

HLA 규격 1.3에 적합한 분산 미들웨어

Java/RTI 개발

엄용섭, 박성봉, 안명수

(주)넷커스터마이즈

대전시 유성구 장동 48 대전S/W지원센터 201호 (msahn@netcus.com)

Development of a HLA Specification 1.3

Compliant Distributed Middleware Java/RTI

Yong Seob Eom, Sung Bong Park, Myung Soo Ahn

요 약

최근들어 분산 시뮬레이션을 위해 미국 국방성에서 제안한 HLA(High Level Architecture)를 산업계에 적용하기 위하여 미들웨어 개발이 활발하다. 주요 응용분야는 게임 및 다자간 회의 시스템, 실시간 제어 시스템 등이다. 본 논문에서는 (주)넷커스터마이즈가 HLA Spec. 1.3 표준안에 기반하여 순수 Java 언어를 사용하여 개발한 미들웨어인 Java/RTI를 소개한다. Java/RTI는 Java 언어만을 사용함으로써 운영체제에 제한을 받지 않고 RTI의 이식이 가능하게 하여 이식성을 크게 향상시켰다. 또한, HLA 규격이 요구하는 모든 기능을 자체 개발함으로써 향후 RTI의 성능 및 서비스 확장을 위한 기반을 마련하였으며 실시간 제어 및 VOIP 응용 소프트웨어 개발을 통해 개발 편의성 및 성능이 검증되었다. RTI의 실행 환경을 모니터링할 수 있는 소프트웨어까지 개발함으로써 소프트웨어의 운용 및 관리 효율성을 높였다. 본 논문에서 소개한 Java/RTI는 군사용 분산 시뮬레이션은 물론 게임, 실시간 제어 소프트웨어 등 다양한 분야에 응용될 수 있으며 현재 패키지 소프트웨어로 상품화가 추진중에 있다. 또한, 개발된 Java/RTI를 확장하여 실시간 환경에서 QoS(Quality of Service)를 제공하는 성능 측정자 및 자원 관리자 개발 중에 있다.

I. 서 론

인터넷의 급속한 발달로 인해 대부분의 컴퓨터 이용자들은 고속의 네트워크 접속 환경을 지원받게 되었다. 이로 인해 VoIP (Voice over Internet Protocol), 능동형 자율 제어 시스템, 분산 의료 서비스 시스템, 가상현실 및 게임 소프트웨어 분야 등과 같이 실시간 분산 처리를 요구하는 고도화된

서비스 분야가 시장의 관심을 끌고 있으며 이를 효율적으로 지원하는 플랫폼에 대한 개발 경쟁이 치열하다.

군사용 분산 시뮬레이션을 위해 미국방성에서 제안한 HLA(High Level Architecture)는 IEEE에 의해 국제 표준으로 제정되었으며 군사용뿐만 아니라 민수분야에도 효과적으로 적용될 수 있는 구조임에도 불구하고 현재까지는 CORBA를 기반으

로 하여 인터페이스만을 일부 구현하여 게임 등 한정된 분야에 적용하기 위한 시도가 있었다.

본 논문에서는 산업계와 민간분야에서 공통적으로 요구되는 실시간 협업을 지원하는 미들웨어를 HLA를 기반으로 하여 구현한 Java/RTI에 대해 소개한다. Java/RTI는 로봇, 분산게임, 정보가전, 실시간 제어 시스템 등에서 공통적으로 요구되는 실시간 QoS 보장 기능을 수백Kbyte 이내의 경량형 소프트웨어로 구현하여 Java Chip 등에서의 이식이 용이하도록 개발이 추진되었다.

II. Java/RTI 설계

HLA는 독립적으로 실행되는 분산 프로세서간 객체 공유를 지원하는 RTI(Run Time Infrastructure)가 제공해야 할 서비스를 규격화한다. 그림 1에서 보여 주는 바와 같이 분산 미들웨어를 이용하는 응용 제품군과 실행 환경은 다음과 같은 구성 요소를 포함한다.

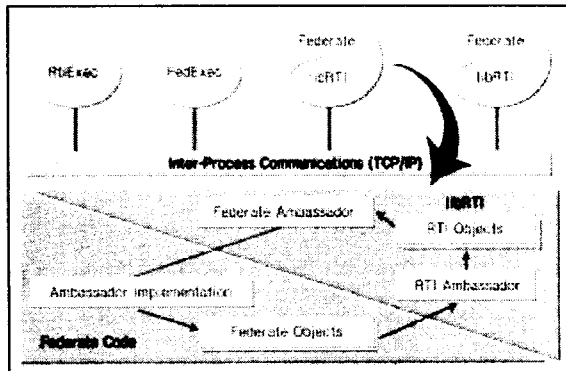


그림 1. Java/RTI 구성 요소

- 1) Federate : 독립적으로 수행되는 분산 프로세스들로써 다른 Federate와의 객체를 공유한다.
- 2) FedExec : 분산 프로세스(Federate)들로 구성되는 분산 소프트웨어인 Federation을 관리하기 위한 소프트웨어 모듈로써 Federate간 객체 공유를 위한 서비스 기능을 실제로 수행하는 모듈이다.

