

Plug-In 방식을 이용한 확장 가능한 비행임무지원체계 프레임워크 설계

강정훈, 김혜진
한국국방연구원

e-mail : kjhoun@kida.re.kr, im861025@kida.re.kr

Expandable Framework Design for Flight Mission Support System Using Plug-In Method

Jung Hun Kang, Hye Jin Kim
Korea Institute for Defense Analyses

요 약

첨단 항공기와 더불어 효과적인 조종사의 임무수행을 지원하는 소프트웨어인 비행임무지원체계의 개발은 많은 시간과 비용이 소요되며 높은 개발 수준을 요구하고 있는 고도의 소프트웨어이다. 이러한 소프트웨어의 개발에 있어 상호 연동성과 재사용성의 도입은 개발에 필요한 많은 시간을 줄이고 보다 효율적인 구조설계를 통해 품질을 보다 향상시킬 수 있다. 하지만 여러 복합적인 기능을 지원하는 항공기 특성상, 새롭고 다양한 기능을 접목한 비행임무지원체계의 개발은 빈번한 기능요구를 반영하기에는 개발 여력상 힘든 것이 현실이다. 따라서 본 논문은 확장 가능한 구조설계를 위해 Plug-In 을 통한 기능 확장을 구현함으로써 보다 효율적인 비행임무지원체계 프레임워크의 설계 및 개발 방향을 제시하고자 한다.

1. 서론

최근 우리 군에서는 F-15K 등과 같은 최첨단 전자장비를 갖춘 항공기들을 다양하게 도입하여 사용하고 있다. 이러한 첨단 항공기들의 운용을 위해서는 비행에 필요한 경로, 목표물, 수행 임무, 연료 계산 및 무장 장착이 정확하게 계획되어야 하는데, 이때 사용되는 것이 비행임무지원체계이다.

최근 제안된 비행임무지원체계 개발을 위한 프레임워크에서는 모듈화를 통한 재사용성의 향상을 가져올 수 있었으나, 최근 여러 임무를 통한 다양한 항공기의 기능의 확장을 따라가기 위해서는 보다 확장된 구조의 프레임워크가 필요하게 되었다. 공통된 구조를 기반으로 다양한 항공기와 임무를 고려하기 위해서는 특화된 Avionics Specific Component 설계가 필요하게 되었기 때문이다. 따라서 Plug-In 형식의 확장형 모듈을 통해 적용 항공기의 특성과 임무의 종류에 따라 필요 모듈을 적용하거나 제거할 수 있도록 하는 Plug-In 형식의 프레임워크 설계를 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존 프레임워크 연구들에서 나타난 특성과 문제점을 찾아보고, 3 장에서는 Plug-In 방식을 고려한 프레임워크 개발 프로세스에 대해 알아본다. 4 장에서는 Plug-In 방식의 확장 가능한 프레임워크설계 방법에 대해 살펴본 후 5 장에서 결론을 맺는다.

2. 관련 연구

2.1 JMPS(Joint Mission Planning System)

JMPS[1] 구축 프로젝트는 기존의 비행임무계획체계를 교체하고, 항공기, 무기, 센서들의 임무 수행원에게 잘 구조화된 비행 계획 도구를 제공할 목적으로 설계된 비행임무계획체계이다. JMPS 는 프레임워크, 공통컴포넌트, UPC(Unique Planning Component)로 구성되어 있어, 각종 플랫폼과 무기체계에서 공통의 컴포넌트들을 사용하도록 하여 소프트웨어의 중복개발을 방지하고, 상호 운용성을 증진시키도록 설계되어 있다. 하지만 JMPS 프레임워크는 지원하는 컴포넌트들의 구현에 있어 제한된 범위의 기능만을 지원하기 때문에, 실제 비행에 필요한 컴포넌트들을 프레임워크 상에서 확장 연동하는데 많은 제약이 있을 뿐만 아니라, 폐쇄적인 API 들을 지원함으로써 3rd party로부터의 다양한 확장 가능한 컴포넌트 개발을 완벽하게 지원하지 못한다는 단점이 있다.

