

IEEE 1516 HLA/RTI 표준을 만족하는 데이터 분산 관리 모듈의 설계 및 구현

안정현^{1†} · 홍정희¹ · 김탁곤¹

Design and Implementation of Data Distribution Management Module for IEEE 1516 HLA/RTI

Jung Hyun Ahn · Jeong Hee Hong · Tag Gon Kim

ABSTRACT

The High Level Architecture (HLA) specifies a framework for interoperation between heterogeneous simulators, and Run-Time Infrastructure (RTI) is a implementation of the HLA Interface Specification. The Data Distribution Management (DDM) services, one category of IEEE 1516 HLA/RTI management services, control filters for data transmission and reception of data volume among simulators. In this paper, we propose design concept of DDM and show its implementation for light-weighted RTI. The design concept of DDM is to minimize total amount of message that each federate and a federation process generate using the rate of RTI service execution. The design of our proposed DDM follows that a data transfer mechanism is differently applied as the rate of RTI service execution. A federate usually publishes or subscribes data when it starts. The federate constantly updates the data and modifies associated regions while it continues to advance its simulation time. Therefore, the proposed DDM design provides fast update or region modification in exchange of complex publish and subscribe services. We describe how to process the proposed DDM in IEEE 1516 HLA/RTI and experiment variable scenarios while modifying region, changing overlap ratio, and increasing data volume.

Key words : Data distribution management, HLA, IEEE 1516 RTI

요 약

High Level Architecture (HLA)는 이기종의 시물레이터간 연동을 위한 명세(Specification)이며 Run-Time Infrastructure (RTI)는 이를 구현한 소프트웨어이다. IEEE 1516 HLA/RTI 관리 서비스 중 하나인 데이터 분산 관리 서비스는 시물레이터간에 주고 받는 데이터양을 필터링하는 방법이다. 데이터 분산 관리의 목적은 다른 페더레이트가 필요로 하는 데이터만 전달함으로써 효율적인 데이터 통신을 가능하게 하는 것이다. 본 논문은 경량화된 RTI를 위해 새로운 데이터 분산 관리 설계 발상에 대해서 제안하고 그것의 구현에 대해서 보여준다. 이 데이터 분산 관리의 설계 발상은 페더레이트와 페더레이션 프로세스에서 생산하는 메시지 수를 최소화하는데 있다. 각 페더레이트는 보통 시물레이션이 시작할 때 데이터를 송신(Publish) / 수신(Subscribe) 한다. 또한 각 페더레이트는 시물레이션 시간이 계속적으로 진행되는 도중에 데이터를 갱신하고 이와 연관된 Region을 수정한다. 그러므로 제안된 데이터 분산 관리 디자인은 복잡한 송신(Publish) / 수신(Subscribe) 서비스를 이용하여 빠른 데이터 갱신과 Region을 수정을 할 수 있도록 한다. 이를 적용하여 제안된 DDM 데이터 전달 방식으로 IEEE 1516 HLA/RTI 인터페이스에 만족하도록 구현하였다. 그리고 제안된 DDM 데이터 전달 방식의 성능을 측정하였다. 또한 region을 변경시키며 overlap ratio를 변화시키며 데이터양을 증가시키며 다양한 시나리오로 실험하였다.

주요어 : 데이터 분산 관리, HLA, IEEE 1516 RTI

* 본 연구는 과학재단 특정기초연구(R01-2006-000-111180) 지원으로 수행되었다.

2008년 3월 12일 접수, 2008년 6월 6일 채택

¹⁾ 한국과학기술원 전자전산학과 시스템 모델링 시물레이션 연구실

주 저 자 : 안정현

교신저자 : 안정현

E-mail ; jhahn@smslab.kaist.ac.kr

1. 서 론

대규모 분산 환경에서 개별 시뮬레이터들이 연동하기 위해서는 시뮬레이터 간에 데이터 교환과 시간 동기화가 필요하다. HLA(High Level Architecture)는 DMSO(Defence Modeling and Simulation Office)주도 하에 DoD(Department of Defence)에서 개발된 여러 가지 다른 타입의 시뮬레이터들의 연동을 지원하기 위해 정의된 아키텍처이다^[1-3]. 여기에서 정의된 명세(Specification)를 구현한 소프트웨어가 RTI(Run Time Infrastructure)이다^[4]. DMSO에서는 RTI 1.3을 개발하여 발표 공개하였다. 이는 전체 연동 페더레이션 내에서 각 페더레이트가 정보를 교환할 수 있는 인터페이스와 분산 시스템을 지원하기 위한 함수들을 제공한다. 이 중에 데이터 교환을 위한 주된 서비스는 선언 관리 서비스와 객체 관리 서비스, 데이터 분산 관리 서비스 등이 있다.

그중에서 데이터 분산 관리는 네트워크상에 연결된 여러 개의 시뮬레이터 간에 통신을 할 때 네트워크 양을 줄이기 위해서 보내는 쪽에서 데이터의 필터링하는 방법으로서 주고 받는 데이터양이 많은 대규모 분산 시뮬레이션에서는 반드시 있어야 하는 필요 사항이다.

