

System Entity Structure 형식론을 이용한 분산 시스템 개발 프로세스

*김병수, 김탁곤

KAIST 전기 및 전자공학과

e-mail : *bskim@smslab.kaist.ac.kr*, *tkim@ee.kaist.ac.kr*

Development Process of Distributed Systems using System Entity Structure Formalism

*Byeong-Soo Kim, Tag-Gon Kim

Department of Electrical Engineering
KAIST

Abstract

Importance of distributed systems has grown in the multidisciplinary field. IEEE 1516.3 is one of the processes to develop a distributed system. However, when we develop distributed systems several times by using IEEE 1516.3, it spends a lot of time and cost. This paper proposes a development process of the distributed systems using System Entity Structure(SES) formalism to overcome the problem. SES formalism is a modeling method which represents structures and alternatives of the systems. By using SES formalism, it is possible to manage a structure of a distributed system effectively, and it reduces the development time and cost.

I. 서론

분산 시스템이란 네트워크를 통해 분산되어 있는 컴퓨터와 주변장치 등을 하나로 연결시켜 놓은 시스템이다. 분산 시스템은 국방, IT 등 다양한 분야에 걸쳐 이용되고 있으며, 이러한 분산 시스템을 위한 규약의 하나로 IEEE 1516 표준으로 지정된 HLA(High Level

Architecture)가 존재하고 이를 지원하는 도구로 RTI(Run Time Infrastructure)가 있다[1]. 많은 분산 시스템들이 HLA를 기반으로 개발되었으며, 이러한 페더레이트들은 IEEE 1516.3에 정의된 페더레이션 개발 및 실행 표준 프로세스(FEDEP, Federation Development and Execution Process)를 따라 개발된다[2]. 예를 들어 Live, Virtual, Constructive 시뮬레이터들로 구성된 LVC 연동체계와 같은 경우에는 연동 목적에 따라서 다양하게 설계 및 개발될 수 있다.

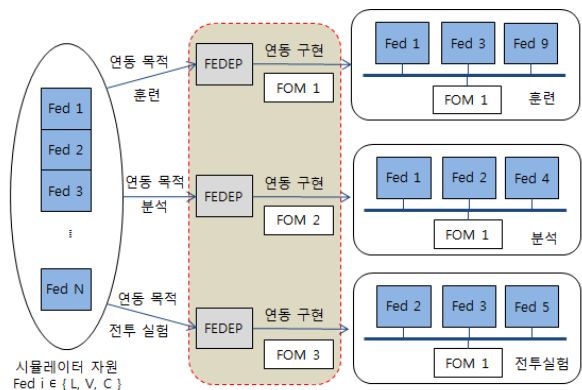


그림 1 목적에 따른 LVC 연동 체계 개발 과정

그림 1과 같이 연동 목적이 훈련, 분석, 전투 실험이냐에 따라서 각각 FEDEP 절차를 거쳐 분산 시스템이 개발되고 시뮬레이션을 하게 된다[3]. 이 때 여러 가지

연동 목적에 따라서 분산 시스템을 개발하거나, 동일한 목적에 대해서 시뮬레이터를 변경해가며 시뮬레이션을 할 경우가 발생한다. 그 때마다 FEDEP을 거쳐 반복적으로 분산 시스템을 개발하게 되고, FEDEP 절차 중에서 중복되는 과정이 발생하기 때문에 이에 따른 불필요한 시간과 비용이 소모된다.

본 논문에서는 효과적인 분산 시스템 개발을 위해서 SES(System Entity Structure) 형식론을 이용한 분산 시스템 개발 프로세스를 제안한다. SES 형식론은 시스템의 모든 대안들을 총체적으로 표현하는 모델링 방식인 다측면적인 시스템 모델링 방법의 하나로, 시스템이 어떻게 구성되어 있고 어떠한 대안들을 가질 수 있는지 표현하는 방법이다[4]. 본 논문에서는 FEDEP에 SES 형식론을 적용함으로써 효과적인 분산 시스템 개발을 통해 시간과 비용을 줄이고, 분산 시스템의 구조를 정형적으로 관리한다.

