

국방무기체계 모델 개발을 위한 SysML 적용 사례 연구

장재덕*, 최상택*, 정운호*, 최상욱*, 최광목**
*LIG 넥스원(주)
**MDS 테크놀로지(주)
e-mail : jaedeok.jang@lignex1.com

A Case Study on Application of Model Development for Defense Weapon System Using SysML

Jae-Deok Jang*, Sang-Taik choi*, Sang-Wook Choi*, Yun-Ho Jung*, Kwang-Mook Choi**
*LIG Nex1
** MDS Technology

요 약

본 논문은 대형복합 체계인 국방무기체계를 설계 프로세스에 따라 SysML 을 적용하여 개발하는 방법 및 사례에 대해 다룬다. ISO-IEC 15288 에 제시된 기술 프로세스의 핵심 단계인 요구사항 분석, 기능 분석, 아키텍처 설계를 모델 기반으로 수행하는 것은 이해관계자 사이의 의사소통을 원활히 유지하는 효과적인 방법이다. 더불어 SysML 을 기반으로 작성된 체계 모델은 S/W 설계를 위해 많이 사용되는 UML 과 설계 일관성을 유지하기 용의하다. 본 논문은 SysML 을 통해 체계 개발에 필요한 단계별 모델의 종류와 사용방법을 대공유도무기 사례를 통해 제안하고자 한다.

1. 서론

최근 체계 모델링에서 소프트웨어 모델링까지 일관성을 유지하기 위해 UML(Unified Modeling Language) 과 SysML 을 많이 활용한다. UML 은 객체관련 표준화 기구인 OMG 에서 1997 년 11 월 객체 모델링 기술, OOSE(Object Oriented System Engineering) 방법론 등을 연합하여 만든 통합 모델링 언어로 주로 소프트웨어 분야에서 활용되고 있다. 그러나 INCOSE 와 OMG 에서는 2004 년 부터 시스템 엔지니어링 모델링 언어로써 활용을 목표로 둔 SysML 의 출시를 위해 OMG 를 중심으로 여러 학제에 걸쳐 개발의 노력을 기울이고 있다. 대표적인 연구 사례를 살펴보면 Jon Holt 는 UML 을 통한 비 기능 요구사항의 추적성을 확보하기 위한 노력의 일환으로서 새로운 스테레오 타입을 정의하고 이를 사용한 요구사항 사이의 관계 정립을 위한 방안을 제시하고 있다. Axelsson Jacobson 은 연속적인 시간에 따른 거동의 표현을 UML 의 확장 모델에 대해 제시하고 있다. 시스템의 요구사항을 표현하는 UML 활용 방안에 대해서도 많은 연구가 이루어지고 있는데 Ivar Jacobson 은 유스케이스(Use Case)를 “시스템에서 수행해야 하는 업무 처리”로 정의하고 유스케이스 사용에 관한 사용법과 유스케이스로 표현할 요구사항의 식별 방안에 대해 제시하였다. Larman 은 UML 을 사용한 성능 요구사항 및 제약 요구사항에 대한 표현 방법으로 노트(Note) 표기법을 사용하는 방안에 대해 제시하였다. 또한 UML 이 시스템 엔지

니어링 수준에서 어떻게 사용되어야 하는지에 대해서 Howard Lykins 는 OOSEM 기법을 활용한 방안을 제시하고 있다. 이외에도 RSI(Requirements/Service/Interface) 접근법과 같은 많은 방법들이 제시되고 있다. 이러한 부분적인 UML 활용방안은 2001 년 SE DSIG(Systems Engineering Domain Special Interest Group)가 시스템 엔지니어링을 위한 UML 의 요구사항으로 수렴하여 요구사항 개발을 지원하고 있다.