데이터 교환에서 쓰이는 기존 통신 방식은 각 페더레이트들이 하나의 멀티캐스팅 그룹(Multicasting Group)에 가입하게 되어 이 멀티캐스팅 그룹안의 페더레이트들끼리 그룹 통신을 하게 된다^[5-6]. 최근에는 네트워크 관련 전 분야의 거의 모든 응용에서 Multicast 통신을 사용한다. 그 이유는 Uicast통신 방식은 네트워크 참여 수가 증가함에 따라 네트워크 양이 비례적으로 증가하므로 확장성(Scalability)을 가지지 않지만 Multicast통신은 네트워크 참여의 수와 상관없이 하나의 메시지에 멀티캐스팅 그룹의 주소를 담아 네트워크상으로 전송하면 해당 그룹 주소에 소속된 것은 데이터를 수신할 수 있어서 확장성(Scalability)를 가질 수 있기 때문이다.

하지만 Multicast통신도 멀티캐스팅 그룹을 유지/갱신하는데 비용이 많이 든다. 즉 페더레이트들의 참여/탈퇴에 따라 어느 멀티캐스팅 그룹에 추가/삭제시킬지 여부를 결정하는데 오버헤드가 많고 또한 기존 Multicast 그룹을 계속 관리하는데 비용도 크다. 그리고 Multicast통신을 위해서는 Multicast기능을 제공하는 라우터(Router)가 필요하지만 기존의 ISP(Internet Service Provider)들은 Multicast 기능을 제공하지 않기 때문에 멀티캐스트 패킷(Packet)을 차단한다. 그래서 LAN(Local Area Network)에서만 사용하는 제약 사항이 있다.

또한 본 논문의 경량화 된 RTI에 Multicast 통신을 적용하는 것은 합리적이지 않다. 크지 않은 양의 데이터를 많이 전달할 수 있는 빠르고 가벼운 경량화 된 RTI에 적용하기에는 무리가 있다.

따라서 본 논문에서는 경량화 된(Light-weighted) RTI에 적용 가능한 데이터 분산 관리 서비스를 설계하고 구현하고자 한다. 이는 빠르고 간단한 RTI 소프트웨어를 제작하는데 있어서 페더레이트들 간에 Unicast통신을 하되 확장성(Scalability)을 유지할 수 있도록 최소한의 연결(Connection)을 가지고 페더레이트들 간에 주고 받는 메시지 수를 최소화하는 데이터 전달 접근 방식을 제안하는 것이다.

2. 데이터 분산 관리

2.1 HLA

HLA(High Level Architecture)는 여러 가지 다른 타입의 시뮬레이터들의 연동을 지원하기 위해 정의된 아키텍처로 2000년도에 IEEE 1516 표준으로 채택되었다. HLA는 차세대 DIS표준의 기반으로 네트워크를 이용한 군사용 모의 시뮬레이션 시스템의 구성 요소, 디자인 룰, 인터페이스 등에 관한 전반적인 아키텍처이다. 이 아키텍처의 목적은 시뮬레이션간의 상호 연동성과 시뮬레이션 컴포넌트의 재사용성을 높이기 위한 것이다. HLA는 다음의 세 가지 세부적인 기본 개념으로 정의된다.

HLA Compliance Rules은 HLA를 따르는 시뮬레이션이 되기 위해 필수적으로 따라야 하는 룰이다^[1]. HLA의 페더레이션의 구성 요소들의 역할과 상호 관계에 관한 10여 개의 기본적인 규칙으로서 페더레이션이 되기 위해 필수적으로 따라야 하는 규칙이다. 이중 5개는 페더레이션에 관한 규칙이고 또 다른 5개는 페더레이트에 관한 규칙이다.

RTI는 페더레이트가 연동하는 동안 일반적인 서비스 인터페이스를 제공해주는 분산된 소프트웨어 모음이라고 할 수 있다^[2,4]. 그 특징에는 기반 운영 체제 및 통신 시스템을 포함한 여러 하드웨어 플랫폼으로의 편리한 이식성을 제공하고, 시뮬레이션 간의 상호 연동성을 제공한다는 것이다. RTI의 서비스는 HLA Interface Specification에 기술되어 있다.

OMT(Object Model Templates)는 페더레이션에 존재할 개체와, 개체의 속성, 페더레이션 내에서 개체간의 상호작용(Interaction)을 표현하기 위한 수단으로 FOM(Federation Object Model)과 SOM(Simulation Object

Model)에 있는 정보이다^[3]. FOM은 HLA의 특정 페더레이션을 위해서 필수적으로 필요로 하는 개체, 개체 속성, 페더레이트 간의 상호작용 등에 관한 정보를 기술한다. 또한 특정 페더레이션 실행(Federation Execution)을 맡고 있는 시뮬레이션과 페더레이션 실행에서 적용될 특정 시나리오 리스트를 언급하기도 한다. SOM은 페더레이션에 제공될 각각의 시뮬레이션의 필수적인 역할 또는 능력에 관하여 기술하는데, 현재 SOM에 관한 표준 포맷은 페더레이션 개발자로 하여금 페더레이션 내에 특정 역할을 맡을 시뮬레이션 시스템의 적합성을 빠르게 결정할 수 있는 수단을 제공한다. XML기반의 테이블로 구성된 OMT는 표준 데이터 문서 형식이기에 모델의 재사용을 장려한다.

