

HLA/RTI 기반 항공교통관제 시뮬레이션의 객체 관리 설계 및 구현

김세환, 채수환
한국항공대학원 컴퓨터공학과
e-mail:fnskksk@hotmail.com

Object Management Design and Implementation for Air Traffic Control Simulation Based on HLA/RTI

Se-Hwan Kim, Soo-Hoan Chae
Dept of Computer Engineering, Graduate School, HanKuk
Aviation University

요 약

HLA는 상호운용성과 재사용성을 목적으로 미국 국방성 산하 기관인 DMSO에서 개발되어진 분산 시스템 프레임워크이다. 이러한 HLA가 민간 부분 적용을 위해 표준이 발표되었고, HLA는 여러 분야의 범용성을 가지기 위해 복잡한 구현 시스템을 가지게 된다. 특히, 객체의 속성변화를 반영하기 위해서는 많은 통신 부하가 발생한다. 이런 단점을 해결하기 위해 제안하였던 객체 관리 모듈을 이용한 항공교통관제 시뮬레이션을 구현하여 객체 관리 모듈의 성능 및 적용 분야를 평가한다.

1. 서론

HLA(High Level Architecture)는 미국 국방성 산하 기관인 DMSO(Defense Modeling & Simulation Office)의 주도하에 개발된 분산 시뮬레이션의 표준화를 위한 개념적인 분산 시스템 프레임워크로서, 응용프로그램간에 상호 운용성(interoperability), 재사용성(reuse)을 목적으로 하며, RTI(Run Time Infrastructure)라는 HLA의 하부구조로서 메시지 전송 및 객체관리 등의 시뮬레이션 진행에 필요한 서비스를 제공하는 시스템이다.

즉, HLA는 시뮬레이션의 수행중에 데이터 교환 및 운용에 대한 조율, Federate Interface Specification에 의하여 정의되어지는 서비스들의 집합을 제공하기 위한 RTI(Run Time Infrastructure) 소프트웨어를 이용하는 아키텍처이다[1].

초기 HLA는 JWAR(Joint Warfare System), JSIM(Joint Simulation System), JMASS(Joint Modeling and Simulation System)등 군방 관련 시뮬레이션을 개발에 사용되었지만, HLA에서 제공하

는 서비스는 일반적인 분산 시뮬레이션 개발에도 적용될 수 있어 최근에는 다양한 분산 시뮬레이션에서 사용되고 있다. 그러나 다양한 시뮬레이션 환경에 적용되어 상호 운용성과 재사용성을 유지하기 위해서는 시뮬레이션 자체가 점점 복잡해지게 되었다.

통합 시뮬레이션등의 규모가 큰 시뮬레이션에서는 객체의 속성 변화를 전달하기 위한 federate들 간의 데이터 교환 및 객체와 속성을 관리하기 위한 초기화 및 종료 절차에서 많은 통신 부하가 발생한다. 이 통신부하의 주원인은 시뮬레이션 수행중 객체의 속성 변화 전송시 값의 변화가 없는 속성을 포함하여 전달하는 경우가 발생하기 때문이다.

본 논문에서는 이러한 문제를 해결하기 위해서 기존에 제안하였던 객체 관리 모듈을 사용한 항공교통관제 시뮬레이션을 구현하고 객체 관리 모듈을 통한 통신 부하 감소 및 적용 분야를 평가한다.

