한국시뮬레이션학회 논문지
Vol. 24, No. 4, pp. 51-58 (2015. 12)
http://dx.doi.org/10.9709/JKSS.2015.24.4.051
ISSN 1225-5904

재사용 가능한 시뮬레이션 시나리오 프레임워크 개발

이용헌* · 이승영 · 김세환

Development of reusable simulation scenario framework

Yong Heon Lee* · Seung Young Lee · Sae Hwan Kim

ABSTRACT

Since SBA is regarded as mandatory for DM&S, many M&S frameworks and solutions were developed and applied to military simulation systems. Simulation systems developed based on such framework or solution are consist of simulation control part, scenario generation part and modeling part in common. Although the functions of these parts are similar between another simulation systems, these are developed separately by different implementation, so it causes increase of development cost. In order to solve these problems, we developed a simulation scenario framework which can be reusable regardless of simulation purpose. In this paper, we propose an architecture of simulation scenario framework and its functions.

Key words : M&S, Simulation, Scenario, Reusability, Framework, QUEST

요 약

국방 M&S 분야에서 모의 기반 획득 방법이 필수적인 정책으로 인식되면서 M&S 관련 프레임워크 및 솔루션들이 개발되어 현업에 적용되고 있다. 이러한 프레임워크나 솔루션을 기반으로 개발되는 시뮬레이션 시스템은 공통적으로 통제부, 시나리오 생성부, 모의부로 나눌 수 있고 각 파트가 수행하는 기능은 시뮬레이션 시스템에 관계 없이 유사하다. 그럼에도 불구하고 시뮬레이션 시스템마다 각기 다른 시뮬레이션 생성 기능을 구현함으로써 시스템 개발 비용이 증가하는 문제를 겪고 있다. 이러한 문제를 해결하기 위하여 목적 시뮬레이션 시스템의 모의 내용과 관계 없이 재사용이 가능한 시뮬레이션 시나리오 프레임워크를 개발하였으며 이를 통해 다양한 시뮬레이션 시스템을 효과적으로 개발할 수 있었다. 본 연구에서는 QUEST[1][2][3]를 기반으로 하여 모의 모델에 관계없이 재사용 가능한 시뮬레이션 시나리오 프레임워크의 구조와 기능 요소들에 대해 설명한다.

주요어 : M&S, 시뮬레이션, 시나리오, 재사용, 프레임워크, QUEST

1. 서 론

최근 모델링 및 시뮬레이션(M&S) 기술이 다양한 분야에서 활용되고 있으며, 특히 국방 분야에서는 모의기반획득(SBA; Simulation Based Acquisition) 이 필수적인 정책으로 인식되고 있다[4]. 모의기반획득은 무기체계 소요분석 및 결정에서 분석/설계, 제작, 시험/평가, 훈련/운용/군수지원에 이르는 전 과정 상에서 M&S을 활용함으로써 무기체계 획득의 시간, 자원, 위험을 실질적으로 감소시킬 수 있다. 이를 위해서 획득 단계 별 M&S적용을 위한 M&S 관련 솔루션, 프레임워크 등을 통해 다양한 M&S 시스템들이 개발되어 현업에 광범위하게 적용되어 있다. 이러한 시스템들은 일반적으로 모의 환경을 가지고 있으며 이러한 모의 환경은 공통적으로 모의 시스템을 통제하기 위한 통제 기능, 실제 모의를 수행하는 모의 기능, 모의를 위한 시나리오를 생성하고 이를 모의부에 전달하는 시뮬레이션 시나리오 생성 기능을 가지게 된다. 이러한 기능들은 M&S 시스템마다 각기 다른 구조와 형태로 구현되어 적용되고 있기 때문에 시뮬레이션 시스템간에 시나리오를 공유하거나 이기종 시스템을 통합 연동하는 것이 어려우며 시뮬레이션 시스템마다 시뮬레이션 시나리오의 구조

Received: 2 November 2015, **Revised:** 20 November 2015, **Accepted:** 27 November 2015

***Corresponding Author:** Yong Heon Lee
E-mail: yongheon.lee@lignex1.com
LIG Nex1 Co. LTD.

와 명세가 달라지기 때문에 이미 구현된 시뮬레이션 시스템의 시나리오 생성 기능을 재사용하기도 어려운 문제가 있다.

시뮬레이션 시스템마다 시뮬레이션 시나리오의 구조와 명세가 다르지만 각 시스템에서 사용하는 시뮬레이션 시나리오 관련 기능은 매우 유사하다. 일반적으로 시뮬레이션 시스템은 시뮬레이션 시나리오와 관련하여 다음과 같은 기능을 제공한다.

- 시뮬레이션 시나리오 작성(생성)
- 시뮬레이션 시나리오 수정/삭제
- 시뮬레이션 시나리오 배포
- 시뮬레이션 시나리오 정보를 모델에 설정

이러한 기능들은 개발되는 시뮬레이션 시스템마다 다른 방식으로 구현이 되고 시뮬레이션 시나리오의 구조와 구현이 종속되기 때문에 구현된 기능을 새로운 시뮬레이션 시스템을 개발하기 위해 재사용하기가 어렵고 시스템을 개발할 때마다 매번 시뮬레이션 시나리오 관련 기능 개발을 위해 높은 비용을 지불해야 하는 문제가 발생하게 된다. 이는 새로운 시뮬레이션 시스템을 개발할 때 뿐만이 아니라 기개발된 시뮬레이션 시스템을 유지보수하는데 있어서도 큰 문제로 작용한다.

