

HRTI 기반의 계층적 페더레이션을 나타내는 수학적인 방법에 대한 연구

안정현*, 김탁곤*

A Formal Approach to describing for Hierarchical Federations in HRTI

Jung Hyun Ahn, Tag Gon Kim

Abstract

High Level Architecture (HLA) is a standard architecture for interoperation between heterogeneous simulations, and Runtime Infrastructure (RTI) is the implementation of the HLA interface specification. A limitation of the current HLA is that it allows only single federation and the federation has one Federation Object Model (FOM) which is merged with different organizations. In addition, to develop large and complex systems, appropriate formal methods need to model the systems. However, HLA allows each simulator to interoperate with the context of a flat federation and does not specify how a hierarchical system architecture organizes. To overcome such limitations, we propose a FOM Structure formalism to describe the hierarchical structure of federations-related management of FOM information. Hierarchical Runtime Infrastructure (HRTI) framework provides a general approach to supporting multiple federations with hierarchical structure in any depth with being compliance with HLA interface. The design and the experimental results show that it may facilitate the practical usage of simulation interoperations by ensuring the formal method, FOM Structure formalism, approaching while the HRTI framework complies the existing HLA standards.

Key Words : HRTI Framework, HLA/RTI, FOM Structure Formalism, Hierarchical Federations

* 한국과학기술원 전자전산학과, 시스템 모델링 시뮬레이션 연구실

1. 서론

HLA(High Level Architecture)는 여러 가지 다른 타입의 시뮬레이터들의 연동을 지원하기 위해 IEEE 1516 표준으로 채택된 아키텍처이다. HLA는 시뮬레이션 시스템의 구성 요소, 디자인 룰, 인터페이스 등에 관한 전반적인 아키텍처를 제공한다 [1]. 여기에서 정의된 명세를 구현한 소프트웨어가 RTI(Run Time Infrastructure)이다. RTI는 HLA 페더레이션의 수행 과정 동안 일반적인 서비스 인터페이스를 제공해주는 분산된 소프트웨어 모음이라고 할 수 있다. RTI의 서비스는 HLA 페더레이트 인터페이스 명세에 기술되어 있다 [2].

HLA의 주된 목적은 시뮬레이션간의 상호연동성과 시뮬레이션 컴포넌트의 재사용성을 높이기 위한 것이다. 페더레이트 인터페이스 명세를 따름으로 시뮬레이션간의 상호연동성을 높일 수 있다. 또한, 개별 모델간에 주고 받는 데이터를 명세한 객체 모델 템플릿(Object Model Template)의 사용과 이를 기반으로 개발된 페더레이트 프로그램은 시뮬레이션 컴포넌트의 재사용성을 높일 수 있다.

현재의 HLA는 단일 페더레이션만 지원하고 여러 페더레이션들이 상호연동할 수 있는 계층적인 페더레이션을 HLA는 지원하지 않는다. 따라서 한 페더레이션에 참여된 페더레이트는 다른 페더레이션에 참여된 페더레이트와 서로 작용할 수 없다.

이러한 제한점과 더불어, 단일 페더레이션은 계층적인 구조를 가지는 복잡한 시스템을 모델링하는데 적합하지 않다 [3, 4]. 시스템을 계층적으로 표현하는 것은 크고 복잡한 시스템을 모델링하는 필연적인 요소이다. 계층적 구조를 가지는 모델을 RTI에 적용하기 위해서는 모든 페더레이트를 단일 페더레이션으로 배치해야 한다. 이러한 배치는 잠재적으로 기존의 계층성을 파괴하고 보안 및 재사용성에 관련된 문제를 야기시킨다 [5, 6]. 따라서 본 논문에서

서는 계층적인 페더레이션을 명세할 수 있는 FOM Structure 형식론을 제안하고 이를 지원하는 틀인 HRTI 프레임워크를 구현한다. 이는 페더레이션 레벨에서의 상호운용성과 재사용성을 제공하는 일반적인 틀을 제공하는 것이다.

본 논문의 구성은 다음과 같다. 2장에서 FOM Structure 형식론과 결합 알고리즘에 대해서 살펴보고 3장에서 전체 HRTI 프레임워크에 대해서 살펴본다. 4장에서 예제로 HRTI의 효용성을 입증하고 5장에서 결론을 맺는다.