2. 관련연구

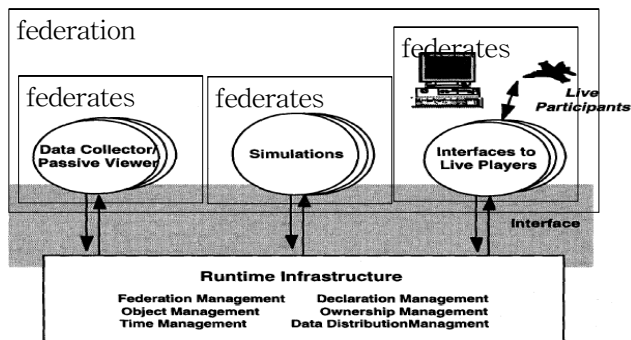
2.1. HLA를 이용한 시뮬레이션 구성

HLA는 시물레이션들을 더 큰 시물레이션으로 통합가능하게 하는 아키텍처이다. 예를 들어 항공기 기종별 조종을 위한 시물레이션과 관제사들을 훈련하기 위한 항공교통관제 시물레이션을 HLA를 이용해 각각 통합한다면, 더 큰 규모의 항공 관련 시물레이션이 가능하다. (그림 1)에 보인 바와 같이 통합되어진 항공 관련 시물레이션을 federation이라한다. 또한 federation에 참가하는 각 요소들을 federate라 부른다. 이런 시물레이션을 구현하기 위해서 필요한 요소는 federation과 federate들이 시물레이션 수행 중 지켜야하는 HLA Rules, federate들 사이에서 정의되어진 기능적 인터페이스의 설계인 Interface Specification(RTI), 시물레이션을 기술하는 문서화인 OMT 3가지가 있다[2][3][4][5].

2.2. RTI 서비스

(그림 1)에서와 같이 federation의 제어 관리, 각 federate들의 데이터 교환등을 위하여 아래의 6가지 서비스를 지원한다[4].

- (1) Federation Management : federation 수행의 생성, 제어 관리, 수정, 삭제를 관리하는 서비스 집합
- (2) Declaration Management : 참가한 federate들이 사용할 정보의 생성 및 전달을 관리하는 서비스 집합
- (3) Object Management : 참가한 federate들이 사용한 객체의 등록, 수정, 삭제 및 interaction의 교환을 관리하는 서비스 집합
- (4) Ownership Management : 참가한 federate들 사이의 속성에 대한 소유권 이동을 관리하는 서비스 집합
- (5) Time Management : federation 수행중의 시간행을 관리하는 서비스의 집합
- (6) Data Distribution Management : 참가한 federate들 통신구역을 정의하는 서비스의 집합



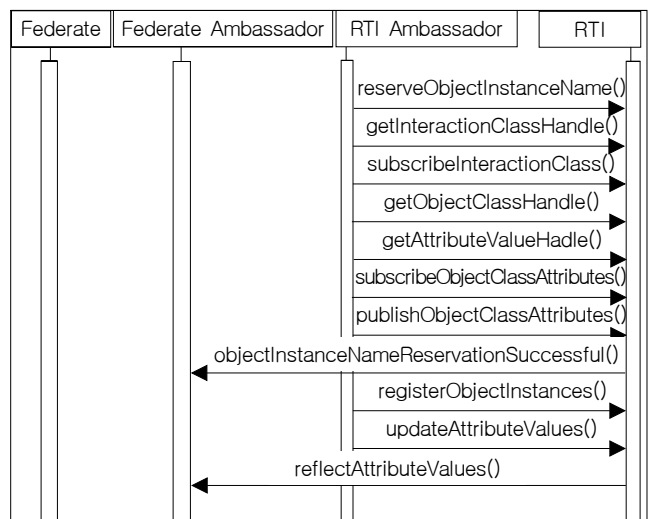
(그림 1) HLA federation의 구조

2.3. 객체 관리 모듈(object management module : OMM)

HLA를 이용한 시물레이션에서 객체를 사용하기 위해서는 (그림 2)과 같은 절차를 수행해야 한다. 특히, UpdateAttributeValues()를 통한 federate들 간의 속성의 값 전달시 불필요한 데이터가 발생하게 된다. (그림 3)과 같은 해당 객체의 속성들을 모두 묶은 구조의 AttributeHandleValueMap의 데이터 형을 사용하기 때문이다. 만약 해당 객체의 속성 값으로 x, y, z의 3차원 위치 좌표를 가진다면, 해당 객체가 x 좌표만 이동하여도 좌표를 모두 전송해야한다. 이 경우 참여한 federate의 수가 많을 경우 발생하는 통신 부하는 큰 폭으로 증가한다. HLA에서 사용되는 통신 방법은 peer-to-peer 방식이 아닌 broadcasting 방식을 사용하기 때문이다. 즉, 참여한 federate의 수가 많을 경우 발생하는 통신 부하는 배수로 증가한다.