이러한 문제를 해결하기 위해서는 시뮬레이션 시스템의 시나리오 구조의 변경에 유연하고 재사용이 가능한 시뮬레이션 시나리오 체계와 이를 통해 시뮬레이션 시스템을 구현할 수 있는 시뮬레이션 시나리오 프레임워크가 필요하다. 본 연구에서는 시뮬레이션 시스템마다 공통적으로 재사용할 수 있는 시뮬레이션 시나리오 프레임워크의 구조와 설계를 제안한다. 2장에서는 관련 연구에 대해 소개하며 3장에서는 시뮬레이션 시나리오 프레임워크의 구성 요소에 대해서 설명한다. 4장에서는 시뮬레이션 시나리오 프레임워크의 구조와 구체적인 설계에 대해서 설명하고 5장에서 결론을 맺는다.

2. 관련 연구

QUEST는 DEVS[5] 기반의 혼합시간 시뮬레이션을 위한 모델 작성, 실험 틀, 가시화, 분석 및 Monte-Carlo 시뮬레이션 도구를 Fig. 1과 같이 하나의 GUI에서 제공하는 시뮬레이션 도구이다. QUEST에서는 GUI기반으로 이산사건 및 연속시간 모델을 작성하고 시뮬레이션 할 수 있으며 2차원/3차원 전시 및 실시간/사후 분석 기능을 탑

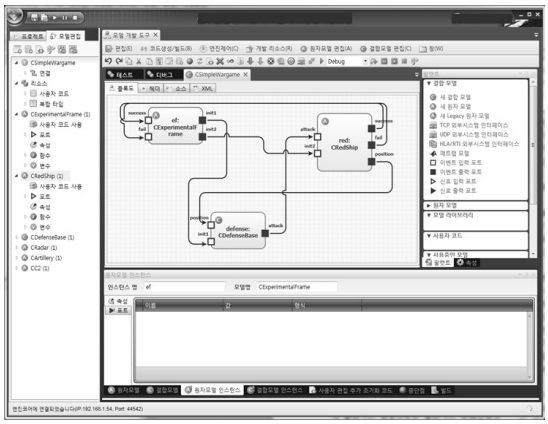

Fig. 1. QUEST's main window

재하여 하나의 도구에서 SBA 전순기에 필요한 필수적인 기능 요소를 제공하고 있다. QUEST에서 제공하는 여러 기능을 통하여 시뮬레이션 소프트웨어와 TCP/IP, UDP, HLA 등의 외부체계 통신 프로토콜을 사용하는 소프트웨어를 개발하는 것이 매우 용이하다. 이러한 QUEST에는 모델의 초기 속성값의 집합을 시나리오라고 명명하고 있으며 이러한 시나리오를 설정할 수 있는 기능을 제공하고 있지만 시나리오 설정을 외부 도구에서 수행할 수 없고 분산 시뮬레이션 환경에서 분산된 여러 모의기의 시나리오를 한 곳에서 관리할 수 없는 문제가 존재한다. 뿐만 아니라 모의 모델의 인스턴스의 개수를 모델 개발 시점에 정해야 하는 방식을 취하고 있어서 모델 인스턴스의 개수를 실행시간 중에 변경하는 시나리오를 작성하지 못하는 한계가 있다.

이러한 문제를 해결하기 위하여 분산 시뮬레이션 환경에서 QUEST 기반으로 개발된 여러 모의기의 시나리오를 관리할 수 있는 환경과 프레임워크의 필요성이 대두되었으며 본 논문에서 소개하는 시뮬레이션 시나리오 프레임워크의 개발로 이어지게 되었다.

3. 시뮬레이션 시나리오 프레임워크 구성

시뮬레이션 시나리오 프레임워크는 Fig. 2와 같이 시나리오 구조 편집기(Scenario structure editor), 시나리오 생성기(Scenario generator), 시나리오 구조 해석기(Scenario structure parser), 시나리오 명세 해석기(Scenario description parser), 모델 연동기(Model interface unit)로 구성된다. 시나리오 구조 편집기와 시나리오 생성기는 시뮬레이션

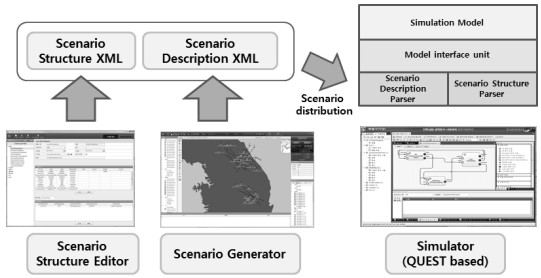

Fig. 2. Simulation scenario framework overview

시스템의 시나리오 생성부 또는 운용 통제부에 위치하게 되고 시나리오 구조 해석기, 시나리오 명세 해석기 및 모델 연동기는 시스템의 모의부에 위치하게 된다. 본 연구에서는 시나리오 생성부와 모의부가 HLA/RTI로 연동하는 구조를 취하고 있으며 시나리오 구조와 시나리오 명세는 각각 다른 스키마를 가진 XML 파일 형태를 취한다.