2.2 KMSS(Korea Mission Support System)

KMSS[2]는 한국형 비행임무지원체계로서, 한국형 비행 임무에 맞도록 항공기, 무기, 지형, 센서 등의 임무 수행을 위한 요소들을 구조화하여 조종사에게 제공하기 위한 목적으로 설계된 프로젝트이다. KMSS 는 기반 컴포넌트와 확장 컴포넌트로 구성된 프레임워크와, 공통서비스, 그리고 지원하는 항공기와 장비, 특수 무장들을 위한 항공기 특화 컴포넌트인 ASC

(Avionics Specific Component)로 구성되어 있어 특정 항공기에 제한적이지 않고 다양한 항공기와 무장, 특수 장비 등을 지원하는 컴포넌트들을 연동할 수 있다. 하지만, 이 프레임워크 역시 폐쇄적인 API 지원을 통해 3rd party 로부터의 개발이 용이하지 않으며, 확장하려는 컴포넌트의 개발을 위해서는 기존 KMSS 체계에 대한 수정이 발생하는 경우도 있어, KMSS 프레임워크로부터의 지속적인 컴포넌트 확장이 불가능하고, 폐쇄적이며 수동적인 컴포넌트 개발지원이라는 단점이 있다.

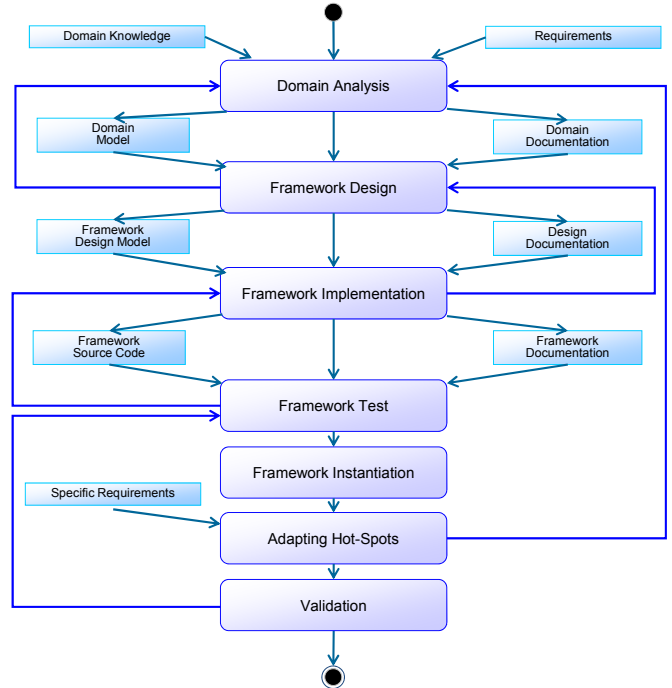
2.3 Plug-In 방식의 시뮬레이션 프레임워크

Plug-In 방식의 시뮬레이션 프레임워크 연구[3][4]에서는 시뮬레이션을 위한 프레임워크 개발 환경에서 Plug-In 방식을 통해 별도의 컴파일이나 작업 없이 시뮬레이션에 필요한 모듈을 필요에 따라 교체할 수 있도록 제안하고 있다. 같은 시뮬레이션을 수행해도 목적에 따라 시뮬레이션에 참여하는 모델의 특성이 바뀌게 되는데, 이러한 경우 기존의 방법에서는 모델의 내용이 변경되면 시뮬레이터 내부의 해당 모델의 내용을 수정한 후에, 다시 컴파일하고, 링크한 후에 시뮬레이션을 재개하는 작업이 필요하지만, Plug-In 방식을 사용함으로써, 모델을 바로 교환하여 시뮬레이션을 재개하는 것을 가능하도록 제안하고 있다. 하지만, 이러한 방식의 경우 정형화되고 공개된 API의 제공은 필수적이어야 하며, 이를 위해서는 상당히 고수준의 정형화된 프레임워크 구조 설계가 필요하다. 또한, 공통된 인터페이스를 통해 추가되는 모듈과 프레임워크간의 통신은 절대적인 신뢰도가 요구된다.

점차 다양화 되어가는 항공기 운용 추세와 더불어 다양한 전자장비를 통한 전자전이 가시화되어 가는 환경 속에서 다양한 무장과 장비를 지원하는 비행임무지원체계의 필요성은 점차 높아져 가지만, 현재 운용 중에 있는 비행임무지원체계 프레임워크는 그 지원이 제한적인 것이 사실이다. 따라서, 자율적인 컴포넌트 개발을 지원하고 다양한 항공기에 특화된 성능과 특수 장비와 무장의 특성이 반영된 컴포넌트들을 개발하는데 있어, 기존 프레임워크의 수정이 발생하지 않으면서도 확장 및 재 조합을 위한 모듈 설계가 필요하게 되었으며, 이를 위해 공통된 형식의 인터페이스를 지원하고 다양한 개발을 지원하는 공개된 공용 API 를 통해 Plug-In 방식의 확장 가능한 프레임워크 설계의 필요성은 절실하다.

