한국시뮬레이션학회 '99춘계학술대회 논문집 1999.4.17 동의대학교

HLA를 이용한 자동차 주행의 실시간 분산 시뮬레이션

임 성 용, 김 용 재, 김 탁 곤
한국과학기술원
전기 및 전자공학과
시스템 모델링 시뮬레이션 연구실
URL: http://smsl.kaist.ac.kr
E-mail : {sylim@smslab, yjkim@smslab, tkim@ee}.kaist.ac.kr

HLA(High Level Architecture)는 시뮬레이션 모델간의 상호 운용성(Interoperability)을 높이기 위해 제안되었고, 분산되어 있는 시뮬레이션 모델간의 데이터 교환 및 시뮬레이션 시간 진행을 조정하는 등 여러 가지 장점을 가진 분산 시뮬레이션에 적합한 구조이다. 본 논문에서는 자동차 주행 시뮬레이션을 HLA를 기반으로 구현하고, HLA의 실시간 분산 시뮬레이션에 대한 적용 가능성을 확인한다.

1. 서론

분산 시뮬레이션은 시뮬레이션 모델을 여러 컴퓨터에 나누어 실행하여 전체 시뮬레이션 수행 시간을 줄이는 방법이다. 이 때, 분산된 시뮬레이션 모델들은 동일한 시뮬레이션 언어로 구현된 모델로 제한된다.

반면에, HLA는 분산된 시뮬레이션 모델들간의 상호 운용성을 보장하기 위해서 미국 국방성에서 제안하였다[1, 2, 3]. 특히, HLA는 전쟁 게임(War Game)이나 훈련 교육(Training)등과 같이 시뮬레이션 모델과 사람이 상호작용(Interaction)할 수 있는 실시간 시뮬레이션에 적합한 환경이다. 또한, HLA는 이기종 시뮬레이터간의 연동시에 기본 구조로 사용될 수 있고[4], 실시간 분산 처리에도 잘 응용될 수 있다.

자동차 주행 시뮬레이션의 경우 도로에서 운행중인 자동차간의 상호작용을 효과적으로 표현하는 것은 매우 어려운 일이지만, 사람을 포함한 실시간 시뮬레이션으로 구현하면 자동차간의 상호작용을 보다 잘 표현할 수 있다. 본 논문에서는 HLA를 이용하여 자동차 주행 시뮬레이션을 구현하고, HLA의 실시간 분산 시뮬레이션에 대한 적용 가능성을 확인한다.

2. HLA 개요

HLA는 분산된 시뮬레이션 모델들간의 상호 운용성을 높이기 위해 제안되었다. 이 때, 시뮬레이션 모델(또는 노드)을 federate라 부르고 federate들의 조합을 federation이라 한다. HLA는 HLA Rules[1], Interface Specification[2], Object Model Template[3]의 세 가지로 정의된다. HLA Rules는 각 federate와 RTI(Run-Time

Infrastructure)의 책임을 정의한 규칙이다. Interface Specification은 federate와 RTI간의 기능적 인터페이스를 여섯 가지의 서비스로 나누어 기술한다. Object Model Template는 federate들 사이에 교환되는 데이터를 기술하는데 사용된다.

RTI는 HLA에서 정의된 시뮬레이션 하부구조로 각 federate들의 데이터 교환 및 시뮬레이션 시간 진행에 필요한 여러 가지 기능을 제공한다. 특히, 시뮬레이션 시간을 네가지로 분류하여 서로 다른 시간 진행 방법을 사용하는 federate들도 상호 운용할 수 있는 장점이 있다. 그림 1 에는 RTI의 6가지 서비스를 시뮬레이션 진행 단계별로 나타낸 것이다.

