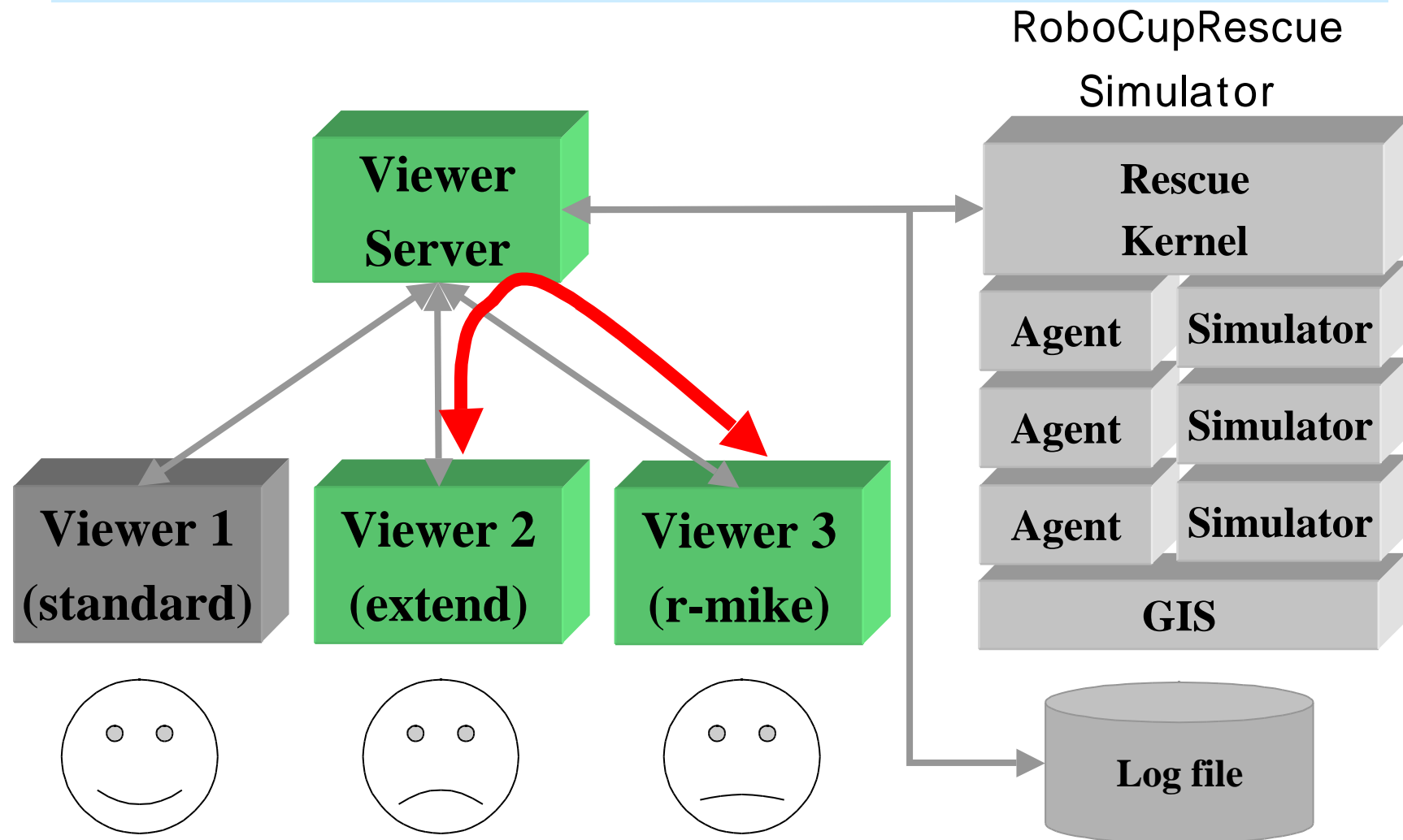
Viewer Sub-Systemの構造



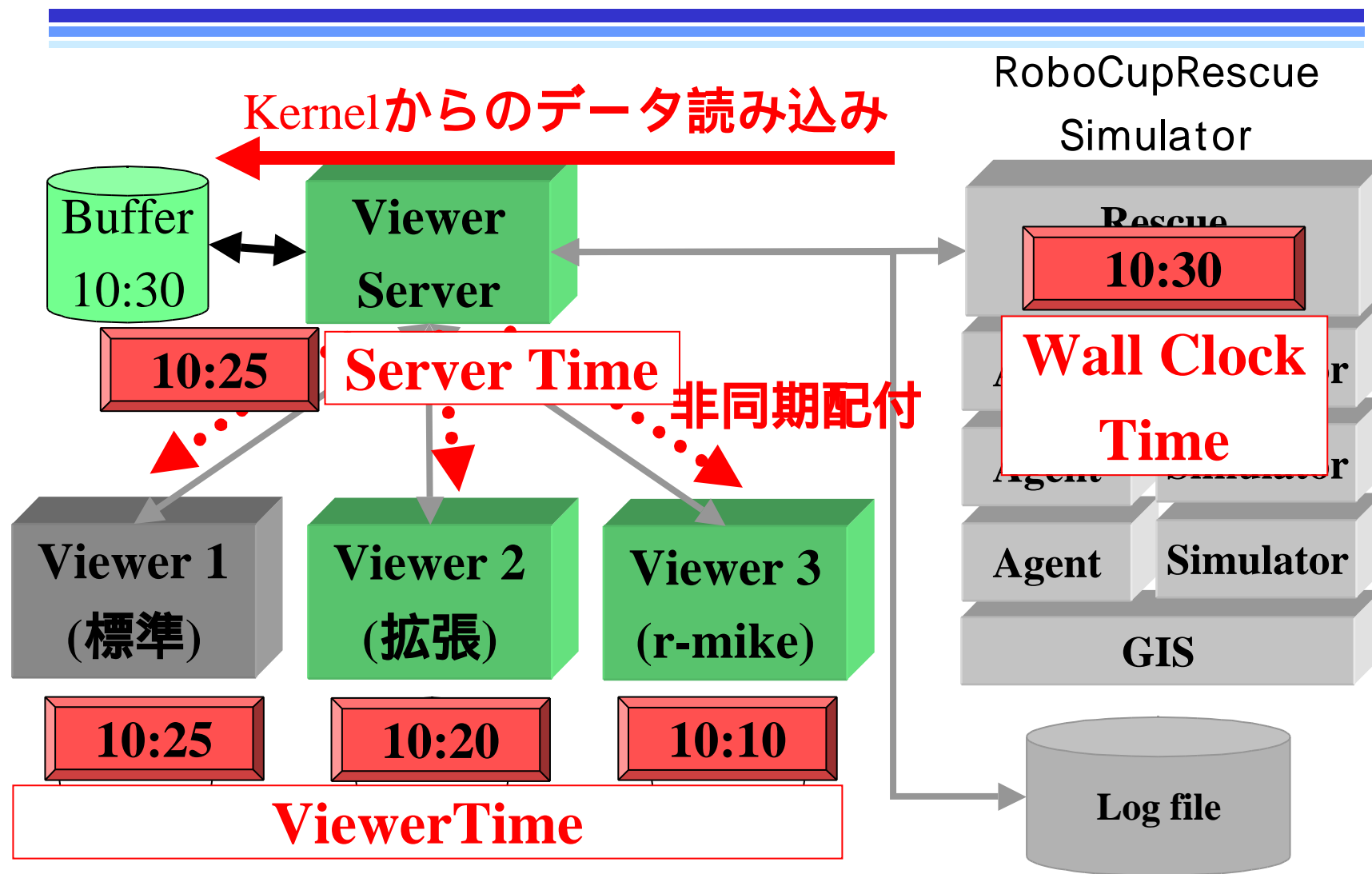
データの流れ：シミュレーション結果



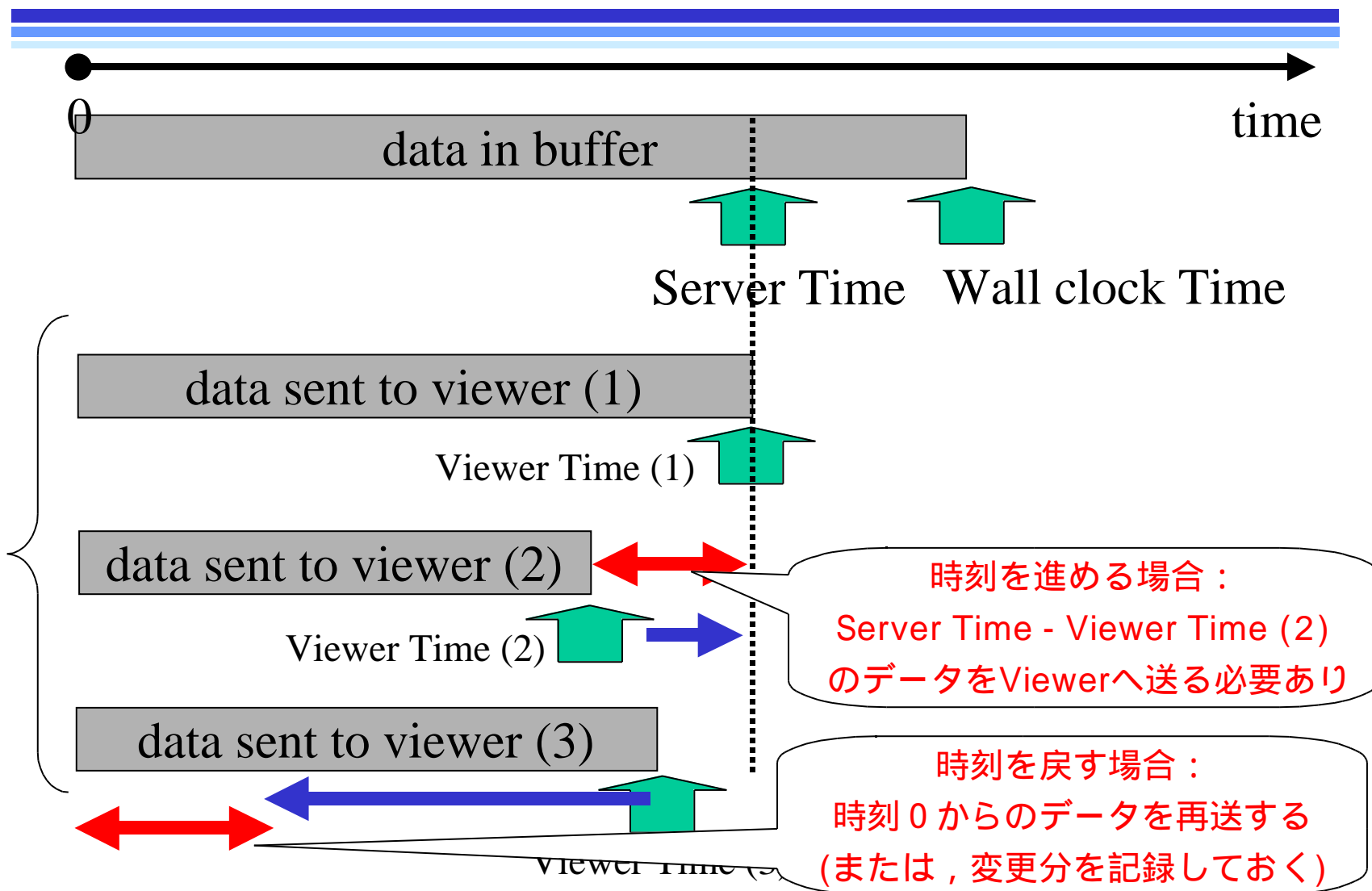
データの流れ：共有情報



時刻管理機能



時間管理処理アルゴリズム



起動の方法（例）

- **viewerServerの起動**

ログファイルを開く (viewer側は6060番ポートを指定)

```
% java -Djava.class.path=logViewer.jar viewerServer.logController -f logfile -P6060
```

localhostのカーネル6000番ポートで接続する場合

```
% java -Djava.class.path=logViewer.jar viewerServer.logController -h localhost -p 6000 -P6060
```

