HLA をベースとした並列分散シミュレーションの実現 —Data Distribution Management 機能の実現方式—

2H－6

渡部修介† 古市昌一† 水野政治† 和泉秀幸† 尾崎敦夫† 徳本修一† 佐藤啓紀‡

†三菱電機（株），‡三菱電機システムウェア（株）

1. はじめに

HLA(High Level Architecture)[1]は，分散計算機環境上で多数の異機種シミュレータを統合するための，分散シミュレーション統合基盤である．米国防総省下の DMSO(Defence Modeling & Simulation Office)が 1995 年に提案し，SISO(Simulation Interoperability Standardization Organization)が中心となって 2000 年の IEEE 標準化に向けて活動を行っている．

我々は，HLA による分散シミュレーションの実行時に必須となるソフトウェアである RTI(Run Time Infrastructure)の実現と実用化を目指した研究開発を行っており，その一環として eRTI(experimental RTI)[2]を開発している．本稿では，HLA で規定されているシミュレータ間通信フィルタリング機能の一種である Data Distribution Management(DDM)の，eRTI での実現方式について述べる．

2. RTI による通信フィルタリング

HLA では，複数のフェデレート(HLA では分散環境下の各シミュレータをフェデレートと呼ぶ)間でのデータ交換の際に，不必要なデータ通信を減らして通信トラフィックを低減するためのフィルタリング機能が，RTI の機能として規定されている．

代表的なフィルタリング機能として，Declaration Management(DM)を挙げることができる．DM では，シミュレートされるオブジェクト(例えば個々の航空機や船舶)やインタラクション(例えば交信などのイベント)のクラスを用いて送信フェデレート(Publisher)と受信フェデレート(Subscriber)のデータ交換定義を行うことにより，クラスレベルで必要とされないデータ通信の削減を実現する．

Implementation of Parallel and Distributed Simulation System based on HLA: Implementation of Data Distribution Management

S.Watanabe†,M.Furuichi†,M.Mizuno†,H.Izumi†,A.Ozaki†,
S.Tokumoto†,H.Sato‡

†Mitsubishi Electric Corporation, ‡Mitsubishi Electric Systemware Corporation

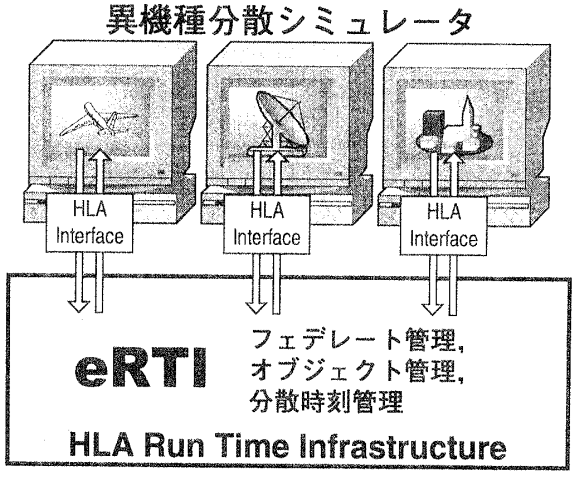

Fig.1: eRTI を用いた HLA 分散シミュレーション

3. DDM の概略

DM がオブジェクトやインタラクションのクラスレベルでの通信フィルタリングを実現する対して，DDM ではインスタンスレベルでの通信フィルタリングを実現する．DDM では，全フェデレートが共有する **Routing Space** という多次元の空間定義と，Routing Space 上に各フェデレートが実行中に作成してオブジェクトやインタラクションのインスタンスと関連付ける **Region** という領域定義を用いて動的なデータ交換制御を行う．

Fig.2 に示す利用例を用いて DDM の概要を説明する．Publisher *Fa* は，自身がシミュレートするオブジェクトインスタンス *ship1*, *ship2* に対して，Routing Space (*Game Field*)上に生成した Region *U1*, *U2* をそれぞれ関連付けることにより，*Game Field* 上での *ship1*, *ship2* の位置を保証する．一方，Subscriber *Fb* は同じ *Game Field* 上に自身の興味領域である Region *S1* を生成し，この領域に関連付けたデータ受信を宣言する．

データ交換制御を行う RTI では，これらの各 Region 領域のオーバラップ判定を行い，領域の重なる関係となった Publisher と Subscriber についてデータ通信を実行する．この例では，*U1* と *S1* がオーバラップしていることから *ship1* のデータが実際に交換され，他の Region とのオーバラップが無い *U2* と関連付けられた