3. Plug-In 방식을 고려한 프레임워크 개발 프로세스

Plug-In 방식을 고려한 프레임워크 개발 프로세스를 개발하기 위해, 비행임무지원체계 개발에 필요한 군사정보, 무장, 통신, 기상 등의 여러 자료 공유는 필수적인데 이를 지원함과 동시에 프로세스를 분석하기에는 매우 힘든 것이 사실이다. 따라서 비행임무지원체계에 가장 적합한 프레임워크 개발 프로세스를 정의하고 적용하기 위해서는 프레임워크 개발에 적합한 프로세스를 (그림 1)과 같이 크게 7단계로 나누었다.



(그림 1) Plug-In 방식을 고려한 프레임워크 개발 프로세스

①Domain Analysis 는 비행 임무지원체계에서 공통적으로 필요한 기능들을 식별(Domain Scoping)하는 단계이다. 충분한 도메인 지식과 체계 요구사항들을 기반으로 프레임워크를 개발하는데 있어서, 프로그램으로 개발되었을 때의 중요 기능 요소들을 식별하는 것이 먼저 선행되어야 한다. 또한 프레임워크가 개발된 이후, Plug-In 모듈 개발을 위한 작은 규모의 Domain Analysis 가 이루어져야 한다.

②Framework Design 은 ①단계의 산출물을 기반으로 프레임워크에 대한 설계를 진행한다. 각각의 기능요소들 중에서 Plug-In 연동 요소(Interface)를 식별하는 것이 중요하다. 특히, 이 Plug-In 요소는 프레임워크를 사용하여 실제 비행임무지원 체계 개발 시, 항공기에 따른 특정 정보 요구 사항(항공기 및 무장 상태 등)을 임무가 부여된 항공기에 맞게 Plug-In 과 프레임워크 간 데이터 연동이 이루어져야 하는 부분으로, 정의된 Plug-In Interface 를 바탕으로 Service 와 모듈간의 설계가 완성되어야 한다.

③Framework Implementation 에서는 개발을 위해 선택한 프로그래밍 언어 및 .Net 과 같은 플랫폼 지원 기술을 통해 프레임워크를 구현하게 된다. 이 단계의 Framework Documentation 은 프레임워크를 사용하여 특정 프로그램을 개발할 때 반드시 참고하여야 하는 문서로서, 프레임워크의 목적 및 사용 방법, 설계 디자인에 따라 세부적이고 명확하게 기술되어야 한다.

④Framework Test 단계에서는 프레임워크가 사용자의 요구사항을 만족하는지 그리고 비행임무 지원체계에 요구하는 기능들을 개발할 수 있는지에 대해 시험하며, Plug-In 모듈 장착 시 정상적인 운용이 가능한지 기능을 시험한다. 이 시험이 완료되면 유효성 검사를 거쳐 최종 산출물인 Plug-In 방식의 비행임무

원체계를 위한 프레임워크가 나오게 된다. 프레임워크 개발 후에 추가적으로 변경을 요하는 경우 먼저 ⑤프레임워크를 실체화(Instantiation)하고 요구 기술에 맞는 특수한 요구 사항을 적용하여 ⑥Plug-In 확장에 따른 요구사항 반영 지점의 클래스들을 구체화시키는 과정이 추가될 수도 있다. 요구사항에 맞는 Plug-In의 추가로 인해 새로운 개발 사항이 발생하는 경우, 다시 1 단계인 Domain Analysis로 되돌아가 개발된 프레임워크를 진화시킴으로써 보다 기능면에서 요구 충족도를 높일 수 있는 프레임워크 설계 모델을 제시할 수 있다.