이 문제를 해결하기 위해 객체 관리 모듈을 제안하였다. 이 모듈은 (그림 4)과 같이 구성되어 있다. HLA core와 객체 관리 모듈 사이에 공통적으로 속성의 식별자(id)만을 유지하고 HLA core는 객체에 대한 정보를 포함하지 않는다. 즉, (그림 2)의 object class, object instance의 선언 부분이 HLA core에서 삭제되어진다. HLA core는 기존의 역할을 모두 수행하지만 객체 관리에 관한 권한만을 객체 관리 모듈에 이양한다.

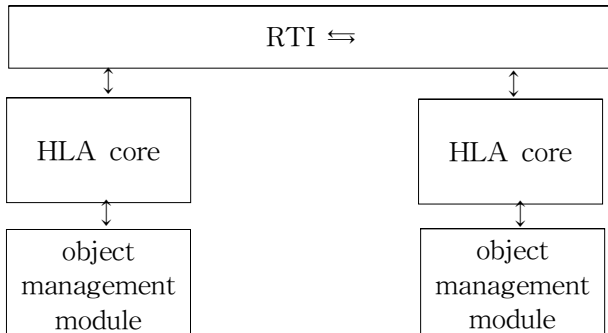
기존 HLA의 구현에서 AttributeHandleValueMap을 사용하여 해당 객체의 속성을 일괄적으로 전송하던 방식의 통신 부하를 줄이기 위해서 식별자(id)를 이용한다. interaction(id, 변화된 속성) 형태의 통신을 이용하여 변경된 속성만을 전달하게 된다[6].



(그림 2) HLA 객체의 정의 및 통신 순서

handle	-> add	(x_point	y_point	z_point)
	

(그림 3) AttributeHandleValueMap의 구조



(그림 4) object management module의 구성

3. 항공교통관제 시뮬레이션

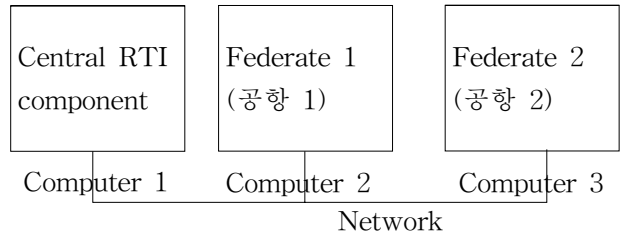
3.1. 항공교통관제 시스템

객체 관리 모듈을 적용하기 위한 분야로 항공교통관제 서비스를 선택하였다. 항공교통관제 서비스는 비행장(공항)을 중심으로 일정범위의 공역(통상 관제권이라 함) 및 공항지상이동지역 내에서 관제탑 관제사에 의해 제공되는 관제업무로서 공항이동지역(Airport Movement Area)의 감시(Surveillance) 및 통제, 이·착륙 및 관제권통과 항공기에 대한 비행허가발부, 항공교통관제허가의 중계(ATC Clearance Delivery) 등을 포함한다.[7] 본 시뮬레이션에서는 항공기를 객체로 정의하여 항공기의 이륙부터 착륙까지의 과정을 다룬다. 실제 항공교통관제 시스템의 절차는 비행전(preflight), 이륙(takeoff), 출발(departure), 항공로(enroute), 강하(descent), 접근(approach), 착륙(landing)의 7단계로 이루어져 있지만 객체생성, 비행, 도착 세단계로 기존의 단계를 포함한다.