2.2 Data Distribution Management (DDM) Service

IEEE 1516 HLA/RTI에서 데이터 분산 관리는 RTI를 통해 모든 페더레이트간에 객체를 발견하거나 객체의 속성 변수(attribute)나 상호 작용의 파라미터의 흐름을 제어한다^[7-8]. 선언 관리(Declaration Management)에서의 송신 선언과 수신 선언을 통해서 데이터의 주고 받음을 먼저 설정할 수 있다. 즉 전체 페더레이션 중에서 자신의 수신 선언한 데이터만 받을 수 있고 자신의 송신 선언한 데이터를 보낼 수 있는 것이다. 이러한 타입의 필터링을 클래스 기반(Class-based) 필터링이라고 한다. 규모가 작은 페더레이션이나 주고 받는 데이터의 수가 적을 때 클래스 기반 필터링은 성능에서나 확장성에서 효율적이다. 하지만 주고 받는 데이터가 많고 규모가 큰 페더레이션의 경우에는 성능이나 확장성을 개선하기 위해서 좀 더 세부적인 필터링이 필요하다. 여기에 관련된 것이 데이터 분산 관리이다.

데이터 분산 관리(Data Distribution Management) 서비스는 Region을 생성 변경하거나 생성된 Region과 선언된 객체 혹은 상호 작용에 대해서 연관시킨다. 또한 Region안에 있는 각 차원의 실제적인 바운드 값을 변경시킨다. 서로 다른 페더레이트들은 이 바운드 값의 교차 여부에 따라 데이터를 주고 받을 수 있다. 이는 HLA의 페더레이션 실행의 수행 과정 동안 선언한 데이터를 적절히 지역적으로 분배시킴으로써 유연하고 확장 가능한 메커니즘을 제공한다. 데이터 분산 관리는 먼저 수행된 데이터의 송신 / 수신 선언 후 이러한 특정 데이터를 가지는 Region 정보와 연관 지어 데이터를 등록하는 것이다.

그림 1에는 3개의 페더레이트가 있고 페더레이트 A는 갠신 Region U1을 가지고 있고 페더레이트 B와 페더레

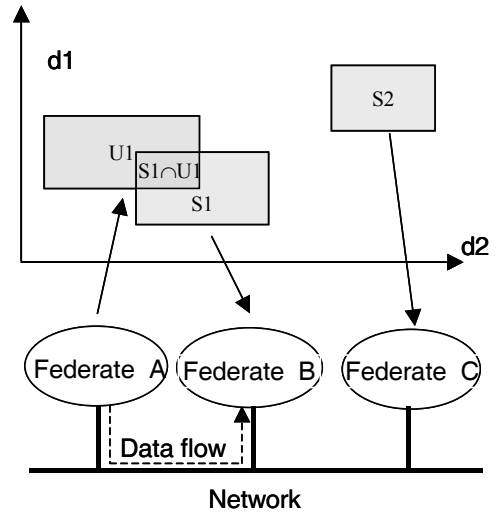


그림 1. 데이터 분산 관리^[9]

이트 C는 수신 Region S1, S2를 각각 가지고 있고 이는 2 차원을 가지는 통신 공간에 선언되어 있다. 페더레이트 A의 갠신 Region U1이 페더레이트 B의 수신 Region S1에 겹치기 때문에 페더레이트 A의 갠신 Region U1과 연관성이 있는 데이터가 페더레이트 B로 보내지는 것이다. 하지만 페더레이트 A의 갠신 Region U1이 페더레이트 C의 수신 Region S2에 겹치지 않기 때문에 데이터가 페더레이트 C로 가지 않게 되어 필터링이 되는 것이다.

3. 제안된 데이터 분산 알고리즘

IEEE 1516 HLA/RTI의 데이터 분산 관리 서비스를 이용하여 서로 다른 시뮬레이터간의 데이터 교환을 위해서 기존의 RTI1.3은 Multicast통신을 한다. 이것은 각각의 페더레이트들간의 직접 통신하는 방식인 Unicast통신 방식에 비해 확장성 면에서 유리하기 때문이다. 하지만 Multicast통신은 멀티캐스팅 그룹을 유지/갱신하는데 비용이 많이 든다. 또한 Multicast 통신은 RTI에서 데이터를 주고 받을 때 불필요한 오버헤드가 든다.

본 논문은 이와 다르게 접근하여 Unicast통신을 하며 Unicast통신의 문제점인 제한된 회선 용량을 여러 수신자가 공유해야 하는 문제점과 여러 개의 연결이 필요한 문제점을 동시에 해결하고자 한다. 페더레이트들 간에 Unicast통신을 하되 확장성(Scalability)을 유지할 수 있도록 최소한의 연결(Connection)을 가지는 데이터 분산 관리의 설계를 제안한다.

3.1 기본적인 서비스 빈도수

기본적으로 페더레이트의 생명 주기를 살펴보면 페더레이트가 가지는 데이터가 정적으로 이루어지는 경우와 동적으로 이루어지는 경우로 나뉜다. 개별 페더레이트의 참여/탈퇴나 페더레이션에 쓰이는 데이터를 위해 선언하는 초기화 부분은 전체 페더레이트의 생명 주기에서 정적으로 이루어짐을 알 수 있다. 하지만 시뮬레이션 수행 중에 객체의 갱신이나 Region의 변경은 동적인 데이터 갱신으로 시뮬레이션을 하는 동안은 계속적으로 이루어짐을 알 수 있다.

위에서 살펴본 바와 같이 초기화에서 데이터를 정적으로 생성하는 경우와 시뮬레이션 수행에서 데이터를 동적으로 생성하는 서비스는 다를 뿐더러 사용하는 빈도수도 다르다. 초기화 부분에서 사용된 P(Publish) / S(Subscribe) 데이터의 선언, Region의 생성, 생성된 Region과 P / S 데이터의 연관성 지움 등의 서비스는 사용하는 빈도수가 낮은 서비스이고 시뮬레이션 부분에서 사용하는 Region 값의 변경 혹은 데이터의 갱신 등의 서비스는 사용하는 빈도수가 높은 서비스이다.