본 논문의 구성은 다음과 같다. 2장에서는 SES 형식론에 대해서 간단히 살펴보고, 3장에서는 SES 형식론을 이용한 분산 시스템 개발 프로세스와 이를 적용한 간단한 예제에 대해서 설명한다. 마지막으로 4장에서는 결론 및 향후 연구 방향에 대해서 알아본다.

II. SES 형식론

SES 형식론은 트리 구조를 이용하여 하나의 시스템이 가지는 대안들을 총체적으로 표현하는 형식론으로, 시스템의 구성 관계, 분류 관계, 연결 관계와 같은 세 가지 정보를 포함하고 있다[5]. 구성 관계는 시스템이 어떠한 모델들로 구성되어 있는지를 나타내고, 분류 관계는 시스템이 어떠한 모델을 대안으로 가질 수 있는지를 나타내며, 연결 관계는 모델들끼리의 연결 관계에 대해서 표현한다.

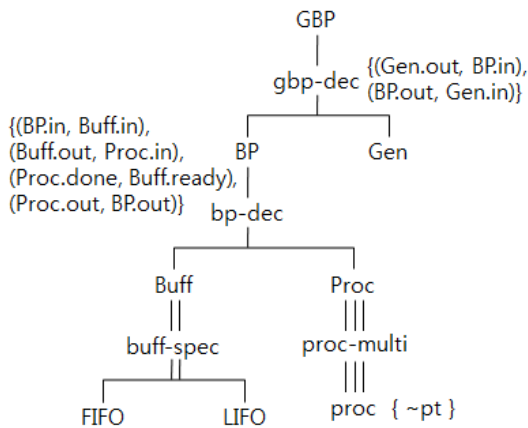


그림 2 SES로 표현한 단일 서버 큐잉 시스템

SES 형식론은 이러한 정보들을 표현하기 위한 몇 가지 종류의 노드들을 가진다. 먼저 entity 노드는 시스템을 구성하는 모델을 의미하며, aspect 노드는 하나의 모델이 어떠한 하위 모델들로 구성되어 있는지를 나타내며 트리에서 실선 한 줄로 그려진다. specialization 노드는 하나의 모델이 어떠한 대안들을 가질 수 있는지를 나타내는 노드로 트리에서 실선 두 줄로 그려진다. 예를 들어, 그림 2에서 entity 노드인 Buff는 buff-spec이라는 specialization 노드를 가지며 이 노드의 자식 노드인 FIFO와 LIFO 중 하나로 구체화된다. 이러한 노드 외에도 각 모델들 사이의 입출력 간의 연결을 명시하고(Coupling Scheme), 각 entity 노드가 가지는 attribute를 명시함으로써 SES 트리를 완성할 수 있다. 그림 2는 단일 서버 큐잉 시스템을 SES 형식론을 이용하여 표현한 예이다.

SES 형식론을 이용하면 여러 가지 대안에 대하여 전체적으로 모델링을 한 후에 각각을 시뮬레이션 함으로써 문제 해결 기법에 유용하다. 또한 방법론적인 시스템을 모델링이 가능하고, 모델의 구조변경이 쉬우며, 기존에 개발된 모델들을 재사용하기에 편리하다[6]. 본 논문에서는 분산 시스템 개발 프로세스에 SES 형식론을 적용함으로써 이와 같은 효과를 기대할 수 있다.

III. 분산 시스템 개발 프로세스

3.1 FEDEP

FEDEP은 IEEE 1516.3에 정의된 페더레이션 개발 및 실행 표준 프로세스로 그림 3과 같은 과정을 통해서 진행된다. 여기서 페더레이트는 HLA 규격을 만족하는 단독 시뮬레이터, 즉 연동에 참여하는 시뮬레이터를 의미하며 이러한 페더레이트들의 집합체를 페더레이션이라 한다. 또한 페더레이트들이 외부에 제공할 객체와 상호작용을 기술하는 SOM(Simulation Object Model)과, SOM을 통합하여 전체 페더레이션 내에서 공유되는 객체와 상호작용을 기술하는 FOM(Federation Object Model)이 정의된다.