2. 계층적 페더레이션을 위한 FOM Structure 형식론

FOM Structure 형식론은 계층적으로 페더레이션을 구성하는데 있어서 필요한 페더레이션의 계층적 관계를 정의하고 이들 사이에 전달되는 외부 FOM 및 내부적으로 재구성되는 내부 FOM의 변환을 명세하는 형식론이다. 이를 수학적으로 표현하면 다음과 같다.

$$H = \langle F, eFOM, iFOM, F-FOM, C, T \rangle$$

F : 페더레이션 집합

$eFOM$: 외부 FOM 집합

$iFOM$: 내부 FOM 집합

$F-FOM = \{(i, eFOM_i, iFOM_i) | i \in F, eFOM_i \in eFOM, iFOM_i \in iFOM\}$: 페더레이션 -eFOM/iFOM 연관 집합

$C \subseteq F \times F$: 내부 결합 관계

$T : eFOM \times iFOM \rightarrow iFOM$: 내부 FOM

상태 천이 함수

페더레이션 집합(F)은 계층적인 페더레이션을 구성하는 페더레이션의 집합이다. 외부 FOM($eFOM$)은 각페더레이션 프로세스가 RTI의 createFederationExecution() API를 실제 수행될 때 쓰이는 FOM 인수이다. 이는 페더레이션의 계층 관계에 따라 상위의 페더레이션에 전달

된다. 내부 FOM(iFOM)은 페더레이션별 정보의 전달을 제어하기 위한 FOM 집합이다. 이는 외부에 노출되지 않고 내부적으로 가지는 FOM 정보로 항상, 하위의 페더레이션의 내부 FOM의 합집합이 된다. 페더레이션-eFOM/iFOM 연관 집합(F-FOM)은 페더레이션-FOM 연관 집합으로 페더레이션과 내/외부 FOM들과의 1:1 연관 관계를 이루고 있는 집합이다. 계층적인 페더레이션은 전체적으로 트리 구조를 띠고 있는데 이들 사이의 결합 관계를 표현한 것이 내부 결합 관계(C)이다. 이 내부 결합 관계를 가지고 각 페더레이션은 외부 FOM을 상위의 페더레이션에 전달 할 수 있다. 페더레이션끼리 결합 관계를 나타 낼 때에는 그 계층 사이의 레벨 차이가 1일 때만 가능하다. 내부 FOM 상태 천이 함수(T)는 실제 RTI에 의해서 페더레이션 프로세스가 생성/소멸 될 때 이들이 가지고 있는 외부 FOM이 자신 및 자신의 상위에 있는 내부 FOM을 변환하는 함수이다.

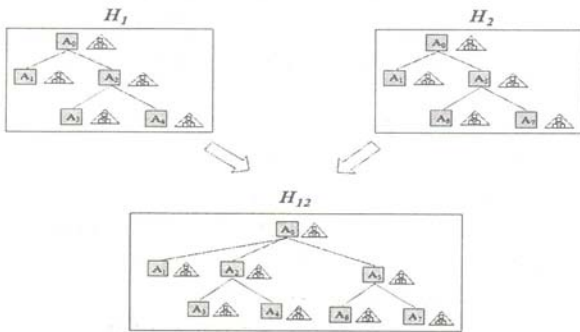


그림 1 결합 알고리즘 예제

FOM Structure 형식론으로 표현된 계층적 페더레이션은 독립적으로 하나의 완벽한 시스템이 된다. 또한, FOM Structure 형식론으로 표현된 계층적 페더레이션은 계층적 페더레이tus끼리의 결합이 허용된다. 결과적으로 결합된 계층적 페더레이션은 결합(Composition)에

달혀있다 [정리 1]. 계층적인 계층적 페더레이션의 결합은 더욱더 큰 계층적 페더레이션을 만들 수 있고 이는 기존의 계층적 페더레이션의 재사용성을 높일 수 있다. 그림1은 H1과 H2의 결합한 결과인 H12를 보여준다.

정리 1. FOM Structure 형식론은 계층적 페더레이션의 결합관계(\circ)에 대해서 닫혀 있다. 즉, FOM Structure 형식론으로 표현된 어떤 계층적 페더레이션끼리 결합한 것은 계층적 페더레이션으로 나타낼 수 있다.