3.1 시나리오 구조 편집기

시나리오 구조 편집기는 GUI(Graphical User Interface) 기반으로 시뮬레이션 시나리오의 구조를 편집하는 기능을 제공한다. 시뮬레이션 시스템마다 모의기의 형상과 모의기의 모델이 다르기 때문에 모델의 시나리오 구조도 시뮬레이션 시스템마다 달라진다. 하나의 시나리오 구조만으로는 여러 시뮬레이션 시스템에 대응하여 재사용이 불가능하므로 본 연구에서는 이러한 시뮬레이션 시나리오의 구조 자체를 편집할 수 있는 기능을 제안하고 있다. 시나리오 구조 편집기에서는 QUEST에서 사용하는 기본/열거/복합 자료형에 대한 편집이 가능하며 QUEST에서 작성하는 모델의 속성 목록을 작성할 수 있다.

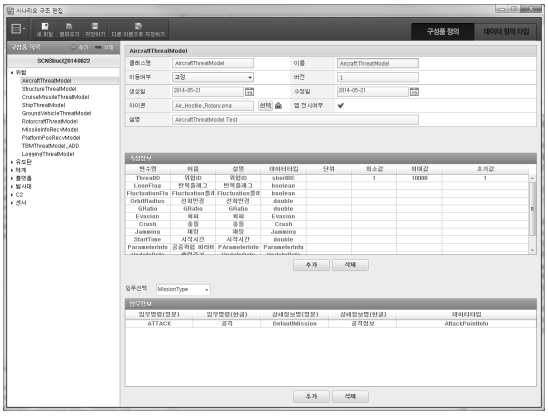

Fig. 3. Simulation scenario structure editor

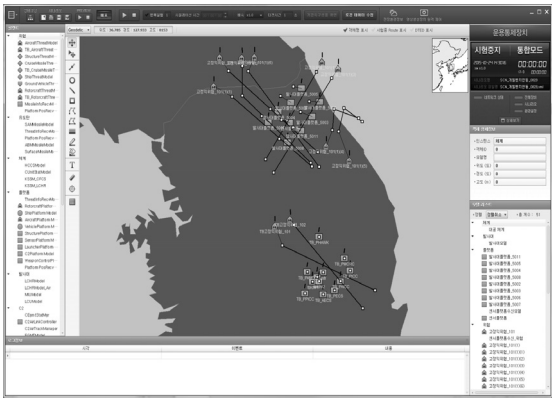

Fig. 4. Simulation scenario generator

3.2 시나리오 생성기

시나리오 생성기는 GUI 기반으로 2차원 지도와 MIL-STD-2525C 형식의 표준 부호체계를 따르는 아이콘을 사용하여 시나리오 구조에 대응하는 시나리오 명세를 만드는 기능을 제공한다. 시나리오 생성기에서는 시나리오 구조 편집기에서 정의한 모의 모델의 속성 목록의 값을 지정할 수 있으며 객체의 이동을 모의하는 모델의 경우 경로점에 대한 정보를 편집할 수 있다. 이 때 각 경로점마다 시나리오 구조 편집기에서 정의한 임무 값을 설정할 수 있다. 시나리오 생성기에서 편집한 시나리오 명세 정보는 XML 형태로 저장되어 HLA/RTI 네트워크 통해 모의부로 전달된다.

3.3 시나리오 구조 해석기

시나리오 구조 해석기는 시나리오 구조 편집기를 통해 작성한 시나리오 구조 XML의 정보를 해석하고 이를 모델 연동기에 전달하는 역할을 담당한다. 시나리오 구조 해석기는 XML 스키마를 기반으로 동작한다.

3.4 시나리오 명세 해석기

시나리오 명세 해석기는 시나리오 생성기를 통해 작성한 시나리오 명세 XML의 정보를 해석하고 이를 모델 연동기에 전달하는 역할을 담당한다. 시나리오 명세 해석기도 시나리오 구조 해석기와 마찬가지로 XML 스키마를 기반으로 동작한다.

3.5 모델 연동기

모델 연동기는 시스템의 시나리오 생성부로부터 전달받은 시나리오 구조와 시나리오 명세 정보를 실제 모델에

전달하는 역할을 담당한다. QUEST에서 작성한 모델은 모델마다 속성(Property)을 가질 수 있고 속성의 초기 값을 설정할 수 있는 기능을 QUEST에서 제공하고 있는데 모델 연동기에서는 시뮬레이션 시나리오 명세에 기술된 정보를 모델 속성의 초기 값으로 설정하는 기능을 수행한다. 이러한 모델 연동기를 통하여 QUEST 도구에서 뿐만이 아니라 시나리오 생성부를 통해서도 모델의 초기 속성 값을 설정할 수 있게 된다.

4. 시뮬레이션 시나리오 프레임워크 설계

4.1 시나리오 구조 XML 스키마 설계

시나리오 구조 XML은 시뮬레이션 시나리오에서 사용하는 자료형과 QUEST 기반으로 작성되는 모델의 속성에 대한 자료형의 구조와 목록을 정의하며 시나리오 구조 XML 스키마에서는 이러한 시나리오 구조 표현을 위한 규칙을 정의하고 있다.

본 연구에서 제안하는 시뮬레이션 시나리오 명세는 최종적으로 QUEST에서 작성한 모델의 초기 속성값으로 전달되기 때문에 시나리오 명세에서 정의하는 값들의 자료 구조와 QUEST기반 모델에서 정의하는 속성의 자료 구조가 동일해야 할 필요가 있다. QUEST에서는 자사에서 개발한 플러그인 프레임워크[6]와 XML 기반 객체 모델[7-10]에서 정의할 수 있는 기본/열거/복합 자료형의 자료 구조를 그대로 사용하도록 설계되었으므로 시나리오 구조도 이러한 기본/열거/복합 자료형의 구조를 정의할 수 있는 스키마가 필요하다.