- **logViewerの起動**

標準ビューワーとしてviewerServerに6060番ポートで接続する

```
% java -Djava.class.path=logViewer.jar viewer.Map -h localhost -p 6060
```

拡張ビューワーとしてviewerServerに6060番ポートで接続する

```
% java -Djava.class.path=logViewer.jar viewer.Map -e -h localhost -p 6060
```

viewerServerのスクリーンイメージ

RoboCupRescue Project: viewerServer 1.1

Master Clock

Kernel Time 159 Server Time 159 << || > >>

Viewer	Status	Nickname	Mode	Time	Control
localhost.localdomain/127.0.0.1,	Ready	Extended Viewer	On line	159	<< > >>
localhost.localdomain/127.0.0.1,	Ready	Extended Viewer	On line	159	<< > >>
viewer 3	Disconnected	None	On line	0	<< > >>
viewer 4	Disconnected	None	On line	0	<< > >>
viewer 5	Disconnected	None	On line	0	<< > >>
viewer 6	Disconnected	None	On line	0	<< > >>
viewer 7	Disconnected	None	On line	0	<< > >>
viewer 8	Disconnected	None	On line	0	<< > >>

Rescue kernelから
受け取ったデータ

Server Timeの制御用
ボタン

個別のViewerTime
の制御用ボタン
(未実装)

スクリーンイメージ

The screenshot displays the 'RoboCupRescue Project: viewerServer 1.1' interface. At the top, it shows 'Master Clock' with 'Kernel Time' and 'Server Time' both set to 159. Below this is a table listing viewer connections:

Viewer	Status	Nickname	Mode	Time	Control
localhost.localdomain/127.0.0.1,	Ready	Extended Viewer	On line	159	<< >>
localhost.localdomain/127.0.0.1,	Ready	Extended Viewer	On line	159	<< >>
viewer 3	Disconnected	None	On line	0	<< >>
viewer 4	Disconnected	None	On line	0	<< >>
viewer 5	Disconnected	None	On line	0	<< >>
viewer 6	Disconnected	None	On line	0	<< >>
viewer 7	Disconnected	None	On line	0	<< >>
viewer 8	Disconnected	None	On line	0	<< >>

Below the table are two windows titled 'RoboCup-Rescue LogViewer'. Each window displays a grid of colored rectangles (green, yellow, red, blue) representing log data. A callout bubble points to the viewerServer interface, and two other callout bubbles point to the logViewer windows.

議論点(1)

- Viewerでの処理アルゴリズムの注意点
 - 時間シーケンスに従った入力データを仮定できない
 - 再実行可能:Reentrant化
 - 表示時刻以前のデータを受け取った場合の処理は？
 - 最適なバッファリング処理が必要
 - 統計計算，スコア計算が正しく行われること
- viewerServerでのアルゴリズムの最適化
 - Rescue Kernelからは時刻変化分()を受け取る .
 - 時刻逆変化分(-)分を計算する
 - データ量を最小にする
 - $\times n$ を使うか， $0 + \times m$ を使うか

議論点(2)

- 時刻制御の方法論
 - 現在はviewerServerでViewerTimeを制御
 - 時刻管理はViewer側のリクエスト？
 - Viewer側リクエストのよる他ビューワとの同期
 - 他Viewerの状態をどのように知るか
 - Viewer同士の制御権は誰が握るか
- Rescue-kernelの時刻制御（？）
 - 時刻を戻しても，同じシミュレーション結果を見られるだけ
 - ある時点からのシミュレーションのリスタート
What-ifの実現
 - Re-entrant性