3.2 데이터 교환 방식

데이터를 전달하기 위해서는 송신 선언과 수신 선언을 할 때의 P / S 데이터를 저장뿐 아니라 Region 데이터, Region과 데이터의 연관성 데이터가 미리 페더레이션 실행이나 각 페더레이트에 있어야 한다. 데이터를 교환함에 있어서 데이터를 저장하는 곳에 따라 중앙식(Centralized Approach)과 분산식(Distributed Approach)이 있다. 본 논문에서는 중앙식과 분산식, 두 경우를 다 고려한다. 중앙식의 경우에는 중앙에서 페더레이션 실행이 하나의 정보 저장소가 되어 각 개별 페더레이트에게로 데이터를 전달한다. 이 경우는 페더레이션 실행에서 데이터를 관리하기 때문에 쉽게 관리할 수 있지만 페더레이션 실행이 병목점(Bottleneck)이 될 수 있다. 분산식의 경우에는 각 페더레이트가 정보 저장소가 되어 개별 페더레이트간에 직접 통신이 가능하다. 이 경우는 데이터가 각 페더레이트로 분산되어 있기 때문에 데이터 관리에 어려운 점이 있다.

그림 2는 중앙식의 데이터 전달 방식을 보여준다. 각 페더레이트가 데이터를 전달하기 위해서는 중앙에 페더레이션 실행이 있어서 이것을 통해서 데이터의 흐름을 제어하고 또한 데이터 전달의 책임자가 된다.

그림 3은 분산식 데이터 전달 방식을 보여준다. 이 방식은 분산되어 있는 각 페더레이트가 데이터를 관리하기 때문에 다른 페더레이트로 직접 통신이 가능하다.

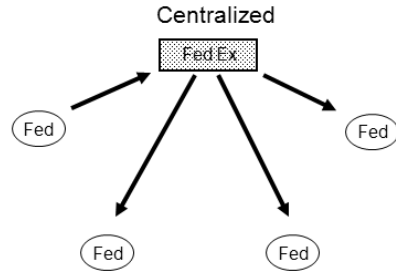


그림 2. 중앙식의 데이터 전달 방식

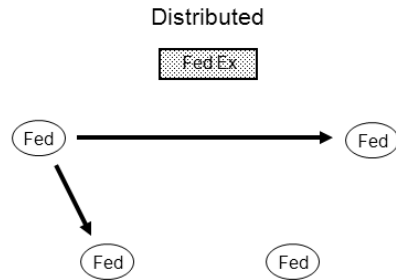


그림 3. 분산식의 데이터 전달 방식

표 1. 서비스 빈도수 및 데이터 전달 방식

서비스	서비스 빈도수	데이터 전달 방식
Associate(Unassociate) Region For Updates	낮음	중앙식
Create Region / Delete Region	낮음	중앙식
Commit Region Modification	높음	분산식
Register Object Instance With Region	높음	분산식
Request Class Attribute Value Update With Region	높음	분산식
Send Interaction With Region	높음	분산식
Subscribe Interaction (Object Class) Class With Region	낮음	중앙식
Unsubscribe Interaction (Object Class) Class With Region	낮음	중앙식

3.3 설계 주요 발상(Concept)

본 논문은 데이터를 전달하는데 있어서 최소한의 메시지 전달을 하는 방향으로 디자인하였다. 제안된 DDM 데이터 전달 방식은 표 1과 같이 서비스 빈도수에 따라 데이터 전달 방식을 다르게 적용하였다. 서비스 빈도수가 낮은 서비스의 경우에는 중앙 식으로 데이터를 전달하고

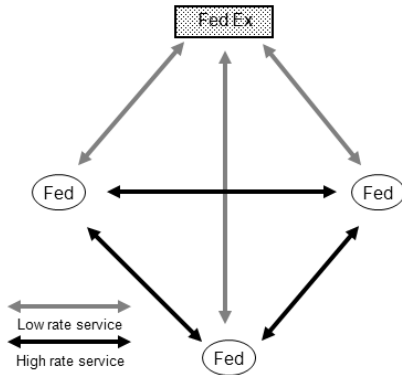


그림 4. 제안된 DDM 데이터 전달 방식

서비스 빈도수가 높은 서비스의 경우에는 분산식으로 데이터를 전달하도록 디자인 발상을 제안한다. 이러한 통합된 데이터 전달 방식은 중앙식과 분산식의 장점을 다 제공한다.

세밀하게 살펴보면, 빈도수가 낮은 서비스는 중앙식으로 데이터를 전달한다. 페더레이션에 쓰이는 데이터를 위해 선언하는 것은 초기화 부분에서 정해진다. 이러한 서비스의 빈도수는 전체 시뮬레이션 수행에서 적게 불린다. 빈도수가 낮은 서비스 같은 경우에 제안된 DDM 데이터 전달 방식은 개별 페더레이트에게 미리 P / S 데이터 및 DDM의 Region 데이터가 저장되도록 페더레이트와 페더레이션 실행간에 다소 복잡하게 메시지를 주고 받는다. 이렇게 함으로써 페더레이트간 직접 통신이 가능하도록 하는 다른 페더레이트의 주소, 다른 페더레이트가 가진 Region 데이터 등의 추가 데이터를 가질 수 있다. 여기서 가질 수 있는 다른 페더레이트의 주소는 다른 페더레이트로 직접 통신하도록 돕는다. 복잡한 메시지 시퀀스를 가지지만 이는 빈도수가 낮기 때문에 전체 메시지 수에 주는 영향은 미흡하다.