증명) FOM Structure 형식론으로 표현된 $H_1 = \langle F_1, eFOM_1, iFOM_1, F-FOM_1, C_1, T_1 \rangle$, $H_2 = \langle F_2, eFOM_2, iFOM_2, F-FOM_2, C_2, T_2 \rangle$ 이 있을 때 $H_{12} = H_1 \circ H_2$ 일 때 H_{12} 가 $H = \langle F, eFOM, iFOM, F-FOM, C, T \rangle$ 에 있음을 보임으로 정리 1에 대해서 증명한다. $F_1 \circ F_2 = F_1 \cup F_2 = F_{12} \subseteq F$ 이고 $eFOM_1 \circ eFOM_2 = eFOM_1 \cup eFOM_2 = eFOM_{12} \subseteq eFOM$ 이다. $C_1 \circ C_2 = C_{12} = \{(i, j) | \text{each } i \in F_{12}, j \in F_{12}, i-j = 1\}$, $C_{12} \subseteq C$ 이다. $T_1 \circ T_2 = T_1 \oplus T_2 = T_{12}$ 이다. 즉, T_{12} 의 입력 인수가 $eFOM_{12} \subseteq eFOM_1$ 이라면 T_{12} 는 T_1 함수에 매핑되고 $eFOM_{12} \subseteq eFOM_2$ 이라면 T_{12} 는 T_2 함수에 매핑된다. $iFOM_1 = T_{12}(eFOM_1, iFOM_1) = T_1 \oplus T_2(eFOM_1, iFOM_1) = T_1(eFOM_1, iFOM_1)$ 이고, $iFOM_2 = T_{12}(eFOM_2, iFOM_2) = T_1 \oplus T_2(eFOM_2, iFOM_2) = T_2(eFOM_2, iFOM_2)$ 이 된다. 따라서 $iFOM_1 \circ iFOM_2 = iFOM_1 \cup iFOM_2 = iFOM_{12} \subseteq iFOM$ 이고 $F-FOM_1 \circ F-FOM_2 = F-FOM_1 \cup F-FOM_2 = F-FOM_{12} \subseteq F-FOM$ 이다.

그러므로 FOM Structure 형식론으로 표현된 계층적 페더레이션은 결합관계(\circ)에 대해서 닫혀 있다.

3. Hierarchical RTI (HRTI) 프레임워크 구현

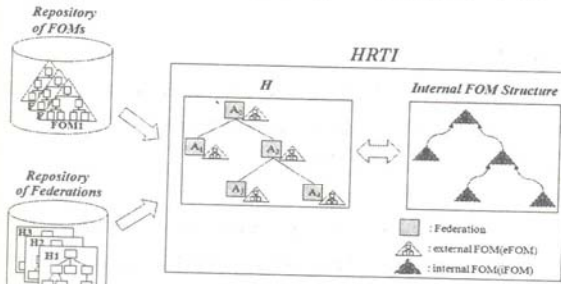


그림 2 HRTI 프레임워크

그림 2와 같이 HRTI 프레임워크는 내부적으로 페더레이션의 계층적인 구성을 한다. 또한 이들 사이에 전달되는 외부 FOM 정보로 내부 FOM을 구성하고 변환시킨다. 기존에 개발되었던 계층적인 페더레이션 및 이들의 외부 FOM을 저장소에 둔다. 이런 저장된 계층적 페더레이션을 이용하여 새로운 계층적 페더레이션을 구성할 수 있다. HRTI 프레임워크는 페더레이션 레벨에서의 상호 운용성과 재사용성을 제공하는 일반적인 틀을 제공한다.

HLA는 표준 명세일뿐 RTI의 구현하는데 있어서는 어떠한 제약사항도 없다. 하지만, HRTI 프레임워크는 SISO(Simulation Interoperability Standards Organization)의 표준인 DLC(Dynamic Link Compatibility) HLA API 1516을 기반으로 구현되었다 [7]. 같은 컴파일러만 사용한다면 사용자는 재컴파일 및 재링크 없이 다양한 RTI에 적용한 어플리케이션을 HRTI 프레임워크에 적용할 수 있다. 또한 HLA 명세의 변경이 없기에 기존에 개발했던 어플리케이션을 바로 적용할 수 있다.

4. 예제 : 단일 서버 큐잉 모델

단일 큐잉 서버 모델은 컴퓨터 혹은 통신 시스템에서 사용되는 일반적인 모델이다. 이 모델의 시뮬레이션은 앞서 제안한 FOM

Structure 형식론에 대한 효율성을 입증한다.

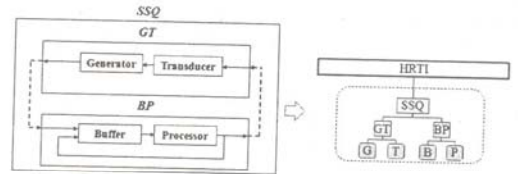


그림 3 전체 SSQ 페더레이션 구성

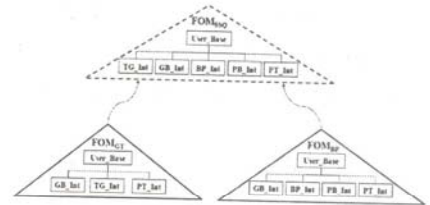


그림 4 SSQ 페더레이션의 FOM 정보

단일 서버 큐잉 (Single Server Queuing) 페더레이션은 Generator-Transducer 페더레이션과 Buffer-Processor 페더레이션으로 구성된다. 이들은 그림 4와 같이 각각 자신의 외부 FOM을 가지고 계층적 페더레이션을 구성하며 전체 페더레이션이 실행된다. 그림 3은 다음과 같이 FOM Structure 형식론으로 표현된다.