시나리오 구조 XML에서는 다음과 같은 기본 자료형을 정의할 수 있고 시나리오 구조 XML 스키마에 해당 자료형들의 기술 방법이 정의된다.

- Boolean
- Byte
- Character (multibyte, unicode)
- Short integer
- Unsigned Short Integer
- Integer (2~8 byte)
- Unsigned Integer (2~8 byte)
- Float
- Double
- String (multibyte, unicode)

이러한 기본 자료형들은 C, C++, Java, C# 등의 프로그래밍 언어에서 사용하는 일반적인 기본 자료형들에 대응될 수 있으며 이러한 기본 자료형들을 사용하여 작성하는 시나리오의 데이터 구조를 XML 형식으로 표현할 수 있게 된다. 일반적으로 프로그래밍 언어에서는 이러한 기본 자료형 외에도 특정 이름을 가지는 정수형의 상수의 집합을 정의할 수 있는 열거 자료형(enumeration type)과 기본 자료형, 열거 자료형을 묶어서 하나의 사용자 정의 자료 구조를 만들 수 있는 복합 자료형(complex type; struct)을 정의할 수 있는데 시나리오 구조 XML 스키마에서도 이를 정의할 수 있는 규칙을 가지고 있다.

열거 자료형은 자료형의 이름과 열거 자료형에서 정의하는 상수 목록을 가지는데 이 상수들은 모두 상수 고유의 이름과 정수 값을 가진다. 복합 자료형은 앞서 정의한 기본 자료형과 열거 자료형을 사용하여 만들어지며 이미 정의한 복합 자료형을 사용하여 새로운 복합 자료형의 구조를 계층적(hierarchical)으로 만들 수도 있다. 복합 자료형은 자료형의 이름과 복합 자료형에서 사용하는 하위(child) 자료형들의 목록을 가지는데 이 하위 자료형들은 필드(field)라고 칭한다. 각 필드는 필드의 이름과 데이터 자료형 이름을 가진다. 이 필드의 자료형 이름은 기본/열거/복합 자료형 정의에서 기술하는 자료형 이름이며 이를 통해 복합 자료형에서 정의하는 각 필드의 데이터 구조를 알아낼 수 있게 된다. 이러한 시나리오 구조 XML 스키마의 데이터 구조 정보는 Fig. 5와 같이 계층적으로 표현할 수 있다.

시나리오 구조에서 사용하는 자료형을 정의한 후에는 QUEST에서 작성하는 모델(이하 모델)의 구조를 정의한다. 모델에 대한 XML 스키마는 다음과 같은 요소를 가진다.

- 모델의 고유 이름(영문)
- 모델의 GUI 표시 이름(영문/한글)

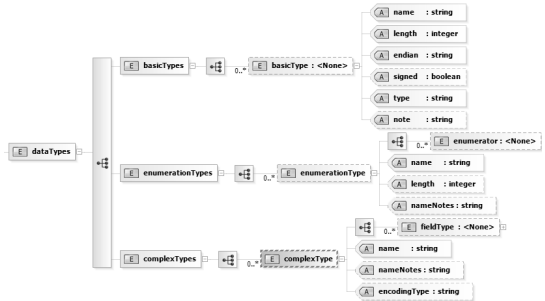

Fig. 5. Scenario structure XML schema (data types)

- 모델에 대한 설명
- 생성일자/수정일자
- 속성(Property; Attribute)
- 임무(Mission)

모델의 고유 이름은 QUEST에서 작성하는 원자 모델의 이름을 지칭한다. 모델의 속성은 모델이 가질 수 있는 고유 속성값을 지칭하며 모델마다 1개 이상을 정의할 수 있다. 시나리오 구조에서 정의하는 모델의 고유 이름과 QUEST에서 작성하는 모델의 이름이 같고 속성의 데이터 구조가 동일한 경우 해당 시나리오 구조를 기반으로 작성하는 시뮬레이션 시나리오의 값을 자동으로 모델의 속성 값으로 입력할 수 있다. 이에 대해서는 4.4 장에서 설명하는 모델 연동기를 통해 수행된다.

모델의 정보를 2차원 또는 3차원으로 전시하거나 이동성을 모의하는 모델의 경우 모델 객체의 위치를 지정하기 위한 경로점의 집합을 시나리오 상에 설정할 수 있어야 한다. 모델의 임무는 이러한 경로점마다 모델이 수행해야 하는 임무의 정보를 정의하기 위한 요소이다. 이러한 임무 역시 하나의 모델에 여러 개를 정의할 수 있으며 시나리오 명세 작성시 경로점에 해당 모델의 임무 정보를 선택하여 적용할 수 있다. 이러한 시나리오 구조 XML 스키마의 모델 속성 정보는 Fig. 6과 같이 계층적으로 표현할 수 있다.

4.2 시나리오 명세 XML 스키마 설계

시나리오 명세 XML은 시뮬레이션 시나리오 구조를 기반으로 실제 시뮬레이션 시나리오의 명세를 기술하며 시나리오 명세 XML 스키마에서는 이러한 시나리오 명세 표현을 위한 규칙을 정의하고 있다.