빈도수가 높은 서비스는 분산식으로 데이터를 전달한다. 시뮬레이션 수행 중에 객체의 갱신이나 Region 데이터의 변경은 시뮬레이션을 하는 동안은 계속적으로 이루어진다. 빈도수가 높은 서비스 같은 경우에 제안된 DDM 데이터 전달 방식은 페더레이트간에 간단하게 메시지를 주고 받는다. 중앙식의 데이터 전달 방식으로 얻은 추가 데이터 중에 수신 페더레이트 주소를 가지고 송신 페더레이트는 수신 페더레이트로 데이터를 직접 통신하게 된다. 이것은 보내는 측에서 전달되는 데이터의 필터링도 가능하게 된다. 본 논문은 제안된 DDM 데이터 전달 방식을 경량화된 RTI에 적용하고 개발하였다.

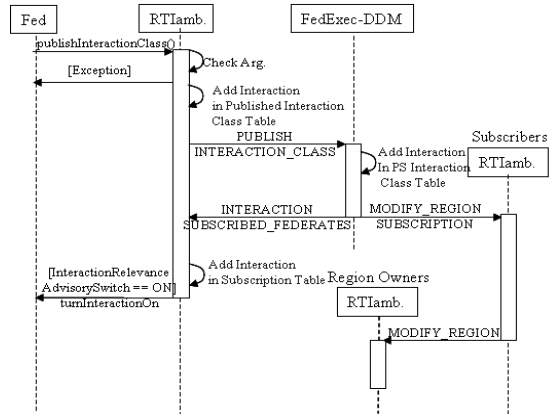


그림 5. publishInteractionClass() 서비스

3.4 메시지 시퀀스

페더레이트와 페더레이션 실행간의 메시지 교환에 대해서 살펴보면 다음과 같다. 먼저 P / S 데이터를 가지고 실제 종착 페더레이트를 구하는 과정을 시퀀스 다이어그램으로 살펴본다. 본 설계의 목적은 최소한의 메시지 전달이다. 그렇게 함으로 경량화된 RTI를 개발하고 빠르게 동작하도록 한다. 여러 개 서비스의 시퀀스 다이어그램 중에서 상호 작용을 송신 선언하는 publishInteractionClass()와 실제 상호 작용을 보내는 sendInteractionWithRegion()에 대한 시퀀스 다이어그램에 대해서 살펴본다.

페더레이션 실행(Fed Exec)은 각 페더레이션을 관리한다. 이는 개별 페더레이트의 참여/탈퇴를 허락하고 참여한 페더레이트 간에 데이터 교환을 허락한다. 페더레이트에는 사용자가 HLA API를 호출하면 이는 RTIambassador를 통하여 페더레이트 내에 있는 로컬 RTI 컴포넌트(Local RTI Component)가 정보를 처리하거나 네트워크 상에 연결된 페더레이션 실행이나 다른 페더레이트와 통신을 한다.

그림 5와 같이 publishInteractionClass()는 정적인 P / S 데이터를 저장하도록 하는 서비스로 전체 페더레이트 수행 시간 중에서 데이터를 초기화 하는 단계에서만 선언된다. 이 서비스의 사용 빈도수는 매우 낮다. 그렇기 때문에 조금 복잡하게 메시지들이 전달하더라도 전체적인 메시지 전달 횟수에 미치는 영향은 작다.

사용자에 의해서 호출된 publishInteractionClass()는 상호작용의 핸들을 이용하여 RTIambassador를 통하여 페더레이션 실행으로 메시지를 보낸다. 페더레이션 실행은 이를 가지고 상호작용 P / S 테이블을 만든다. P / S 테이블에는 그 상호 작용을 송신 / 수신 선언한 페더레이

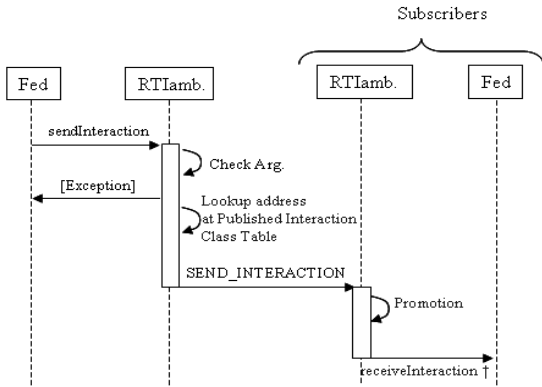


그림 6. sendInteractionWithRegion() 서비스

트의 핸들을 저장한 테이블이다. 그리고 특정 상호 작용을 송신 선언한 페더레이트에게는 그 상호 작용을 수신 선언한 페더레이트의 핸들 값과 연관성이 있는 Region 핸들을 알려주고 반대로 특정 상호 작용을 수신 선언한 페더레이트에게는 그 상호 작용을 송신 선언한 페더레이트를 알려준다. 송신 페더레이트는 상호 작용에 대해서 송신 선언한 페더레이트로 실제 Region안에 있는 바운드 값을 전달함으로 전체 서비스를 마친다. 이렇게 함으로써 P / S 데이터를 구성할 때 송신하는 페더레이트는 이미 수신하는 페더레이트 주소 및 Region 데이터의 구체적인 값에 대해서 다 저장하고 있다.