- 3) RtiExec : 네트워크 상에서 수행되고 있는 Federation들을 관리하는 기능을 수행한다. 네트워크상에서 여러개의 Federation이 실행될 수 있다.

그림 1에서 보여 주는 바와 같이 RTI는 응용 프로그램에 해당하는 Federate에게 객체 공유를 위해 필요한 서비스를 제공해야 한다. HLA에서 정의하는 서비스들이 표 1에 요약되어 있다.

표 1. Java/RTI 서비스

Area	기능
Federation Management	<ul style="list-style-type: none"> - Federation의 생성과 소멸 : RtiExec와 FedExec로 구현 - 분산 Object 공유를 위한 객체 초기화 및 Routing Space와 Data Distribution 관리 기능 - Federate (Java/RTI를 이용하여 분산 처리를 수행하는 응용 프로그램)의 Joining/Resigning 처리
Declaration Management	<ul style="list-style-type: none"> - Federate가 송수신하는 분산 Object의 정의 - Federate의 분산 Object에 대한 관심을 처리 (Subscription/Publication)
Object Management	<ul style="list-style-type: none"> - 분산 Object와 Interaction (Event style의 Temporal Object)의 생성과 삭제, 변경을 처리 - Object 식별자 관리 - Object Registration 및 배포 관리
Ownership Management	<ul style="list-style-type: none"> - Object와 Interaction의 Update 소유권을 관리하고 Object내의 특정 Attribute에 대한 소유권을 관리 - 소유권의 이전 처리 및 소유권 공유 관리
Time Management	<ul style="list-style-type: none"> - TSO(Time-Stamp Order) Event의 처리 - Federation 내에서 Causal Behavior의 지원 - 서로 다른 Timing Scheme을 갖는 Federate 사이의 Interaction을 지원 - Time Advance의 처리
Data Distribution Management	<ul style="list-style-type: none"> - Data 공유를 위한 효율적인 Routing 지원 : Routing Region의 생성 및 소멸 - Data Distribution의 식별 - Routing 조건의 설정 및 관리

표 1과 같은 서비스를 제공하도록 Java 라이브러리를 구현하여 Federate 프로그래머가 사용할 수 있도록 지원하며 RtiExec와 FedExec는 별도의

독립된 응용 프로그램으로 구현된다. Java/RTI의 구현에 대한 내용은 다음 절에서 자세하게 기술된다.

III. Java/RTI 구현

Java/RTI가 Federate 프로그래머에게 제공하는 libRTI는 HLA의 Service를 구현하는 Java Package이다. 즉, 응용 프로그래머가 사용할 수 있도록 제공하는 라이브러리 함수들이다. Federate에서는 libRTI가 제공하는 함수를 호출하여 Federate들과 Object를 공유하게 된다. libRTI는 HLA가 제공하도록 규정한 함수를 Java Class로 구현한 것이다. 그림 2에는 구현된 Java/RTI의 Thread와 Class 구조를 보여 준다.

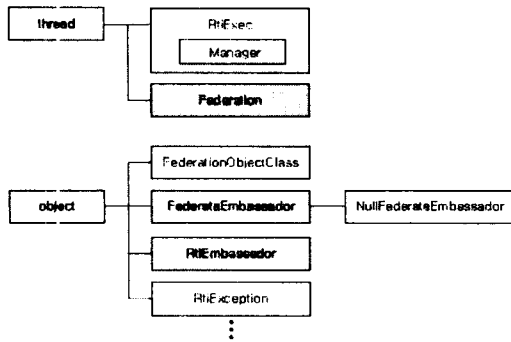


그림 2. Java/RTI의 구현

그림 2에서 보여 주는 바와 같이 불필요한 네트워크 트래픽을 줄이기 위하여 RtiExe와 FedExe는 하나의 프로세스내에서 구동되는 독립 Thread로 구현하였다. 하나의 프로세스로 구현함으로써 향후 QoS를 위해 필요한 성능 측정기 모듈의 추가를 용이하게 할 수 있다. 그림 3은 RtiExe 및 FedExec를 구현하는 프로그램의 기능 모듈을 보여 주며 기능 모듈 중에서 성능 측정자는 추가될 예정이다. FedExec를 구현하는 Thread는 Federate가 요청하는 서비스를 효과적으로 구현하는 큐 관리 기능 및 공유 객체가 변경되었을 시에 이를 전달하기 위한 메시지 라우팅 알고리즘을 구현하였

다. FED(Federation Execution Dta)는 Federation 실행 시에 필요한 구성 정보와 Federate들이 공유하는 객체에 대한 속성 정보를 기록해 놓은 데이터베이스이다.