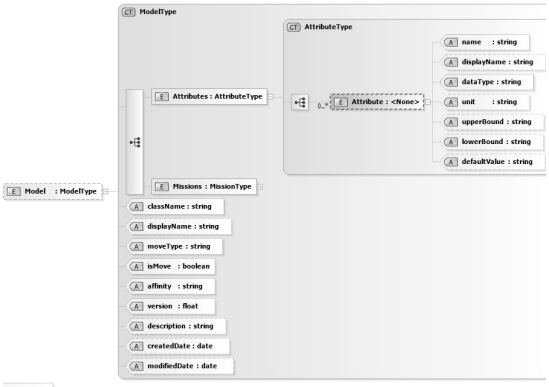

Fig. 6. Scenario structure XML schema (models)

시나리오 명세 XML의 스키마는 모델 인스턴스의 세부 속성 값을 설정하기 위한 규칙과 경로점의 정보를 정의하기 위한 규칙을 정의하고 있다. 모델 인스턴스는 다음과 같은 요소를 가진다.

- 모델 인스턴스의 식별자(ID)
- 모델 인스턴스의 고유 이름(영문)
- 모델 인스턴스의 GUI 표시 이름(영문/한글)
- 모델 인스턴스를 생성할 모델의 고유 이름
- 모델에 할당된 경로점의 식별자
- 모델 인스턴스의 세부 속성 값 목록

시나리오 구조에서 정의하는 모델의 구조 정보는 모델마다 한 번씩 정의하게 되지만 이 모델의 구조를 기반으로 실행 시간에 생성되는 모델의 인스턴스는 다수일 수 있다. 따라서 같은 고유 이름을 사용하는 모델일지라도 인스턴스를 구분하기 위하여 식별자가 필요하며, 이러한 식별자는 정수와 문자열의 형태로 규정한다.

모델 인스턴스의 세부 속성 값은 모델 속성의 'attribute path'와 속성 값으로 규정한다. 모델의 속성은 기본/열거/복합 자료형으로 정의할 수 있는데 기본 자료형과 열거 자료형만 실제 값을 설정할 수 있으므로 복합자료형으로 설정된 속성의 경우 해당 복합 자료형 내부에 존재하는 모든 기본/열거 자료형에 대한 속성 값을 설정해야 한다. 이러한 시나리오 명세 XML 스키마는 Fig. 7과 같이 계층적으로 표현할 수 있다.

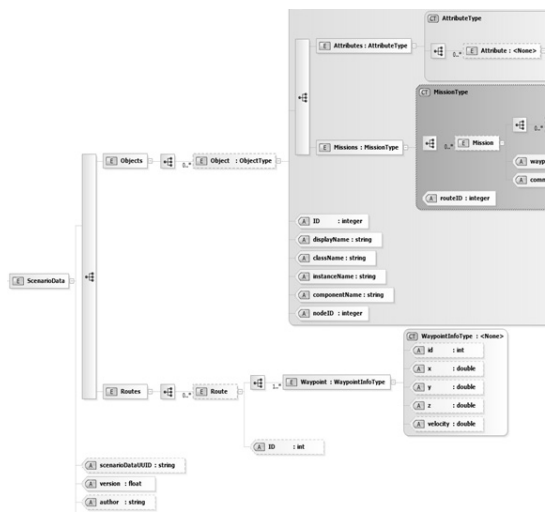

Fig. 7. Scenario description XML schema

4.3 시나리오 구조 및 명세 해석기 설계

시나리오 구조와 명세 XML은 시뮬레이션 시스템의 시나리오 생성부에서 작성하여 HLA/RTI와 같은 외부 통신 프로토콜을 통해 각 모의기로 배포된다. 이 때 시나리오 구조 XML과 명세 XML이 동시에 배포되며 각 모의기에서는 시나리오 구조 해석기와 시나리오 명세 해석기를 사용하여 각 XML 파일을 해석(Parsing)한다. 이 때 시나리오 명세에는 시나리오 구조에서 정의한 모델의 속성에 대한 실제 값이 기록되므로 시나리오 구조와 시나리오 명세의 항목이 대응되는지 검사해볼 필요가 있다. 이를 위하여 시나리오 구조 해석이 먼저 이루어지고 해석된 정보가 시나리오 명세 해석기로 전달되어 시나리오 명세를 해석하면서 시나리오 구조와의 대응 여부를 판단하게 된다. Fig. 8은 이러한 시나리오 구조 해석기와 시나리오 명세 해석기에 대한 Class Diagram이다.

4.4 모델 연동기 설계

모델 연동기는 시나리오 구조 해석기와 시나리오 명세 해석기를 통해 해석된 시뮬레이션 시나리오 정보를 모의기의 모델 속성 값으로 설정하는 역할을 수행한다. 모델 연동기의 세부 기능은 다음과 같다. 각 기능은 시나리오 정보를 전달 받은 이후에 순차적으로 수행된다.