$$\begin{aligned}
 H &= \langle F, eFOM, iFOM, F-FOM, C, T \rangle \\
 F &= \{GT, BP, SSQ\} \\
 eFOM &= \{FOM_{GT}, FOM_{BP}\} \\
 iFOM &= \{iFOM_{SSQ}\} \\
 F-FOM &= \{(GT, FOM_{GT}, \emptyset), (BP, FOM_{BP}, \emptyset), (SSQ, \emptyset, iFOM_{SSQ})\} \\
 C &= \{(GT, SSQ), (BP, SSQ)\} \\
 T &: \\
 &T(FOM_{GT}, iFOM_{SSQ}) = iFOM_{SSQ} \\
 &T(FOM_{BP}, iFOM_{SSQ}) = iFOM_{SSQ}
 \end{aligned}$$

HRTI 프레임워크는 사용자에게 계층적 페더레이션을 구성하는데 있어서 풍부한 편의를 제공한다. 기존에 개발된 계층적 페더레이션을 재사용할 수 있도록 HRTI 프레임워크는 내부적으로 이들을 결합이 가능하도록 만든다. 따라서 사용자는 그림5 같이 자유자재로 계층적인 페더레이션을 구성할 수 있다.

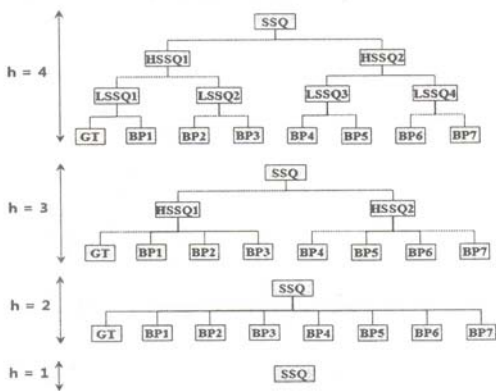


그림 5 다양한 계층에 따른 계층적 페더레이션 재사용

5. 결론 및 추후 과제

본 논문은 계층적인 페더레이션을 명세할 수 있는 FOM Structure 형식론을 제안하였다. 이를 지원하는 틀인 HRTI 프레임워크를 구현하였다. 구현된 HRTI 프레임워크는 SISO 표준인 DLC 기반으로 구현되어 있기에 기존의 개발된 페더레이션에 하위 호환성(backward compatibility)을 제공하여 페더레이션 레벨에서의 상호 운용성과 재사용성을 제공하는 일반적인 틀을 제공한다. SSQ 페더레이션 예제를 이용하여 명세적으로 계층적인 페더레이션을 표현할 수 있고 실용적으로 사용할 수 있음을 보였다.

또한 HRTI 프레임워크는 아직 개발중인 프로토타입이며 추후 과제로 계층적 페더레이션을 실제 물리적인 리소스에 할당하는 연구가 필요하다. 또한, 계층적 구조가 실제 시뮬레이션의 성능

에 어떤 영향을 미치는지에 관한 연구가 필요하다.

참고문헌

- [1] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)," IEEE Std 1516-2000, 2000.
- [2] Defense Modeling and Simulation Office, "High Level Architecture RTI Interface Specification, Version 1.3," April 1998.
- [3] A. Cramp, J. Best, and M. Oudshoorn, "Time Management in Hierarchical Federation Communities," Proceedings of 2002 the Fall Simulation Interoperability Workshop, 02F-SIW-031.
- [4] Aoyama, K., Ninomiya, S., Takeuchi, Y., Miyajima, S. and Tsutai, A, "Hierarchical Multi-Federation Structure of the Sensor Data Fusion Simulation in JUSA," Proceedings of the 1998 Fall Simulation Interoperability Workshop, 98F-SIW-045.
- [5] Jae-Hyun Kim and Tag Gon Kim, "Proposal of High Level Architecture Extension," Lecture Notes in Computer Science, Vol. LNAI 3397, pp. 128 - 137, 2005.
- [6] Jae-Hyun Kim and Tag Gon Kim, "Hierarchical HLA: Mapping Hierarchical Model Structure into Hierarchical Federation," M&S-MTSA'06, Ottawa, Canada, pp. 75 - 80, July, 2006.
- [7] Granowetter, L., "The Dynamic-Link-Compatible C++ RTI API for IEEE 1516," Proceedings of the 2004 Fall Simulation Interoperability Workshop, 04F-SIW-086.