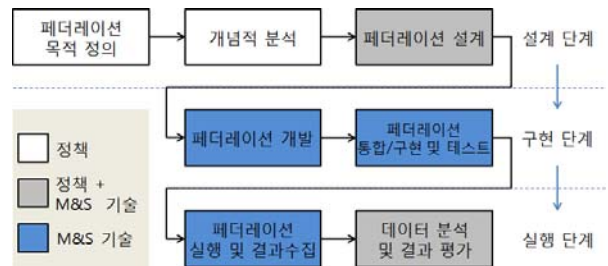


그림 3 FEDEP

FEDEP에서는 먼저 페더레이션 목적 정의 단계에서 사용자의 요구사항을 파악하고 시뮬레이션의 목적을 정의한다. 다음으로 개념적 분석 단계를 통해 최종 결과물에 대한 시나리오 개발 및 페더레이션 개념 모델이 개발되고, 이 결과를 토대로 페더레이션 설계 단계에서 기존에 개발된 페더레이트를 선택하거나 페더레이트 및 인터페이스를 설계한다. 페더레이션 개발 단계에서는 페더레이트 구현, FOM 및 페더레이트 기반 구조 구현이 이루어지며, 페더레이션 통합/구현 및 테스트 단계에서 전체 페더레이트들을 통합하고 테스트하게 된다. 마지막으로 페더레이션 실행 및 결과 수집 과정과 데이터 분석 및 결과 평가 과정을 통해 페더레이션 개발 프로세스가 완료된다.

3.2 SES를 이용한 페더레이션 개발 프로세스

이번 절에서는 SES 형식론을 이용한 새로운 분산 시스템 개발 프로세스를 제안한다. 제안하는 프로세스는 기본적으로 기존에 존재하는 FEDEP에 기반하며 재사용성을 고려한 상향식 개발 프로세스를 따른다[7]. 또한 entity 노드가 분산 시스템을 구성하는 각 페더레이트들을 의미한다고 가정하며, 각 페더레이트들은 attribute로 각각의 SOM을 가진다. 그림 4의 왼쪽은 기존의 FEDEP의 프로세스, 오른쪽은 새롭게 제안하는 프로세스이다. 본 논문에서는 직사화기에 대한 피해 평가를 연동 목적으로 하는 분산 시스템을 예를 들어 설명한다.

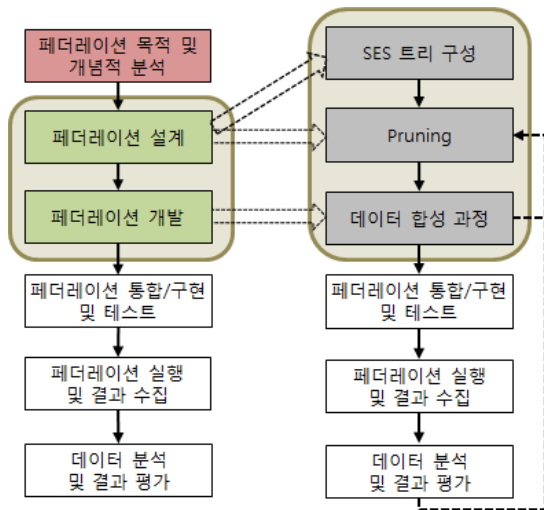


그림 4 새로운 페더레이션 개발 프로세스

가. 페더레이션 목적 및 개념적 분석

이 단계에서는 사용자의 요구사항을 파악하고 대상 페더레이션의 목적을 정의하며, 페더레이션 개념 모델 개발 및 페더레이션 요구사항 명세가 이루어진다. 직사화기 피해 모의에서의 목적은 여러 가지 직사화기가

차량, 병력, 건물 등의 교전 대상에 대해서 어느 정도의 피해를 입히는가에 대한 평가이다. 이 페더레이션은 피해를 입히는 무기(직사화기)에 대한 여러 가지 페더레이트와 피해를 입는 여러 가지 차량, 병력, 건물 페더레이트로 구성된다.