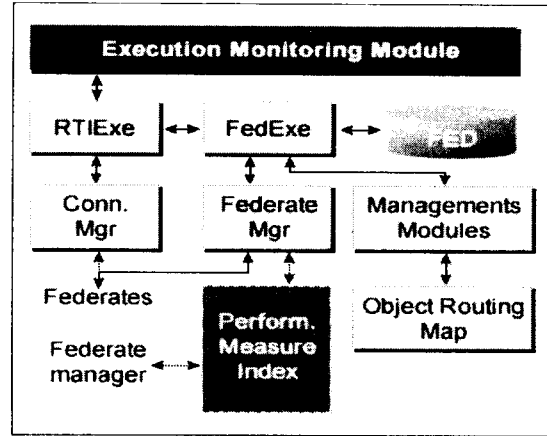


그림 3. RtiExe 구조

RTI는 Federate의 Service Request에 대해 비동기적으로 응답하는 경우가 있다. RTI가 Federate에 Notify해야 할 Event나 Object Update 정보는 FederateEmbassador Class의 Callback 함수들에 의해 Federate에 전달된다. 즉, Federate는 FederateEmbassador를 구현함으로써 RTI를 통해 다른 Federate와 통신한다. Java/RTI는 Federate가 구현해야 할 함수들의 Template와 Default 기능을 구현하여 제공한다. 그러므로, Federate는 불필요한 함수는 구현하지 않아도 된다.

IV. 응용 예 및 결과

개발된 Java/RTI의 성능과 서비스들을 시험하기 위하여 Federate들의 객체 공유 패턴에 따라 몇가지 응용 프로그램을 개발하여 시험하였다.

1) 자동차 주행 시뮬레이션

자동차들이 서로 위치 정보를 교환하면서 주행 방법 및 위치를 자율적으로 결정하는 응용 프로그램이다. 그림 4는 Federate 실행화면을 보여 준다.

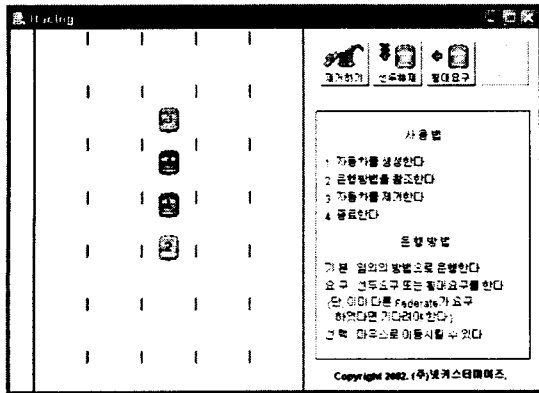


그림 4. 자동차 주행 시뮬레이션

이와 같은 응용 프로그램에서는 객체 공유를 위해 요구되는 Update가 매우 빈번하게 발생하는 특징을 가지고 있다. 즉, RTI의 서비스 빈도 수에 따른 성능 저하를 시험하는 것이 가능하다. 응용 프로그래머는 아래와 같이 간단한 데이터 속성을 FED에 기록하면 Federate간에 데이터 공유를 할 수 있게 된다.

```

; user object classes here
(class Car
  (attribute Name reliable receive) // 자동차명
  (attribute carKind reliable receive) // 종류
  (attribute posX reliable receive) // x좌표
  (attribute posY reliable receive) // y좌표
  (attribute goX reliable receive) // x축 진행방향
  (attribute goY reliable receive) // y축 진행방향
  (attribute speed reliable receive) // 속도
  (attribute head reliable receive) // 선두여부
  (attribute left reliable receive) // 횡대여부
)

```

실험 결과 10대의 자동차가 서로 위치 정보를 교환하면서 주행하는 경우에도 무리없이 RTI가 서비스를 제공하여 주었다. 이 경우 초당 약 100개의 Object Update가 발생하게 되며 Update된 내용이 Federate에 전달되기 위해서는 1000번의 네트워크 액세스가 필요하다. 네트워크의 상태 및 차량 대수의 증가 시에 발생하는 RTI의 성능 저하를 시험하기에 적합한 응용 예이다.

2) VoIP 회의 시스템

다자간 회의 시스템에서는 발제자의 발표를 다른 회의 참여자에게 방송형태로 전달하게 된다. 이러한 방송 기능은 HLA의 Interaction 기능을 사용하여 쉽게 구현하는 것이 가능하다. 그림 5는 다자간 회의 시스템의 제어부를 보여 준다.