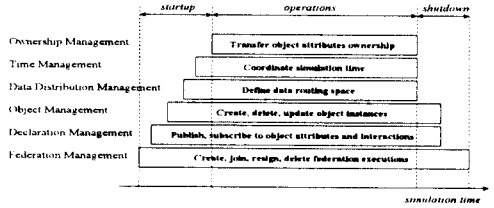

그림 1 시뮬레이션 단계별 RTI 서비스

2.1 기존 분산시뮬레이션과의 차이점

HLA/RTI를 이용한 분산 시뮬레이션은 기존의 분산 시뮬레이션과 여러 면에서 다르다. 먼저, 기존의 분산 시뮬레이션은 속도 향상이 중요한 목표이지만, HLA는 시뮬레이션 모델들간의 상호 운용성을 높이기 위한 것이다. 둘째로, 기존 분산 시뮬레이션은 동일한 시뮬레이션 언어로 구현된 모델만을 사용하지만, HLA는 이 기종 시뮬레이션 모델들도 서로 연동될 수 있는 기반을 제공한다 [4]. 또한, 기존의 분산 시뮬레이션은 모델간의 연결이 정적으로 결정된 시스템에 잘 적용되지만, HLA는 동적으로 모델간의 연결이 변하는 시스템을 잘 표현할 수 있다.

3. 자동차 주행 시뮬레이션

본 논문에서 구현하고자 하는 자동차 주행 시뮬레이션은 단순화된 지능형 교통 관제 시스템(IVHS)으로, 임의로 생성된 자동차들이 플래툰(platoon)을 이루는 과정만을 구현한다. 이때, 컴퓨터에 의한 자동 운전과 사용자에 의해 운전되는 두 가지 전략으로 나눈다.

3.1 자동 주행 전략

자동 주행 전략은 플래툰의 선두 차량에 대한 추적으로 시작된다. 이때, 추적방법에는 여러 가지가 있겠지만, 본 논문에서는 전방 차량에 대한 추적만으로 제한한다. 즉, 자동 주행 차량은 항상 바로 앞의 차량만을 추적하여 상대적인 위치를 감소시키는 방향으로 주행하는 방법이다.

3.2 사용자 주행 전략

운전자의 실시간 입력에 따라 자동차의 좌우 위치 변환과 전후 속도 조절이 가능한 주행 전략이다. 몇 가지 추적 방법에 따라 조건에 맞는 차량을 인식하고 자동적인 위치 및 속도 변화에 의해 목적 자동차를 추적하게 된다. 그림 2와 그림 3에는 이러한 추적의 예를 나타내었다.

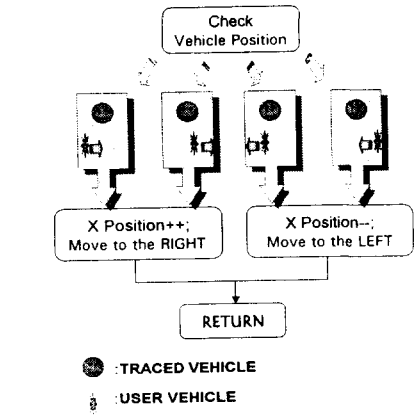

그림 2 X 위치 수정

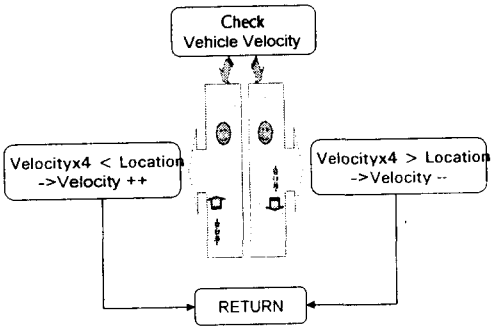

그림 3 속도 수정

4. 시스템 구현

자동차 주행 시뮬레이션 모델을 RTI V1.3R4 [5]를 이용하여 WindowsNT 환경에서 구현하였다. 각 federate는 자신의 자동차 객체를 생성하고, 다른 federate들로부터 받은 차량 정보를 이용하여 GUI를 통해 현재의 상황을 표시한다.