클래스의 구체화 과정을 통해 구축된 프레임워크 개발 프로세스는 ⑦유효성 확인(Validation)이 필요하다. 이를 통해 Framework에서 요구하는 특수한 요구 사항들에 대해 결함이 없는지, 혹은 유효한 요구 사항인지 확인을 거치게 된다. 만약 7 단계의 개발 프로세스 단계를 통해 개발된 프레임워크가 요구사항을 충분히 만족시키지 못하거나 Test 단계에서는 검출되지 않은 결과값에 대한 무결성에 문제가 발생한 경우, 다시 Test 단계로 되돌아가 해당 기능에 대해 점검을 하고 수정을 거치게 된다. 만약 Test 단계에서 문제에서 발생한 문제가 수정을 요하는 경우, Implementation 단계로 인계되어 요구사항에 대한 정확한 수정을 거쳐 다시 Test 단계를 거침으로써 보다 기능 무결성도를 높일 수 있는 프레임워크 설계 모델을 제시할 수 있다.

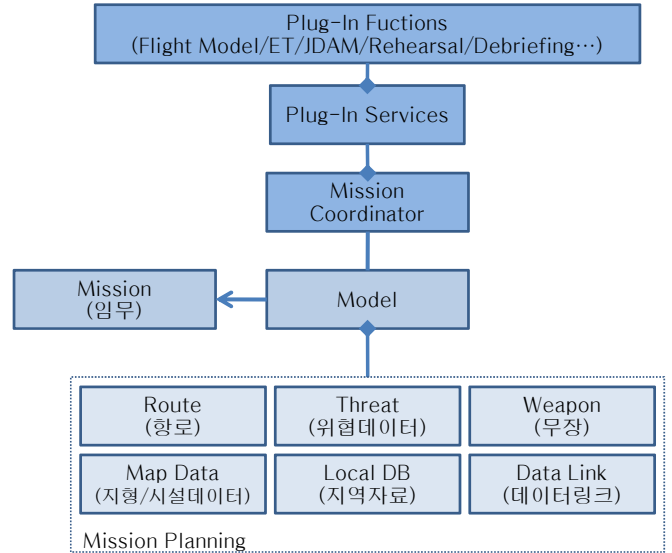
4. Plug-In 방식의 확장 가능한 프레임워크 설계

이 장에서는 Plug-In 방식을 적용한 확장 가능한 비행임무지원체계의 프레임워크 설계를 진행하도록 한다.

4.1 Domain Analysis

Domain Analysis를 분석하기 전에 먼저 비행 업무를 이해할 필요가 있다. 비행 업무의 주요 단계로는 비행계획, 비행준비 및 리허설, 비행 후 디브리핑으로 나눌 수 있다. 또한 정찰 임무와 같은 항공기 특성이 적용된 임무 설정이 반영되기도 하며, 임무에 따라 다양한 무장 체계가 접목되기도 한다. 이러한 기능을 담당하는 비행임무지원체계는 기본임무계획[항로, 시간, 거리, 연료 등], 위협분석, 공격제원 산출, 임무보고서 작성, 지형/지도처리, 임무데이터 관리, 체계연동 등의 컴포넌트들로부터 요구되는 데이터를 받아 처리되어야 한다. 그러나 각 모델의 각종 매개변수들은 임무에 의해 요구되는 데이터들이 결정된다. 따라서 임무는 임무지원체계 전반에 영향을 주는 주요 요소이며, 진행되는 주체가 된다. 임무에 따라 항공기에 장착되는 장비들이 달라지며, 임무의 특성에 따라 항공기의 설정 상태는 수시로 바뀌게 된다. 따라서 임무에 따라 요구되는 기능을 바로 적용할 수 있도록 Plug-In 할 수 있어야 하며, 그러한 기능이 별도의 프레임워크의 수정이나 컴파일 등의 과정 없이 바로 체계에 반영되게 하기 위해서는 Plug-In 형식의 모듈 추가 기능이 필요하다. 그리고 연동되는 Plug-In 과 프레임워크와의 통신을 담당하는 Plug-In Service가 필요

하며, Mission Coordinator가 이러한 Plug-In Service와 임무에 필요한 데이터를 통합 관리하는 역할을 담당할 수 있어야 한다. 이와 같이 설명된 Domain Analysis를 통해 구성된 Domain Model을 작성하면 (그림 2)와 같다.