이러한 연구가 진행되고 있음에도 불구하고 UML 을 활용한 시스템 엔지니어링 프로세스 구현을 위한 명확한 방안이 제시되고 있지 못하고, 시스템 엔지니어링 프로세스 전반에 걸친 활용법 또한 아직 제시되고 있지 않은 실정이다. 따라서 통합 모델링 언어로서 UML 은 많은 한계점이 존재한다.

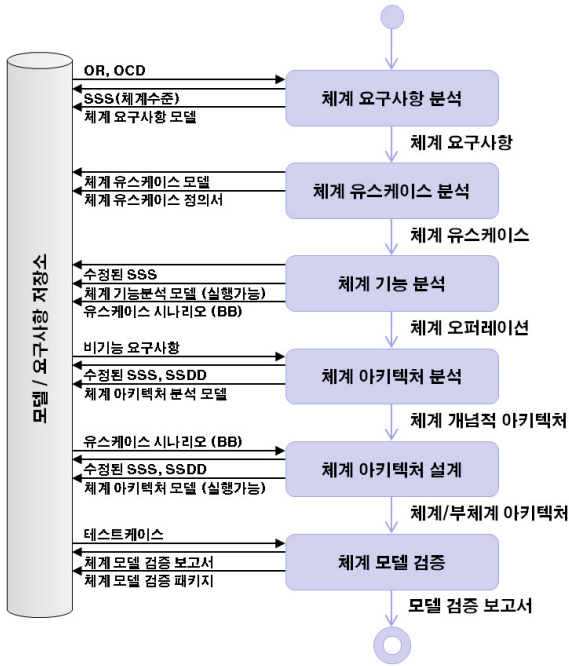
UML 은 체계 모델링을 위한 통합 언어로서 사용하기에는 요구사항 구조, 성능과 물리적 특성, 체계 계층 구조의 표현 등에 대한한계점을 가지고 있었고, 이를 극복하기 위해 확장한 것이 SysML 로 UML 에 다이어그램 및 문법을 추가하여 체계의 분석, 설계, 검증을 지원하기 용이하게 한 모델링 언어이다.[1]

2. 설계 프로세스와 MBSE 방법

ISO-IEC 15288 과 DoD Instruction 5000.02 에 따르면 획득단계별 체계를 개발하는 과정을 기술 프로세스라고 정의한다. 기술 프로세스는 이해관계자 요구사항 정의, 요구사항 분석, 아키텍처 설계, 구현, 통합, 검

증, 배치, 확인의 과정으로 정의된다.[2][3]
 기술 프로세스 중에서도 요구사항 분석, 아키텍처 설계는 흔히 설계 프로세스로 정의하는데 ISO/IEC 15288 의 기반이 되는 MIL-STD-499B 에는 설계 프로세스를 요구사항 분석, 기능분석/할당, 설계 합성 (Synthesis), 체계 분석 및 통제로 정의하고 있다.

MBSE 를 진행하기 앞서 적합한 프로세스의 정의와 프로세스 단계별 수행 활동을 정의하는 것이 무엇보다 중요하다. 일반적으로 모델링 언어인 SysML 의 문법 자체를 알지 못해 적용에 어려움을 겪는 경우보다는 적용을 위한 프로세스, 프로세스 단계별 주요 활동, 이 활동에 알맞은 모델이 무엇인지 알지 못해 MBSE 방법을 적용하는데 어려움을 겪는 경우가 많다.[4] 따라서 프로세스를 진행하기 전 프로젝트 혹은 사례에 적용할 프로세스를 명확히 하는 것이 중요하다. 경우에 따라서 표준 프로세스를 조정(Tailoring)하는 것이 효과적일 수 있다. 그림 1 은 SysML 적용을 위한 MBSE 개발 절차를 도식화 한 것이다.



(그림 1) SysML 기반 체계모델 개발 절차

요구사항 분석 단계는 체계가 무엇을 해야 하는지 (기능적 요구사항)와 얼마나 잘 수행해야 하는지(성능적 요구사항)에 대해 이해관계자 요구사항을 체계 요구사항으로 변환하는 단계이다. 이를 위해 이해관계자 요구사항을 분석하여 체계 요구사항을 생성하고 체계 Use Case 를 정의하여 체계의 운영을 묘사한다.