3.2. 항공교통관제 시뮬레이션의 구성

본 연구에서는 객체 관리 모듈을 사용하여 항공교통관제 시뮬레이션을 구현한다. 기본적인 시뮬레이션의 구성은 3개의 노드(3개의 federate로 구성)로 구성되어 있다.

Central RTI component는 전체 federation의 모니터링을 수행한다. federate 1, 2는 HLA core와 객체 관리 모듈을 포함하며 실질적인 시뮬레이션을 수행한다. 즉, 항공기에서 발생되어지는 정보를 이용하여 항공교통관제를 진행하는 각각의 공항에 해당한다.



(그림 5) 시뮬레이션 시스템 구성

federate 1, 2의 객체 관리 모듈을 살펴보면 실질적인 시뮬레이션의 주체인 항공기 객체의 생성 파괴 항공기 속성의 정의, 속성값의 송수신을 포함하고 있다. 항공기 객체는 모두 10개의 속성을 가지는 class Airplane으로 정의되어 있다.

```

class Airplane {
int    id;           //항공기 id
char   dep_airport; //출발 공항 정보
char   land_airport //도착 공항 정보
char   machine;     //항공기의 기체 정보
int    no_passenger; //탑승 승객 수
char   dep_time;    //출발 시간
char   land_time;   //도착 시간
float  point_x;     //항공기 좌표 정보 x
float  point_y;     //항공기 좌표 정보 y
float  point_z;     //항공기 좌표 정보 z
...
}
    
```

(그림 6) class Airplane의 구성

3.3. 항공교통관제 시뮬레이션의 진행

시뮬레이션의 진행은 기본적으로 객체(항공기)의 생성, 비행, 도착 3단계로 수행된다. 객체의 생성시(그림 6)의 정보를 모두 포함한 Airplane의 instance가 만들어진다. 비행 이후 x, y, z의 좌표 정보는 수시로 변경되어진다. 하지만 이외의 7개의 정보들은 객체 생성 단계와 동일하다. 즉, 객체 관리 모듈(object management module)을 적용하면 값의 변화가 있는 정보만을 전송한다. 반면 객체 관리 모듈을 적용하지 않을 경우 HLA의 통신 특성상 모든 정보를 매번 전송해야 한다. 각 단계에서 발생되어지는 통신에 포함되는 속성의 수는 <표 1>과 같다.

<표 1> 통신에 포함되는 속성의 수

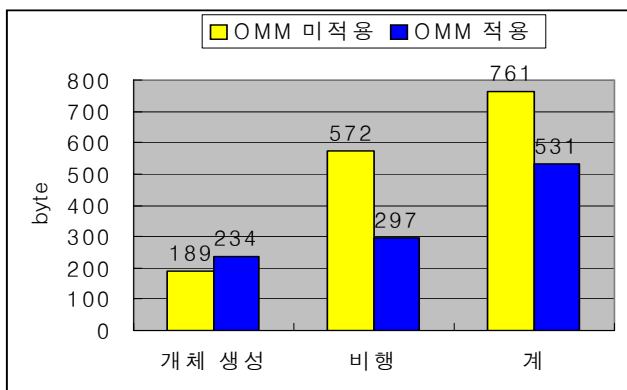
구분	객체 생성	비행	도착
OMM 적용	10개	3개	없음
OMM 미적용	10개	10개	없음

시뮬레이션에서 사용하는 객체의 수는 100개, 각 객체 생성 단계에서 1회, 비행 단계에서 4회의 통신이 federate간에 발생하도록 설정하였다.