- 기 생성된 모델 인스턴스 삭제
- 모델 인스턴스 생성
- 모델 인스턴스 속성 값 설정
- 모델 인스턴스 임무 값 설정

앞서 언급한 바와 같이 QUEST에서 작성하여 도구 자체 내에서 실행하는 모델의 경우 인스턴스의 개수가 정해져 있으나 시나리오 생성기에서 작성하는 시뮬레이션 시나리오에는 이러한 모델 인스턴스의 개수가 정해져 있지 않다. 이러한 부분을 해결하기 위하여 모델 연동기에서는 이미 생성된 모델의 인스턴스를 삭제하고 해석된 시뮬레이션 시나리오 명세를 기반으로 인스턴스를 재생성하는 루틴을 수행하게 된다. 모델 인스턴스 재생성 후에는 각 모델 인스턴스의 속성 값과 임무 값을 설정한다. 이러한 절차는 시나리오 생성기로부터 새로운 시나리오 명세를 배포받을 때마다 수행된다. Fig. 9는 이러한 모델 연동기에 대한 Class Diagram이다.

이러한 시뮬레이션 시나리오 프레임워크 기반으로 시뮬레이션 시나리오를 작성하고 모의기로 전달하는 과정을 다음과 같이 정리할 수 있다.

- QUEST를 사용한 모의기 개발
- 모의 모델의 속성 정보를 기반으로 하는 시뮬레이션 시나리오 구조 작성 (시나리오 구조 편집기)
- 시나리오 구조를 기반으로 시뮬레이션 시나리오 명세 작성 (시나리오 생성기)
- 작성된 시나리오 구조 XML과 시나리오 명세 XML을 모의기로 배포 (HLA/RTI)
- 모의기에서는 배포 받은 시나리오 구조 XML과 시나리오 명세 XML을 해석(Parsing) (시나리오 구조 해석기 및 시나리오 명세 해석기)
- 해석된 시뮬레이션 시나리오의 정보를 모의 모델 인스턴스의 속성 값으로 설정 (모델 연동기)

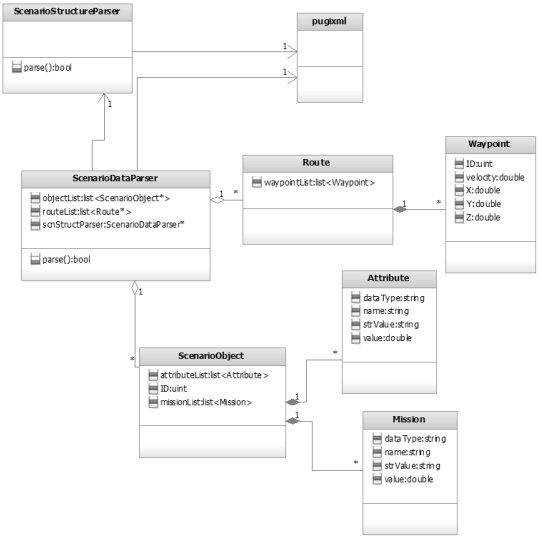

Fig. 8. Class diagram of scenario structure parser and scenario description parser

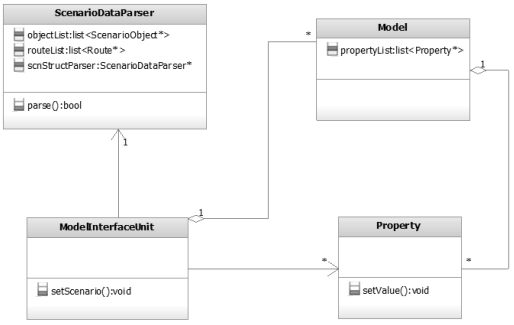

Fig. 9. Class diagram of model interface unit

4.5 시뮬레이션 시나리오 프레임워크 개발 결과

시뮬레이션 시나리오 프레임워크의 구성 요소 중에서 시나리오 구조 편집기와 시나리오 생성기는 3장의 Fig. 3과 Fig. 4와 같이 GUI 형태로 구현되었다. 시나리오 구조와 시나리오 명세는 XML 스키마 기반으로 기술되므로 각각은 XML 형태의 파일로 작성되어 관리된다. Fig. 10과 Fig. 11은 시나리오 구조 편집기와 시나리오 생성기에서 작성한 시나리오 구조 및 시나리오 명세 XML 파일에 대한 예시이다.