4.1 RTI Interface Sequence

각 federate는 federation에 참여(Join)한 후 시뮬레이션을 수행하고, 시뮬레이션이 끝나면 federation에서 탈퇴(Resign)한다. 이때, federation의 생성(Create)은 처음으로 참여를 요청한 federate에 의해 이루어지고, 소멸(Destroy)은 마지막으로 탈퇴한 federate에 의해 이루어진다.

federation에 참여한 후, 각 federate는 자기 차량의 속도와 위치에 대한 접근(Publish) 및 다른 federate가 관리하는 차량의 정보에 대한 구독(Subscribe)을 선언한다. 각 federate에서 생성된 차량에 대한 정보는 RTI를 통해 다른 federate로 전송된다.

시뮬레이션 시간에 대한 정의는 각 federate의 Time regulating과 Time constrained의 두 가지 요소에 의해 결정된다. Time regulating은 federate의 시간이 다른 federate의 시간 결정에 영향을 미침을 나타내고, Time constrained는 federate의 시간이 다른 federate의 시간에 영향을 받음을 나타낸다. 본 논문에서는 자동 주행 차량과 사용자 주행 차량이 동시에 존재하는 실시간 시뮬레이션이므로, Time regulating과 Time constrained는 모두 FALSE가 된다.

HLA/RTI에서는 전장(Battle Field)과 같은 시뮬레이션 영역에 대해 관심있는 부분만을 지정하여 데이터 전달을 최소화 할 수 있는 Data Distribution Management가 지원되지만, 본 논문에서는 시뮬레이션 영역인 도로 상황에 대해 간단한 모형을 사용하므로 다루지 않는다.

4.2 Thread 통신

자동차의 주행과 데이터의 교환, 그리고

사용자의 입력을 실시간으로 처리하기 위해서 federate를 두 부분으로 나누어 구현하였다. 먼저 Main Thread에서는 자동차의 생성 및 초기화를 거쳐서 자동차 주행에 따르는 화면의 위치변화를 표시하고, 사용자의 입력을 받아 Thread간의 매개 변수인 Event에 전달한다. Simulation Thread에서는 사용자 입력에 따르거나 혹은 추적 알고리즘에 의한 위치 및 속도 변화를 처리한다.

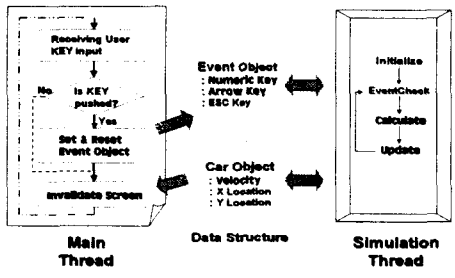

그림 4 Thread 통신

4.2.1 Main Thread

Main Thread에서 수행하는 작업 중 하나는 키보드 혹은 마우스로부터 메시지의 생성을 확인하여 메시지의 종류에 따라 대응되는 Event를 Set하고 그 외의 Event들은 모두 Reset하는 작업이다. Event는 Multithread Programming에서 Thread간의 통신을 담당하는 Flag변수 중 하나로 Event 내부의 boolean값을 변화시키는 함수로 Set()과 Reset()을 사용한다. 나머지 작업은 실시간으로 변화하는 자동차들의 위치 및 속도를 화면에 표시하는 과정이다.

4.2.2 Simulation Thread

먼저 Event변수를 입력으로 받아 들여 설정되어 있는 Event변수를 체크하고, 자동차의 주행 전략에 대한 정보를 얻게 된다. 이에 따르는 추적 알고리즘이나 사용자 입력에 의한 위치 및 속도 변화를 수행하고, 데이터의 수정결과는 방벽, 혹은 다른 자동차와의 충돌 검증 작업을 마친 후 다른 federate들에게 Update 된다.

5. 실행 결과

실험은 ATM 네트워크로 연결된 4대의 컴퓨터에서 수행하였다. federate는 각 컴퓨터에 하나씩 할당되었다. 그림 5는 4대의 자동차가 초기화되어 입력을 기다리고 있는 상태이다. 화면의 우측 상단에는 다른 자동차들에 대한 정보가 표시되고 우측 하단에는 마우스를 사용하는 버튼이 존재한다.