sendInteractionWithRegion()서비스는 동적인 서비스로 이는 전체 페더레이트 수행 시간 중에서 시뮬레이션 중 계속적으로 수행된다. 그러므로 이 서비스의 사용 빈도수는 매우 높다. 그렇기 때문에 간단하게 메시지들이 전달하여 직접 메시지를 전달하도록 하여 전체 전달되는 메시지 수에 직접적인 영향을 미친다. 사용자에 의해서 호출된 sendInteractionWithRegion()서비스는 상호작용의 핸들과 Region 핸들을 이용하여 실제적인 수신 선언의 Region 데이터와 매칭한 결과에 따라 수신 선언한 페더레이트로 상호 작용을 보낸다. 이미 P / S 데이터를 저장할 때 수신하는 페더레이트 및 Region 데이터와 Region의 매칭 결과 직접 수신 페더레이트를 통하여 메시지를 전달하고 전체적인 서비스를 마친다.

3.5 데이터 테이블

페더레이트와 페더레이션 실행간의 메시지 교환할 때 쓰이는 데이터 테이블은 송신 선언과 수신 선언을 할 때의 P / S 데이터를 저장뿐 아니라 Region데이터, Region

표 2. 페더레이션 실행 측의 P / S 테이블

Interaction Class Handle	Published Federate	Subscribed Federate & Region
1	F1	(F2, R1)

표 3. 페더레이트 측의 P / S 테이블

Interaction Class Handle	Subscribed Federate & Region Pair Set
1	(F2, R1)

과 P / S데이터의 연관성 정보도 다루어야 한다. RTI 내에서는 정적인 P / S 정보를 잘 관리하여야 하고 또 다른 동적인 정보를 잘 반영할 수 있어야 한다. 그렇게 함으로 전체적인 메시지 전달 횟수를 최소화하고자 한다. 정적인 P / S 데이터뿐 아니라 동적인 Region 데이터 값의 변경도 잘 반영하는 데이터 구조여야 한다. 그리고 동적 데이터의 변경에 따라 생성되는 변경 메시지를 최소화해야 하면서 이를 기존의 데이터 구조에 빠르게 반영하여야 한다. 마지막으로 모든 데이터를 저장하는데 있어서 메모리를 최소한으로 관리해야 한다. 이를 위해서 다음과 같은 데이터 구조를 제안한다.

페더레이션 실행에서 관리하는 테이블과 페더레이트에서 관리하는 테이블로 나뉜다. 먼저 페더레이션 실행 측의 P / S 데이터를 위한 데이터 구조이다.

표 2에서 Interaction Class Handle은 개별 페더레이트들이 송신 혹은 수신 선언을 할 때 유일하게 부여하는 ID 값으로 이는 RTI내에서 임의로 부과하고 전체 페더레이션 안에서는 동일하게 유지된다. Published Federate는 특정 객체나 상호 작용을 송신 선언을 한 페더레이트를 나타낸다. 그래서 특정 데이터에 대해서 송신하는 페더레이트들이 어떤 것이 있는지 알 수 있다. Subscribed Federate & Region의 경우 특정 객체나 상호 작용을 수신 선언을 한 페더레이트를 나타낸다. 여기에 추가적으로 그 데이터에 연관성을 지은 Region 데이터까지 나타낸다. Region 데이터가 없더라도 Default Region으로 연관성을 지어 수신 선언되었다고 구성할 수 있다.

표 3은 페더레이트측에서 P / S Table이다. 이를 가지고 상호 작용을 보낼 때 수신하는 페더레이트 및 Region 데이터 정보와 Region의 매칭 결과 직접 수신 페더레이트로 메시지를 보내게 된다.

이렇게 설계한 자료들을 바탕으로 본 논문에서 제안된 DDM 데이터 전달 방식을 SMSRTI(Systems Modeling

and Simulation Runtime Infrastructure)에 적용하여 구현하였다^[10].

4. 실험 결과 및 분석

제안된 DDM 데이터 전달 방식으로 제시한 데이터 교환에 의한 지연 시간(Latency) 측정하여 성능을 검증한다. 먼저, 성능 측정이 수행된 실험 환경에 대해 설명한다. 전체 실험 환경은 다음과 같다. 전체적으로 페더레이트들과 페더레이션을 관리하는 RTI 실행(Rtiexec)과 페더레이트를 관리하는 페더레이션 실행으로 구성되어 있다. 전체 페더레이션은 송신 페더레이트와 수신 페더레이트로 구성되어 있다. 송신 페더레이트는 단지 데이터를 보내기만 하도록 설정되어 있으며 수신 페더레이트는 단지 데이터를 받기만 하도록 설정되어 있다. 네트워크에 연결된 컴퓨터에 8개의 페더레이트들이 분산되어 있다.