시나리오 구조와 시나리오 명세는 XML 형태로 기술되기 때문에 매우 직관적인 구조를 취하고 있으며 시나리오 구조 편집기와 시나리오 생성기가 아닌 다른 도구를 사용하여 이를 쉽게 편집할 수 있는 장점이 있으며 XML 스키마를 기반으로 하고 있기 때문에 XML 문서에 대한 무결성 검증에 매우 용이하다.

```
<SCNStruct:ScenarioStructure xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SCNStruct
  <SCNStruct:Components>
    <SCNStruct:Component name="ThreatVariable" displayName="위협" description="2014-05-21" created
      <SCNStruct:Model InstanceName="" className="AircraftThreatModel" PTDisplayName="고정익위협기
        <SCNStruct:Attributes>
          <SCNStruct:Attribute name="ThreatID" displayName="위협ID" nameNotes="위협ID" dataType="s
          <SCNStruct:Attribute name="LoopFlag" displayName="반복플래그" nameNotes="반복플래그" dat
          <SCNStruct:Attribute name="FluctuationFlag" displayName="Fluctuation플래그" nameNotes="F
          <SCNStruct:Attribute name="OrbitRadius" displayName="선회반경" nameNotes="선회반경" data
          <SCNStruct:Attribute name="GRatio" displayName="GRatio" nameNotes="GRatio" dataType="dou
          <SCNStruct:Attribute name="Evasion" displayName="회피" nameNotes="회피" dataType="Evasio
          <SCNStruct:Attribute name="Crash" displayName="충돌" nameNotes="충돌" dataType="Crash" u
          <SCNStruct:Attribute name="Jamming" displayName="재밍" nameNotes="재밍" dataType="Jammin
          <SCNStruct:Attribute name="StartTime" displayName="시작시간" nameNotes="시작시간" dataTy
          <SCNStruct:Attribute name="ParameterInfo" displayName="공중위협 파라미터정보" nameNotes=
          <SCNStruct:Attribute name="UpdateRate" displayName="출력주기" nameNotes="UpdateRate" dat
          <SCNStruct:Attribute name="Attack" displayName="공격제원정보" nameNotes="공격제원정보" d
        </SCNStruct:Attributes>
        <SCNStruct:Missions>
          <SCNStruct:Command enumType="MissionType" />
          <SCNStruct:MissionInfo name="DefaultMission" displayName="공격경로" dataType="AttackPoin
        </SCNStruct:Missions>
      </SCNStruct:Model>
      <SCNStruct:Model InstanceName="" className="StructureThreatModel" PTDisplayName="지상구조물위
        <SCNStruct:Attributes>
          <SCNStruct:Attribute name="ThreatID" displayName="위협ID" nameNotes="위협ID" dataType="s
          <SCNStruct:Attribute name="LoopFlag" displayName="반복플래그" nameNotes="반복플래그" dat
          <SCNStruct:Attribute name="FluctuationFlag" displayName="Fluctuation플래그" nameNotes="F
          <SCNStruct:Attribute name="OrbitRadius" displayName="선회반경" nameNotes="선회반경" data
          <SCNStruct:Attribute name="GRatio" displayName="GRatio" nameNotes="GRatio" dataType="dou
          <SCNStruct:Attribute name="Evasion" displayName="회피" nameNotes="회피" dataType="Evasio
          <SCNStruct:Attribute name="Crash" displayName="충돌" nameNotes="충돌" dataType="Crash" u
```

Fig. 10. Example of scenario structure XML

```
<scn:Object ID="1014" displayName="고정익위협_6103" className="AircraftThreatMo
  <scn:Attributes>
    <scn:Attribute attributePath="ThreatID" value="6103" />
    <scn:Attribute attributePath="LoopFlag" value="0" />
    <scn:Attribute attributePath="FluctuationFlag" value="0" />
    <scn:Attribute attributePath="OrbitRadius" value="0" />
    <scn:Attribute attributePath="GRatio" value="0" />
    <scn:Attribute attributePath="Evasion.EvasionFlag" value="0" />
    <scn:Attribute attributePath="Evasion.EvasionType" value="0" />
    <scn:Attribute attributePath="Evasion.Pattern" value="0" />
    <scn:Attribute attributePath="Evasion.StartRange" value="0" />
    <scn:Attribute attributePath="Evasion.EndRange" value="0" />
    <scn:Attribute attributePath="Evasion.GDuration" value="0" />
    <scn:Attribute attributePath="Evasion.turnG" value="0" />
    <scn:Attribute attributePath="Crash.CrashFlag" value="0" />
    <scn:Attribute attributePath="Crash.CrashType" value="0" />
    <scn:Attribute attributePath="Crash.Pattern" value="0" />
    <scn:Attribute attributePath="Crash.FaultDuration" value="0" />
    <scn:Attribute attributePath="Crash.DetonationDuration" value="0" />
    <scn:Attribute attributePath="Crash.FaultDrag" value="0" />
    <scn:Attribute attributePath="Crash.FaultPerfm" value="0" />
    <scn:Attribute attributePath="Crash.CrashDrag" value="0" />
    <scn:Attribute attributePath="Jamming.JammingFlag" value="0" />
    <scn:Attribute attributePath="Jamming.JamPoint.JamPointECEF.X" value="0" />
    <scn:Attribute attributePath="Jamming.JamPoint.JamPointECEF.Y" value="0" />
    <scn:Attribute attributePath="Jamming.JamPoint.JamPointECEF.Z" value="0" />
    <scn:Attribute attributePath="Jamming.JamPoint.StartRange" value="0" />
    <scn:Attribute attributePath="Jamming.JamPoint.EndRange" value="0" />
    <scn:Attribute attributePath="Jamming.JamPoint.checkSensor" value="0" />
    <scn:Attribute attributePath="Jamming.JamCondition" value="0" />
    <scn:Attribute attributePath="Jamming.JammerInfo.Type" value="0" />
```

Fig. 11. Example of scenario description XML

시나리오 구조 해석기, 시나리오 명세 해석기 및 모델 연동기는 각각 class 형태로 구현되며 이들의 소스코드는 QUEST에서 자동 생성되는 모델의 소스코드와 함께 빌드되어 모의기 소프트웨어로 통합된다. 이렇게 빌드된 모의기 소프트웨어는 시나리오 생성기와 HLA/RTI 또는 TCP/IP 네트워크로 연결되어 시나리오 배포 관리 및 시뮬레이션 제어 명령에 대한 통신을 수행하게 된다.