(그림 2) 비행임무지원체계 프레임워크 Domain Model

4.2 Framework Design

Framework Design 단계에서는 Domain Analysis 단계의 산출물인 Domain Model과 기본 업무 흐름을 바탕으로 프레임워크를 구성하는 클래스들을 디자인하고 이 클래스들 간의 관계를 정의한다. 이 단계에서 가장 중요한 것은 비행임무지원체계의 임무마다 변할 수 있는 메소드는 Plug-In, 임무에 종속적이지 않고 항공기 운용에 변하지 않는 메소드들은 프레임워크로 정의한다.

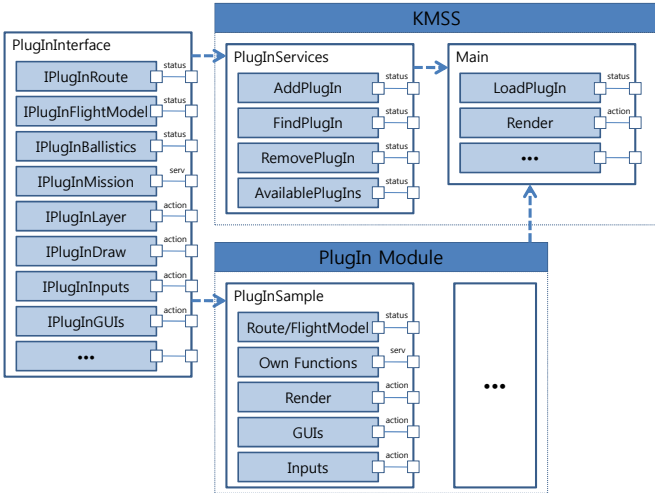
1) Plug-In Interface Design: Plug-In에서 가장 중요한 것은 Plug-In 모듈과 프레임워크간에 미리 규약된 인터페이스를 통해 데이터가 전송돼야 한다. 따라서 Interface를 정의하는 것이 선행되어야 하며, 인터페이스는 기능의 대부분을 프레임워크 안의 적합한 모듈로 전달해 주어야 한다. 따라서 항로와 임무, 그리기도구와 GUI, 그리고 항공기 특성과 항공기 상태 정보 등을 망라하는 인터페이스 정의가 적용되어야 하며, Plug-In 모듈과 서비스 모듈은 이 인터페이스 모듈에 정의된 방식에 따라 상호간 데이터 교환이 정확히 이루어져야 한다.

2) Plug-In Services 구현: 특정 Plug-In 모듈로부터 전달되는 데이터를 프레임워크의 내부 요소에 전달하고, 내부 요소에서 생성된 데이터는 다시 특정 Plug-In으로 전달될 수 있도록 관리하는 역할을 하며, 이 Service 모듈은 Plug-In Interface에서 정의된 방식에 맞추어 데이터를 전달하는 기능을 담당해야 한다.

3) Plug-In Manager: 프레임워크 내부에서 추가되는 Plug-In 모듈을 통합 관리하는 모듈을 설계할 필요가 있는데, 이에 해당하는 것이 Plug-In Manager이다. 프레임워크가 실행될 때, 각각의 DLL(Dynamic Library Link)들로 이루어진 Plug-In Module들을 체계 실행 시

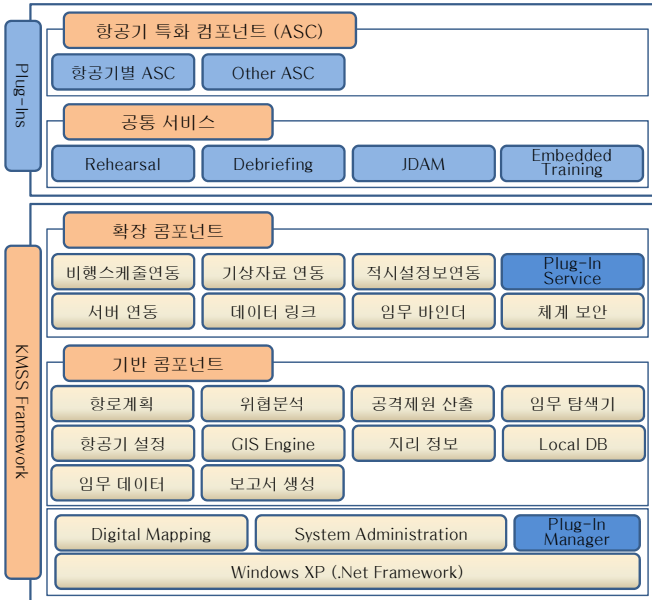
인지하고, 이 중 체계에 적용 가능한 모듈들을 목록화 하여, Plug-In Service 로 넘기는 역할을 한다.