각 페더레이트가 있는 컴퓨터 사양은 Pentium IV 2.8 Ghz, 512Mb 메모리 등으로 구성되어 있고 운영 체제는 Window XP이다. 네트워크는 100Mbps Fast Ethernet switch를 기준으로 했는데 외부 네트워크의 역할을 할 switch와 내부 네트워크의 역할을 할 switch를 따로 두어 다른 기계에 의한 간섭을 피하고 각 연결된 컴퓨터들만의 통신을 위한 독립된 채널을 확보하도록 했다. 지연 시간은 송신 페더레이트에서 수신 페더레이트로 데이터가 이동했을 때 걸리는 시간으로 이 값을 구하기 위해서는 송신 페더레이트와 수신 페더레이트가 실시간으로 동기화가 되어야 한다. 전체 페더레이션은 주기적으로 상호작용을 주고 받는 실험 환경으로 되어있다. 페더레이트간의 메시지 지연시간을 측정하기 위해 메시지를 보내고 난 뒤의 송신 페더레이트 시간과 이 메시지를 받는 뒤의 수신 페더레이트 시간의 차를 구하여 이를 평균을 취했다.

제안된 DDM 데이터 전달 방식의 성능을 평가하기 위한 Region 데이터 변경에는 다음과 같은 파라미터들이 있다^[11]. 이는 크게 3가지로 나뉘는데, 데이터 양(Data Volume)과 Region 데이터 변경 정도(Modification Rate)와 Region 데이터 겹침 정도(Overlap Ratio)가 있다. 먼저 데이터 양은 전달되는 Region 데이터의 개수로 이러한 데이터가 많을수록 많은 메모리 사이즈가 필요하고 처리 시간이 늘어난다. 다음으로 Region 데이터 변경 정도는 특정 시간 간격사이에 전체 Region 데이터 중에 얼마나 변경되는가를 나타내는 비율이다. Region 데이터가 변경될 때마다 RTI는 Region 데이터의 겹침 여부를 다시

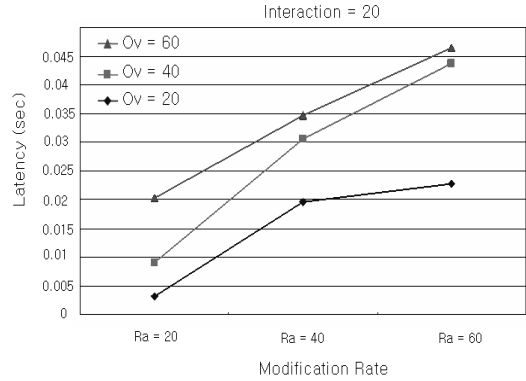


그림 7. Region 시나리오 : FOM에 정의된 Interaction 수 = 20

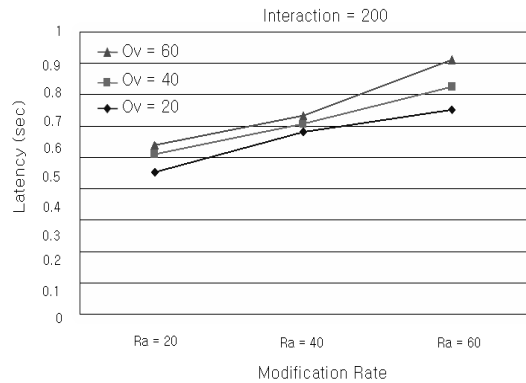


그림 8. Region 시나리오 : FOM에 정의된 Interaction 수 = 200

계산해야 한다. 마지막으로 Region 데이터 겹침 정도는 전체 Region 데이터 중에서 얼마나 겹치는가를 나타내는 비율이다. 높은 Region 데이터의 겹침 정도는 겹침 여부를 확인하는데 많은 처리 시간이 필요하고 또한 많은 데이터들이 전달해야 하기 때문에 데이터 전달 시간을 지연시킨다. 성능은 지연 시간으로 결정되며 여러 파라미터를 변화시키면서 측정된 성능은 다음과 같다.

그림 7은 시나리오 파라미터 중 FOM에 정의된 상호작용 수가 20개이고 주고 받는 총 메시지 수가 4000개 일 때 여러 파라미터를 변경시키면서 지연 시간을 측정된 결과이다. 위 그림에서 나타나듯이 지연 시간은 Region 데이터의 변경 정도가 커질수록 Region 데이터 겹침 정도가 커질수록 증가하는 경향이 나타난다. 이는 파라미터가 변경되는 경우가 많고 겹치는 경우가 많을수록 Region 데이터를 매칭하여 송신 페더레이트를 구하는 시간이 길어지기 때문이다.

그림 8은 시나리오 파라미터 중 FOM에 정의된 상호

작용 수가 200개이고 주고 받는 총 메시지 수가 4000개 일 때 여러 시나리오 파라미터를 변경시키면서 지연 시간을 측정할 결과이다. 위 그림도 마찬가지로 지연 시간은 Region 데이터의 변경 정도가 커질수록 Region 데이터 접침 정도가 커질수록 증가하는 경향이 나타난다. 이와 더불어 FOM에 정의된 Region도 많고 또한 주고 받는 메시지가 많아진 결과 지연 시간이 더욱 커지게 되었다.

5. 결론 및 추후 과제

규모가 작은 페더레이션이나 주고 받는 데이터의 수가 적은 페더레이션의 경우 데이터 분산 관리 서비스가 시뮬레이션의 성능에 큰 영향을 미치지 않겠지만 주고 받는 데이터가 많고 규모가 큰 페더레이션의 경우에는 필터링을 거치지 않은 수많은 데이터의 전달 때문에 생기는 네트워크 양으로 인해 시뮬레이션의 성능에 큰 영향을 끼친다. 본 논문은 대규모 분산 환경에서 정교한 데이터 분산 관리를 할 수 있는 방안을 제안하고 구현하였다.