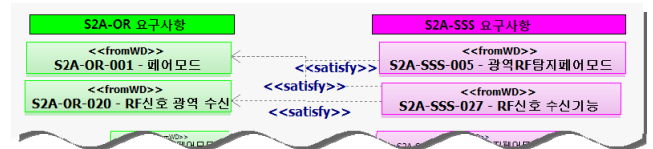
체계 기능 분석 단계는 기능적 체계 요구사항을 체계 기능으로 논리 정연하게 묘사하는 것으로 전 단계인 요구사항 분석에서 정의된 체계 수준의 Use Case 를 실행 가능한 모델로 변환하는 단계이다. 이를 위해 Activity Diagram, Sequence Diagram, State Chart Diagram 을 활용하여 Use Case 시나리오를 정의, Use Case 기능적 흐름을 도출, 인터페이스 정의, 상태기반 거동 도출 및 Use Case 모델의 검증 활동을 수행한다.

설계 종합 단계는 요구된 기능을 수행하는 물리적 아키텍처를 개발하는 것으로 설계 개념의 최적화를 다루는 아키텍처 분석 단계, 최적화 결과를 구조화하는 아키텍처 설계 단계, 전 단계의 아키텍처를 하위 수준으로 분해하는 것뿐만 아니라 포트와 인터페이스를 정의하는 상세 아키텍처 설계로 구성되어 있다. 이를 위해 Block Definition Diagram (BDD), Internal Block Diagram(IBD)를 활용한다.[5][6]

3. 프로세스 단계별 모델 개발

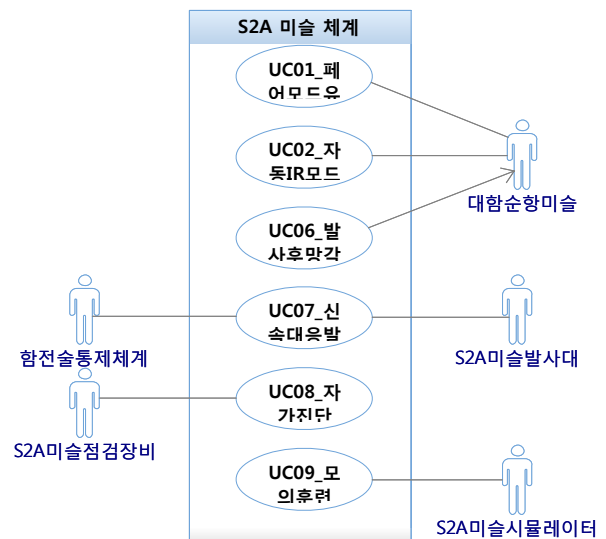
3.1 체계요구사항 및 유스케이스 분석

국방무기체계 개발은 소요군으로부터 요구사항을 수집하는 것으로부터 시작한다. 최초 수집된 요구사항을 원요구사항(Originating Requirements)라하고 이를 바탕으로 체계를 개발할 수 있는 기술 요구사항으로 전환하여 체계 요구사항이라 지칭한다. 이때 수집된 요구사항을 요구사항 모델로 표현한다. 그림 2 는 수집된 요구사항 일부를 모델로 표현한 것이다.