6. 시뮬레이션 결과 및 분석

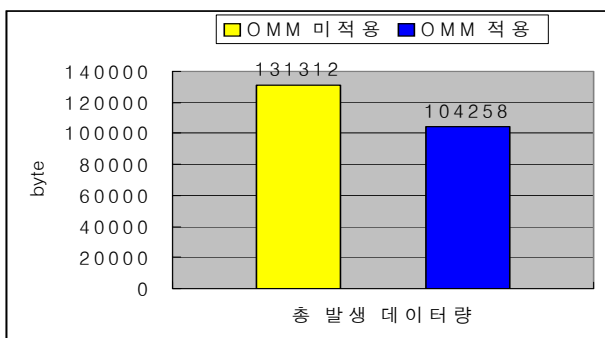
객체 생성, 비행 단계에만 정보 전송이 이루어짐으로 이 두 단계에서 발생되어지는 통신 데이터 발생량을 각각 비교하고 시뮬레이션 종료까지의 전체적인 통신 데이터 발생량을 비교하여 객체 관리 모듈적용의 결과를 조사했다.

객체 별로 객체 생성, 비행 각 단계에서 발생하는 통신 데이터 발생량과 객체 별 총 통신 데이터 발생량을 비교하였다. (단위는 byte)



(그림 7) 객체 별 통신 데이터

객체 생성단계의 결과는 오히려 객체 관리 모듈을 적용한 경우 더 많은 통신 데이터가 발생하였다. 그 이유는 처음 객체 생성시 10개의 모든 속성을 상대 federate에 전달해야하기 때문에 오히려 통신회수가 증가하기 때문이다. 하지만 비행단계에 들어서는 변화되는 속성은 x, y, z 위치 속성 3가지만 변화하기 때문에 전달해야 하는 통신 데이터량은 급격히 감소한다. 그 결과로 한 객체가 시뮬레이션 기간동안에 전달해야하는 최종 통신 데이터량은 객체 관리 모듈을 사용한 경우가 감소한다.



(그림 8) 시뮬레이션 기간동안 발생한 데이터

시뮬레이션 수행동안 발생한 통신 데이터량은 객체 관리 모듈을 적용한 경우에 더 적게 발생하였다.

7. 결론 및 향후 연구

본 논문에서는 기존에 제안하였던 객체 관리 모듈을 항공교통관제 시뮬레이션에 적용하여 통신 부하를 감소 시켰다. 이는 항공교통관제와 같은 시뮬레이션의 대상이 되는 객체의 속성 변화가 적은 경우 객체 관리 모듈을 적용하면 성능을 향상시킬 수 있음을 보여준다. 이 결과는 대규모 시뮬레이션 내 객체 속성의 변화가 많지 않은 경우 통신 부하를 효과적으로 감소시킬 수 있음을 보여준다.

앞으로 객체 관리 모듈을 적용 가능한 시뮬레이션 분야의 조사와 Data Distribute Management등의 서비스와 연계하여 HLA 기반의 분산 시뮬레이션의 성능 향상을 연구 할 계획이다.

<<Acknowledgement>>

본 논문은 산업자원부 한국산업기술평가원 지정 한국항공대학교 부설 인터넷정보검색 연구센터의 지원에 의함.

참고문헌

- [1] Dr. Judith S.Dahmann "High Level Architecture for Simulation" Distributed Interactive Simulation and Real Time Applications, 1997., First International Workshop on , 9-10 Jan. 1997
- [2] Dr. Frederick Kuhl, Dr. Richard Weatherly, Dr. Judith Dahmann "Creating Computer Simulation Systems" Prentice Hall 1999
- [3] IEEE 1516 "Standard for Modeling and Simulation High Level Architecture - Framework and Rules" IEEE 2000
- [4] IEEE 1516.2 "Standard for Modeling and Simulation High Level Architecture - Federate Interface Specification" IEEE 2000
- [5] IEEE 1516.1 "Standard for Modeling and Simulation High Level Architecture - Object Model Template Specification" IEEE 2000
- [6] 김세환, 채수환 "HLA/RTI 기반 분산시뮬레이션의 객체 관리 성능향상" 한국정보처리학회 2005 춘계학술대회논문집. 2005
- [7] 건설교통부 한국안전본부 "항공교통관제업무" http://www.casa.go.kr/safe/safe1_6a.asp