나. SES 트리 구성

페더레이션의 개념적 분석이 완료되면, 그 결과인 페더레이션 목적과 기개발된 페더레이트 목록을 바탕으로 SES 트리를 구성한다. 이 단계는 FEDEP의 페더레이션 설계와 대응되며, 페더레이트 선택 및 인터페이스 설계가 이루어진다. SES 트리는 페더레이션이 어떠한 페더레이트들로 구성되어 있으며, 어떠한 대안을 가지고, 어떠한 정보를 주고 받는지를 표현한다. 그림 5는 앞서 예를 든 피해모의에 대한 SES 트리 구조이다. 피해모의 노드는 dam-asp라는 aspect 노드를 가지고 그 노드는 무기, 차량, 병력, 건물의 네 가지 entity 노드를 가진다. 그리고 각 entity 노드들은 각각 specialization 노드를 가지고 그에 따라 무기는 K-1, K-11, M60을, 차량은 K1A1, M48AK을, 건물은 건물과 갯도를 각각 대안으로 갖는다. 병력 노드와 같이 대안이 존재하지 않는 경우에는 specialization 노드를 갖지 않는다. 또한 각각의 최하위 노드들은 attribute로 교환될 객체 정보인 SOM을 갖는다.

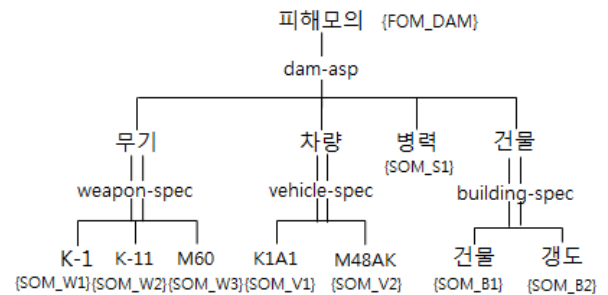


그림 5 SES 트리 구성

다. Pruning

SES 트리가 구성되면, 목적에 따라 사용자가 원하는 페더레이트만을 선택하는 Pruning 과정이 진행된다. Pruning을 통해 원하는 페더레이트 정보와 그 페더레이트의 attribute인 SOM 정보를 얻음으로써 페더레이션 설계가 완료된다. 그림 6은 여러 대안 중에서 무기는 K-1, 차량은 K1A1, 건물은 갯도를 선택하여 치환되는 Pruning 과정을 보여준다. 이 때 선택된 노드들은 자신의 부모 노드인 specialization 노드의 부모 노드와 치환되고 나머지 선택되지 못한 다른 entity 노드들과 specialization 노드는 트리에서 제거된다. 그 결과 그림 7과 같은 SES 트리를 얻을 수 있다.

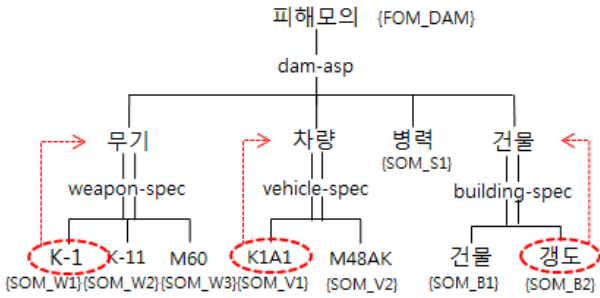


그림 6 Pruning 과정

라. 데이터 합성 과정

데이터 합성 과정은 페더레이션 통합 단계에 대응되는 단계로, 선택된 페더레이트 정보를 가지고 각각의 SOM으로 FOM을 구성하는 과정이다. 선택된 페더레이트의 SOM들의 합집합이 FOM으로 만들어진다. 이때 각 SOM들의 교집합이 공집합이면 페더레이트 간에 교환할 데이터가 없음을 의미하고, 연동의 목적을 상실한다. 이 경우 Pruning 단계로 돌아가 페더레이트들을 재선택한다. 그림 7에서 Pruning 이후 데이터 합성 과정을 통해 FOM인 FOM_DAM이 합성되고, FOM_DAM은 피해모의 노드의 attribute가 된다.