4) Plug-In Module 구현: 실제 프레임워크에서 독립된 특정 목적의 기능을 모듈화하고 Plug-In Interface 에서 정의된 방식을 따라 구현됨으로써, 프레임워크 내부 요소들과 긴밀한 데이터 전송이 구축되어야 한다.



(그림 3) Plug-In 기반 프레임워크 처리 구조

Framework 개발 프로세스에 따른 Domain Analysis 와 Framework Design, Plug-In Module 적용을 통해 국내 현실에 적합하고, 보다 임무 요구에 능동적으로 반영할 수 있는 비행임무 지원체계의 프레임워크 설계 방법을 제시하였으며, 이 결과, (그림 4)과 같은 Plug-In 기반의 확장형 비행 임무지원체계 프레임워크의 정의를 도출할 수 있다



(그림 4) Plug-In 기반의 확장형 프레임워크 정의

기본 프레임워크로는 주요 비행임무지원을 담당하는 범용적 기능을 기본 컴포넌트로 묶어 프레임워크를 구성하며, 비행에 있어 참조가 될 수 있는 데이터 연동이나 정보 획득과 관련된 부분을 확장 컴포넌트

로 묶어 구성하였다. 프레임워크 하부에 Plug-In Manager 를 두어 Plug-In 의 지원 목록과 적용 가능 모듈의 체계 내 로드 여부를 판단하고, 확장컴포넌트에서는 Plug-In Service 모듈이 로딩된 Plug-In 을 통해 Interface 를 공유하고 이벤트와 자료를 기본 프레임워크로 전달하는 일련의 과정을 처리하도록 하였다.

그 외에 Flight Model 이나 JDAM(Joint Direct Attack Munition), Debriefing 과 같은 항공기의 특성과 임무 및 역할에 따라 각기 달리 적용되는 모듈들을 Plug-In 으로 도출하여 필요에 따라 부수적인 역할을 담당하는 모듈들을 Plug-Ins Group 으로 독립함으로써, 기본 프레임워크를 수정하지 않고도 임무와 필요성에 따라 적합한 모듈을 Plug-In 함으로써 모듈 추가에 따른 별도의 컴파일 과정이나 프레임워크의 수정이 없이 재사용에 따른 효율성을 높일 수 있도록 설계하였다

5. 결론

본 논문에서는 국내에서 운용하는 차기 항공기를 지원하는 비행 임무 지원체계를 개발하는데 있어서, 다양한 비행임무 지원체계의 요구사항을 프레임워크의 추가적인 수정을 배제하고 확장 가능한 구조설계를 지원하기 위해 Plug-In 을 통한 기능 확장을 구현함으로써 보다 국내 환경에 적합한 체계를 보장해 줄 수 있는 효율적인 비행임무지원체계 프레임워크의 설계 및 개발 방향을 제시하였다.

향후 연구 방향으로서는 제안된 프레임워크에 대해 Error Recovery 기술 적용 및 Model Checking 방법을 통한 모듈 별 검증 및 효율적인 유효성 검토 방법 연구가 진행될 예정이다.

Acknowledgement

본 연구는 지식경제부의 산업 기술 융합 산업 원천 기술 개발 사업 기술 과제에 의해 수행되었음 (KI001697-2010-04, 항공기 임베디드 시스템 개발)

참고문헌

- [1] William R. Clemons, "The Case for Consolidating Tactical and Operational Systems," The Land Warfare Papers, No.52, September 2005.
- [2] 강정훈, 양주석, 김성원, 오영민. "한국형 비행임무 지원체계 개발을 위한 프레임워크 설계," 제 30 회 한국정보처리학회 추계학술발표대회 논문집 제 15 권 제 2 호, pp. 441-444, 2008.11
- [3] 배장원, 김탁곤, "플러그인 방식을 이용한 연동 가능한 시뮬레이션 프레임워크와 환경," 한국군사과학기술학회 2009 년 종합학술대회, pp. 393-396.
- [4] Jan Himmelspach and Adelinde M. Uhrmacher, "Plug'n simulate," Proceedings of the 40th Annual Simulation Symposium(ANSS'07), pp.137-143, Mar. 2007.
- [5] Davis K.P. and Anderson A.R., "Improving the Composability of Department of Defense Models and Simulations," RAND technical report, 2003