그림 5. VoIP 응용 예

VoIP 응용 소프트웨어는 Federate사이에 전달되는 데이터의 양이 매우 큰 형태를 시험하기에 적합한 예제이며 음성의 실시간 전송을 위해 필요한 네트워크 자원의 사전 예약 및 QoS 지원 기능 시험을 위해 필요하다. 구현된 예제에서는 MP3 음악 파일을 재생하여 회의에 참여한 Federate들에게 전송하였을 때 각 Federate에 끊임없이 전달되는 것을 시험하였다.

3) 실시간 제어 응용 소프트웨어

그림 6은 실시간 제어 시스템에서 주 제어 기능을 수행하는 Federate의 실행 화면을 보여 준다. 주 제어 기능을 수행하기 위하여 각 탦의 순위 정보를 RTI를 통하여 수집하여 홍수를 방지하기 위한 수문 제어 기능을 수행한다. 주 제어 Federate의 제어 신호는 각 Federate에 실시간으로 전송되게 된다. 각 탦에 설치된 Federate는 다음과 같은 절차로 제어 기능을 수행한다.

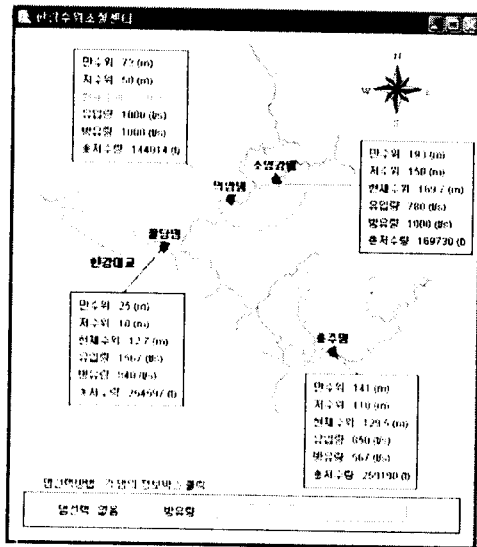


그림 6. 실시간 제어 예제

- 1) Federation에 Join : RtiExe에 Federation의 접속을 요청하고 RtiExe는 Federation이 존재하지 않으면 생성한 후에 접속 요구를 수락
- 2) Federate가 Update하는 수위 변수의 등록 : Object Management Service가 제공하는 Register 기능 호출
- 3) Federate가 필요한 제어 Signal에 대한 수신 등록 : Object Subscribe Service를 이용한 수신 등록
- 4) 수위 정보 Update
- 5) 제어 Signal 처리

Federate 프로그래머는 HLA FED 구성 요소 설정 방법과 Java/RTI가 제공하는 libRTI Interface를 이용하여 분산 제어 소프트웨어를 구현하는 것이 가능하다.

V. 결 론

본 논문에서는 HLA 규격을 Java 언어를 이용하여 구현한 Java/RTI를 소개하였다. 현재까지 개발된 RTI 환경이 대부분 CORBA를 이용하여 구현되어 소프트웨어가 대형화되어 처리 속도의 효율성이 보장되지 않는 단점이 있었다. 이를 해결하기

위하여 RTI의 주요 Management 기능을 Java로 구현함으로써 약 700Kbyte의 경량형 미들웨어를 개발하는 것이 가능했다. 개발된 Java/RTI를 자동차 경주, VoIP 솔루션, 실시간 제어 시스템에 적용함으로써 미들웨어의 성능 및 상용화 가능성을 검증하였다.

Java/RTI의 DMSO 인증 획득을 위해 필요한 적합성 시험 절차에 대한 연구가 진행 중이며 각 Federate의 논리시간을 실제 시간으로 Mapping하는 실시간 기능과 주요 분산 객체의 Update를 제한된 시간 안에 처리할 수 있도록 보장하는 QoS 기능을 추가하고 있다. 실시간 QoS 기능이 완성되면 로봇 및 분산 제어, 정보가전용 응용 제품 개발에 활용될 수 있을 것으로 예상되며 시뮬레이션 관련 기반 소프트웨어로서의 역할이 기대된다.

Acknowledgements

본 연구는 한국산업기술평가원에서 지원하는 산업기술개발사업 “실시간 QoS를 보장하는 분산 협업용 경량형 미들웨어 Java/RTI 개발”과제로 수행되었음.

참 고 문 헌

- [1] U.S. Department of Defense, *High Level Architecture Interface Specification Version 1.3*, 20 April 1998
- [2] U.S. Department of Defense, *RTI 1.3-Next Generation Programmers's Guide Version 3.2*, 28 April 2000
- [3] 임성용, 김용재, 김탁곤, “HLA를 이용한 자동차 주행의 실시간 분산 시뮬레이션“, *한국시뮬레이션학회 99년 춘계 학술대회 논문지*, p169-173, 1999년 4월
- [4] P.M. Tyma, G. Torok, T. Downing, *Java Primer Plus*, The Waite Group Press, 1996