본 논문에서 설명한 시뮬레이션 시나리오 프레임워크는 시나리오 구조와 느슨한 결합(loosly-coupled)으로 연관 관계를 가지고 있기 때문에 시나리오 구조 변경에 매우 유연한 특징을 가지고 있다. 즉, 시나리오 구조가 변경이 되더라도 시나리오 프레임워크 자체에는 영향을 미치지 않으며 프레임워크를 통하여 개발되는 시나리오와 모의기의 시나리오 정보에 대해서만 영향을 미치게 된다. 이러한 특징으로 인하여 시뮬레이션 시나리오 프레임워크는 목적 시뮬레이션 시스템의 시나리오 구조와 관계없이 재사용할 수 있으며 이를 통해 시뮬레이션 시스템의 시나리오 관련 기능에 대한 개발 비용을 대폭 감소시킬 수 있다.

당사에서는 QUEST와 시뮬레이션 시나리오 프레임워크를 통하여 4개 사업에서 4개의 시뮬레이션 시스템과 각 시스템을 구성하는 35종의 모의기를 개발하였다. 각 시스템에서 사용하는 시뮬레이션 시나리오 구조는 서로 상이하나 모든 모의기들은 동일한 시뮬레이션 시나리오 프레임워크를 재사용하여 개발되었기 때문에 시나리오 관리 기능 구현을 위한 비용이 소요되지 않았으며 이를 통하여 시뮬레이션 시스템에 대한 개발 기간을 대폭 감소시킬 수 있었다. 추후 개발되는 시뮬레이션 시스템에도 동일한 시뮬레이션 시나리오 프레임워크를 적용함으로써 많은 비용을 절감할 수 있을 것으로 예상한다.

5. 결 론

본 논문에서 설명한 시뮬레이션 시나리오 프레임워크는 시뮬레이션 시나리오의 구조와 모의기 모델의 속성간의 느슨한 연결을 통하여 시나리오 구조 변경에 유연한 구조를 갖추고 있으며 시뮬레이션 시스템에 관계없이 재사용 가능하다. 이러한 시뮬레이션 시나리오 프레임워크를 통하여 시뮬레이션 시나리오와 관련된 개발 비용을 대폭 감소할 수 있는 기반을 마련하였다. 실제로 시뮬레이션 시나리오 프레임워크 기반으로 개발된 시뮬레이션 시스템들은 종래의 방식보다 시나리오 관련 기능을 개발하는데 소요되는 시간과 비용을 많이 절감할 수 있었으며

앞으로 개발되는 시스템에도 같은 효과를 기대할 수 있을 것이다.

References

1. Hwang, K. C., Lee, M. K., Han, S. J., Yoon, J. M., Yoo, Y. J., Kim, S. B., Nah, Y. I., Kim, J. H., Lee, D. H., “The DEVS Integrated Development Environment for Simulation-based Battle experimentation”, *Journal of the Korea Society For Simulation*, Vol. 22, No. 4.
2. Lee, Y. H., Cho, K. T., Lee, S. Y., Hwang, K. C., Kim, S. H., “Framework and Tools for Rapid M&S Component Development and Reusability”, *Journal of the Korea Society For Simulation,* Vol. 22, No. 4.
3. Lee, Y. H., Cho, K. T., Lee, S. Y., Hwang, K. C., Kim, S. H., “Simulation framework development for GUI-based modeling & simulation environment”, *Proceeding of the Korea Society For Simulation*, 2013. 05.
4. Simulation Based Acquisition Industry Steering Group (SBA ISG), “SBA Functional Description - Version 1.1”, 1999.
5. Zeigler BP, Praehofer H and Kim TG., “Theory of Modeling and Simulation”, 2nd ed. San Diego, New York: Academic Press, 2000.
6. Shim, J.Y., Lee, Y.H., Kim, S.H., “A design of DLL Plug-in Framework for Component Reuse”, *Proceeding of the Korea Information Processing Society*, Vol. 17. No. 2.
7. Lee, Y.H., Shim, J.Y., Cho, K.T., Lee, S.Y., Kim, S.H., “The design of XML based object model for Publish-subscribe message protocol”, *Proceeding of the Korea Society For Simulation*, 2010. 05.
8. Lee, Y.H., Shim, J.Y., Kim, S.H., “An application of XML based Object odel for the development of RPR-FOM based HLA simulator”, *Proceeding of the Korea Society For Simulation*, 2010. 10.
9. Lee, Y.H., Shim, J.Y., Kim, S.H., “The Design of XML based Object Model Structure supports User-defined type”, *Proceeding of the Korea Information Processing Society*, 2010. 11.
10. Lee, Y.H., Shim, J.Y., Kim, S.H., “Applications of XML based Object Model for flexible binary data processing”, *Proceeding of the Korea Information Processing Society*, 2011. 05.

이 용 헌 (yongheon.lee@lignex1.com)

2007 포항공과대학교 컴퓨터공학과 학사

관심분야 : 모델링&시뮬레이션, HLA/RTI, 프레임워크, 객체지향 설계

이 승 영 (seungyoung.lee@lignex1.com)

2000 인하대학교 전자공학과 학사
2002 인하대학교 전산학과 석사

관심분야 : 모델링&시뮬레이션, SBA, T&E

김 세 환 (saehwan.kim@lignex1.com)

1985 경북대학교 전자공학과 학사
1987 경북대학교 전자공학과 석사
2006～2007 국방과학기술조사 기술전문가
2007～2008 국방과학기술조사 M&S분야 자문위원
2009～2010 국방과학기술수준조사 전문위원

관심분야 : 모델링&시뮬레이션, 프레임워크