(그림 2) 요구사항 다이어그램

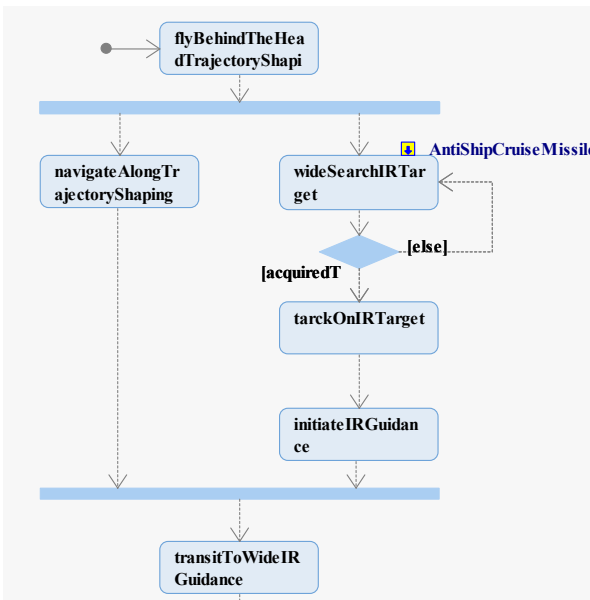
요구사항 다이어그램에서와 같이 원 요구사항에 대한 기술 요구사항이 정의되면 체계에 대한 정황을 정의하기 위해 유스케이스 다이어그램을 작성한다. 유스케이스 다이어그램은 요구사항 분석 과정에서 식별된 체계 범위, 외부체계, 체계 주요기능들을 체계 경계, 액터, 유스케이스로 정의한다. 유스케이스 다이어그램은 액터와 유스케이스 간 상호작용 관계 및 체계 범위에 대한 내용을 중심으로 작성한다. 그림 3 은 유스케이스 다이어그램을 표현한 것이다.



(그림 3) 정황 정의를 위한 유스케이스 다이어그램

3.2 체계기능 분석

기능 분석 단계에서는 시스템 수준의 기능 요구사항을 기반으로 임무 목표를 달성하기 위한 상세 거동을 정의함으로써 시스템 기능을 세분화 한다. 유스케이스 다이어그램을 통해 체계에 대한 주요 기능이 유스케이스로 식별되면 각 유스케이스에 대한 액티비티 다이어그램(Activity Diagram)을 작성하여 체계의 거동(Behavior)을 정의한다. 액티비티 다이어그램은 체계가 수행하는 액션을 제어흐름에 따라 표현하는 것이다. 이때 필요에 따라 추가적인 요구사항이 등장할 수 있는데 이를 식별하여 파생 요구사항(derived Requirements)으로 정의하고, 요구사항 다이어그램에 체계 요구사항과 추적 관계를 적용하여 포함한다. 그림 4는 액티비티 다이어그램을 표현한 것이다.