본 논문에서 IEEE 1516 HLA/RTI 서비스 중 데이터 분산 관리 서비스를 위해 P / S 데이터를 이용한 필터링을 위한 설계 발상을 제공하였다. 이는 시뮬레이션 초기화 단계에서 P / S 데이터를 가지고 메시지 전달 횟수를 최소화 하는데 있다. 빈도수가 높은 서비스의 경우에는 간단하고 빠르게 메시지를 전달하는 분산 식의 데이터 전달 방안을 쓰고 빈도수가 낮은 서비스의 경우에는 복잡한 메시지 전달하는 중앙 식의 데이터 전달 방안을 썼다. 빈도수가 낮은 서비스는 복잡하게 메시지를 전달하지만 이는 빈도수가 높은 서비스에 사용할 P / S 데이터를 미리 각 페더레이트에 가지고 있도록 설계하였다. 이를 설계하고 구현함에 있어서 필요한 메시지 전달 방안과 데이터 구조에 대해서 살펴보았다.

본 논문에서 제안된 DDM 데이터 전달 방식은 간단하면서도 빠른 경량적인 RTI에 적합하다. 또한 제안된 방식을 SMSRTI 개발에 있어서 데이터 분산 관리 모듈에 적용하여 구현하였고 이것으로 다양한 Region 데이터의 파라미터 변경에 따라 성능을 분석해 보았다. SMSRTI같은 경우에는 이제 막 설계와 구현 단계를 마쳤기에 성능 향상이 추후 과제로 남아있다.

감사의 글

본 논문은 과학재단 특정기초연구(R01-2006-000-111180)의 지원으로 “Collaborative BPMS 환경에서 모델링, 설

계, 분석, 구축을 위한 BPR 도구로 활용될 수 있는 HLA/RTI 기반의 BPM middleware 및 요소기술 개발” 과제의 연구 결과이다.

참고 문헌

1. IEEE, “IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules, IEEE Std 1516,” *Institute of Electrical and Electronics Engineers, New York, December 11, 2000.*
2. IEEE, “IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification, IEEE Std 1516.1-2000,” *Institute of Electrical and Electronics Engineers, New York, March 9, 2000.*
3. IEEE, “IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template, IEEE Std 1516.2-2000,” *Institute of Electrical and Electronics Engineers, New York, March 9, 2001.*
4. Defense Modeling and Simulation Office, “High Level Architecture RTI Interface Specification, Version 1.3,” *April 1998.*
5. Daniel J. Van Hook et al; 1996, “Approaches to RTI Implementation of HLA Data Distribution Management Services,” *Proceeding of the 15th DIS Workshop.*
6. Mark Hyett, “Implementation of the Data Distribution Management Services in the RTI-NG,” *02S-SIW-044, Spring Simulation Interoperability Workshop, March 2002.*
7. K. L. Morse and J. S. Steinman, “Data Distribution Management in the HLA: Multidimensional Regions Physically Correct Filtering,” *Proceedings Spring SIW Workshop, 1997, 97S-SIW-052.*
8. Gary Tan, Rassul Ayani, Yusong Zhang and Farshad Moradi, 2000, “Grid-based Data Management in Distributed Simulation,” *in Proceeding of 33rd Annual Simulation Symposium.*
9. K. L. Morse and J. S. Steinman, “Data Distribution Management in the HLA: Multidimensional Regions Physically Correct Filtering,” *Proceedings Spring SIW Workshop, 1997, 97S-SIW-052.*
10. J. H. Kim, “Proposal of High Level Architecture Extension and Run-Time Infrastructure Implementation,” *Ph.D. Thesis, KAIST, 2006.*
11. Katherine L. Morse, “Characterizing Scenarios for DDM Performance and Benchmarking RTIs,” *SIW Workshop, 1999, 99S-SIW-054.*



안 정 현 (jhahn@smslab.kaist.ac.kr)

2005 부산대학교 전자전자정보컴퓨터공학부 학사
 2007 KAIST 전자전산학과 석사
 2007~현재 KAIST 전자전산학과 박사과정

관심분야 : HLA/RTI, Data Distribution Management, Distributed Simulation



홍 정 희 (jihong@smslab.kaist.ac.kr)

2005 부산대학교 전자전자정보컴퓨터공학부 학사
 2007 KAIST 전자전산학과 석사
 2007~현재 KAIST 전자전산학과 박사과정

관심분야 : HLA/RTI, Time Management, Distributed Simulation



김 탁 곤 (tkim@ee.kaist.ac.kr)

1975 부산대학교 전자공학과(학사)
 1980 경북대학교 전자공학과(석사)
 1988 Univ. of Arizona, 전기및컴퓨터공학과 (박사)
 1980~1983년 부경대학교, 통신공학과, 전임강사
 1987~1989년 (미)아리조나 환경연구소, 연구엔지니어
 1989~1991년 Univ. of Kansas, 전기및컴퓨터공학과, 조교수
 1991~현재 KAIST 전자전산학과, 교수
 한국시물레이션 학회 회장 역임
 SIMULATION(국제시물레이션 학회(SCS) 논문지)
 편집위원장 역임

국제 학회 : SCS Fellow, IEEE Senior Member, ACM Member

자격증 : 모델링 시물레이션 기술사(미국)

국방 M&S 자문위원 : 국방부, 합참, KIDA, ADD 등

관심분야 : 모델링/시물레이션 이론, 방법론 및 환경개발, 시물레이터 연동