*ship2*のデータは交換されないこととなる．

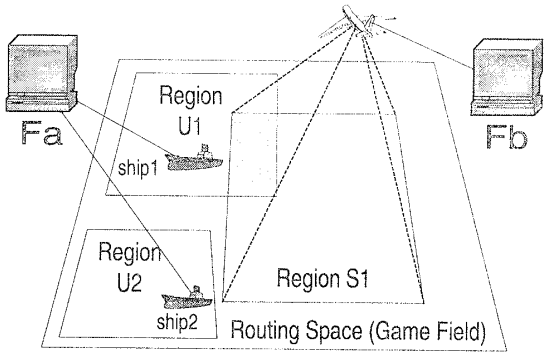

Fig.2: DDM の利用例

4. eRTI における DDM 実現方式の検討

4.1. 実現モデル

eRTI は，RTI が分散シミュレーション管理に必要とする機能を，分散計算機環境下の単一ノードに集約した「集中型」モデルを採用している．従って，各フェデレートが生成する Region のオーバラップ判定は，eRTI が実行されている単独ノード上で実行する．Fig.2 の例に対応した実現例を Fig.3 に示す．

4.2. Region の生成，更新，削除処理

これらの処理はフェデレートが起動するが，Region の実体は上で述べた様に eRTI 側に存在する．そこで，フェデレートが Region を生成した際の戻り値として Region の Proxy オブジェクトを返し，フェデレートは Region Proxy を用いて更新，削除処理を起動するような処理方式を採用する．Fig.4 に，Region Proxy による各処理の概要を示す．

4.3. オーバラップ判定処理

各 Region の Dimension(次元)ごとに設定された下限値と上限値を用いてオーバラップを判定する．オーバラップ判定処理の開始は，Region の一つが更新されたタイミングとなる．

5. おわりに

DDM で Publisher と Subscriber 間の不必要なデータ交換を最小にするためには，各 Region の領域を真に必要な大きさまで小さくすればよい．しかし，関連付けられる Region の領域を必要最小限にしてしまうと頻繁に Region 位置情報を更新しなければならず，その更新に必要な通信量とオーバラップ再検出処理の負荷により，全体の処理性能の低下が懸念される．

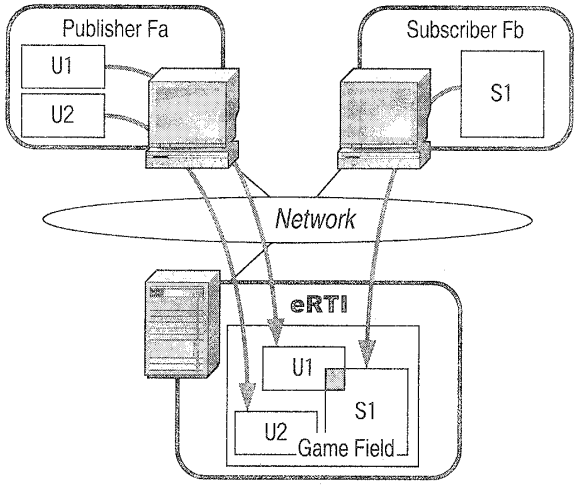

Fig.3: eRTI における実現例

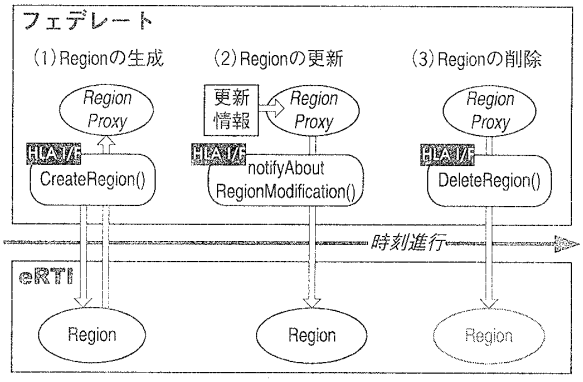

Fig.4: Region の生成，更新，削除

今後は，今回の実装に対する性能評価を行い，eRTI による Region 更新処理とオーバラップ検出処理の高速化を検討する．これによって DDM によるフィルタリング精度を向上し，より大規模な分散シミュレーションへの適用が可能な RTI の実現を目指す．

参考文献

[1] Department of Defence: High Level Architecture Interface Specification Version 1.3, 1998.2.

[2] 水野政治 他: Distributed Interactive Simulation (DIS)システムの試作(2) - ランタイムインフラストラクチャの実装 -, 本会第 54 回全国大会, 5N-2, 1997.3.

[3] 和泉秀幸 他: HLA をベースとした並列分散シミュレーションシステムの実現 - eRTI1.3 の実現方式概要 -, 本会第 57 回全国大会, 4G-02, 1998.9.

[4] 古市昌一 他: HLA をベースとした並列分散シミュレーションシステムの実現 - eRTI1.3 の実現方式と性能評価 -, 日本シミュレーション学会第18回シミュレーション・テクノロジーコンファレンス, pp.255-258, 1999.