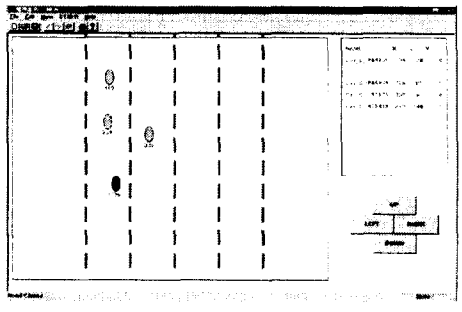

그림5 임의선택 초기상태

그림 6에서는 사용자의 입력에 의해 설정된 주행 상태에서 가장 후방에 위치한 차량이 전방차량에 대한 측면으로의 추적을 실행한 결과이다.

그림 7은 자동 주행 차량들이 추적을 시행하여 4대의 자동차가 플래툰을 형성하고 있는 상황이다.

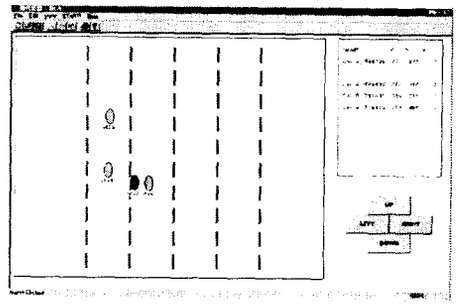

그림6 전방측면추적

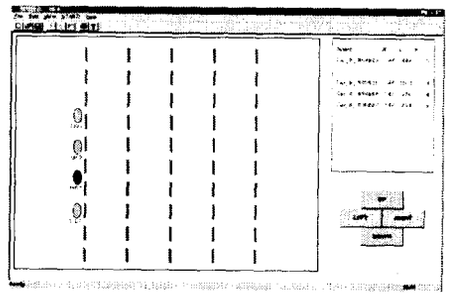

그림7 추적에 의한 플래툰 형성

6. 결론 및 추후 과제

본 논문에서는 HLA의 응용으로써 간단한 자동차 주행 시뮬레이션을 수행하였다. HLA는 여러 대의 자동차들이 네트워크 상에서 실시간 주행 시뮬레이션을 수행하는데 많은 이점을 가지고 있다. 먼저, HLA는 각각의 자동차들이 시뮬레이션 영역에 존재하는 다른 자동차들과 정보를 공유 할 수 있는 상호 데이터 교환 구조를 제공한다. 두 번째로는 시뮬레이션이 진행되는 동안 보수적인 시간 동기 방법을 제공하여 시뮬레이션 노드간의 시간 진행에 필요한 동기방법을 직접 제어하지 않고 HLA/RTI의 함수들만을 조합하는 것으로 네트워크 상에서의 시뮬레이션을 가능하게 한다.

본 논문에서 구현된 자동차 주행 시뮬레이션에서는 도로의 변화가 없는 일직선 도로를 가정하였으나, 좀 더 현실감있는 시뮬레이션을 위해서 도로 상황에 대한 데이터베이스를 구축, 적용하여야 한다.

참고 문헌

[1] DMSO, "HLA Rules version 1.3", 1998 IEEE Standards Draft, February 5, 1998

[2] DMSO, "HLA Object Model Template Specification version 1.3", 1998 IEEE Standards Draft, April 20, 1998

[3] DMSO, "HLA Interface Specification version 1.3", 1998 IEEE Standards Draft, April 20, 1998

[4] Y. J. Kim and T. G. Kim, "A Heterogeneous Simulation Framework based on the DEVS BUS and the High Level Architecture", *Winter Simulation Conference 98*, pp. 421-428, Dec. 13-16, 1998, Washington D.C., U.S.A.

[5] DMSO, "HLA Run-Time Infrastructure Reference Manual Version 1.3", March 27, 1998