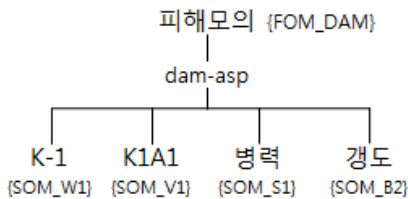


그림 7 Pruning 결과

마. 페더레이션 통합, 실행 및 결과 분석

이후의 과정은 선택된 페더레이트 정보과 FOM을 이용하여 페더레이션을 통합하고, 시뮬레이션하는 과정이다. 이 과정을 통해 시뮬레이션을 한 후 시뮬레이션 결과가 얻어지면 결과 분석 과정을 거친다. 이 때 피해모의 시뮬레이션의 결과 분석을 통해 원하는 결과가 얻어지면 프로세스를 마무리하고, 그렇지 않을 경우 다시 Pruning 과정으로 돌아가 프로세스를 다시 진행하여 새로운 페더레이션과 시뮬레이션 결과를 얻을 수 있다.

즉, 새로운 프로세스를 통해서 FEDEP의 반복적인 목적 설정 과정을 줄일 수 있고, 기존 페더레이트들의 재사용과 정형적인 페더레이션 구조 관리를 통해서 페더레이션 개발을 효율적으로 할 수 있다.

IV. 결론 및 향후 연구 방향

본 논문은 SES 형식론을 이용한 분산 시스템 개발 프로세스를 제안하였다. 본 논문에서 제안한 프로세스는 SES 형식론을 이용함으로써 분산 시스템 구조를 효과적으로 관리할 수 있고, 구조 변경 및 시뮬레이터의 재사용에 용이하다. 이를 통해 분산 시스템의 효율적인 개발이 가능하다. 향후 연구 방향으로는 사용자의 편의성을 높이기 위해 제안된 프로세스를 적용하여 주어진 시뮬레이터를 자동으로 합성하는 분산 시스템 합성 환경에 대해 연구하도록 한다.

감사의 글

본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다(UD110006MD).

참고문헌

- [1] IEEE, IEEE Standard for Modeling and Simulation(M&S) High Level Architecture(HLA) - Framework and Rules, IEEE Std 1516-2000, 2000.
- [2] IEEE, IEEE Standard for Modeling and Simulation(M&S) High Level Architecture(HLA) - Federation Development and Execution Process(FEDEP), IEEE Std 1516.3-2000, 2003.
- [3] 김탁곤, 강의노트: 국방 M&S 이론 및 기술 - 연동편, 전기 및 전자공학과, KAIST, 2011.
- [4] Bernard P. Zeigler, Multifaceted Modelling and Discrete Event Simulation, ACADEMIC PRESS, 1984.
- [5] T. G. KIM, C. LEE, E. R. CHRISTENSEN and B. P. ZEIGLER, "System Entity Structuring and Model Base Management," IEEE Transactions on Systems, Man and Cybernetics, vol. 20, no. 3, pp. 1013-1024, Oct., 1990.
- [6] Wan Bok Lee, "Development of The Multifaceted System Modelling/Simulation Environment," Master Thesis, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, 1994.
- [7] 안정현, 김탁곤, "IEEE 1516 HLA 표준에 기반한 분산 시스템 개발 프로세스," 대한전자공학회 하계 종합학술대회, pp. 1582-1585, 2010년 6월.