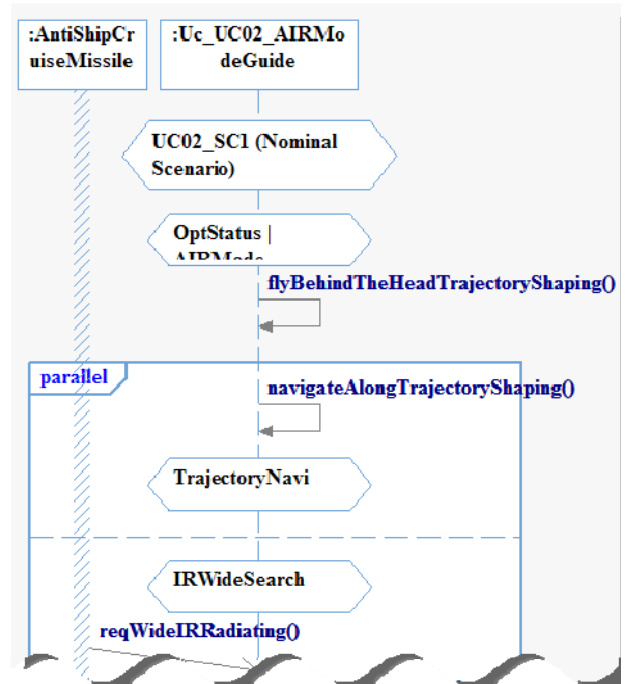


(그림 4) 거동 표현을 위한 액티비티 다이어그램

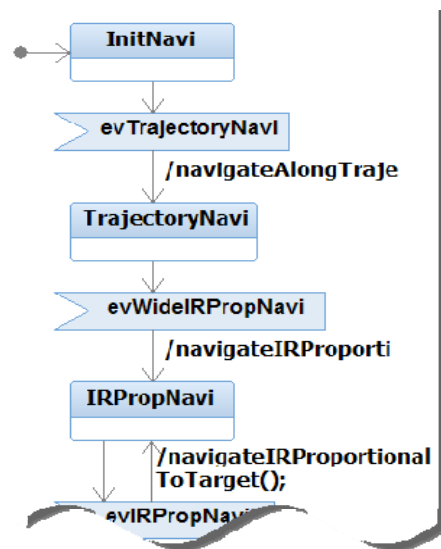
각 유스케이스에 대한 액티비티 다이어그램이 작성되면 특정 시나리오 흐름에 대하여 시퀀스 다이어그램을 생성한다. 시퀀스 다이어그램에는 유스케이스 다이어그램에서 식별된 외부 체계와 특정 유스케이스 사이의 시나리오를 표현한다. 주로 외부 체계에서 발생된 이벤트에 대상 체계가 어떻게 반응을 하는지를 주로 표현하게 되고, 이때 액티비티 다이어그램에서 식별된 모든 액션은 최소 한번 이상 표현되어야 한다. 그림 5는 시퀀스 다이어그램 예제이다.

시퀀스 다이어그램 분석 결과 상태 전이가 식별되면 상태 차트(State chart)에 해당 상태 전이를 반영한다. 또한 시퀀스 다이어그램 상의 이벤트, 메시지, 제약사항을 분석하여 상태와 상태 사이의 전이(Transition)를 정의하는 근거로 사용한다. 상태 차트와 시퀀스 다이어그램 및 액티비티 다이어그램은 서로 보완적이다. 경우에 따라서 상태차트가 먼저 작성되는 경우가 있을 수 있다. 이때는 주로 기능을 더 확장하여 식별하기 쉽지 않는 경우이다. 상태차트는 시스템의 상태 및 모드를 정의하는 것이다. 시스템에

서 기대되는 환경과 사용 의도를 분석하여 시스템이 겪게 될 상태(State)와 모드(mode)를 정의하고, 시스템이 반드시 수행해야 할 기능 관점에서 시스템의 운영 거동과 지원 거동을 도출한다. 이때 시스템의 상태 및 모드는 기능을 도출하기 위해 기능을 추론해가는 과정을 보다 쉽게 접근하기 위한 방법으로 사용된다.[7] 그림 6은 상태차트 예제를 표현한 것이다.



(그림 5) 시나리오 표현을 위한 시퀀스 다이어그램



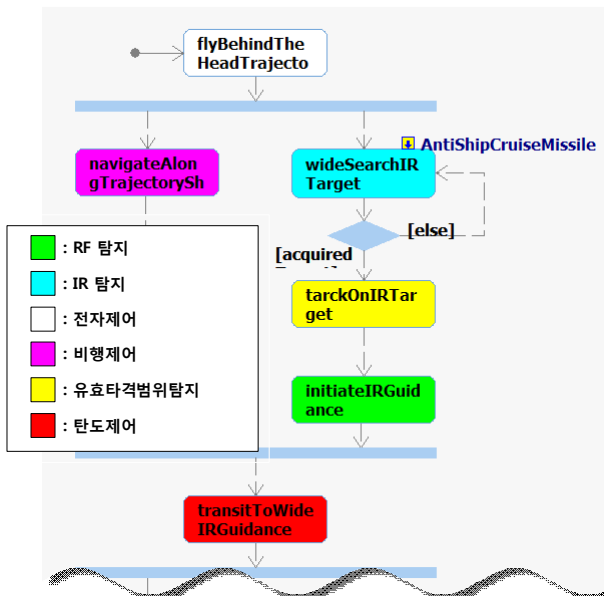
(그림 6) 상태전이 및 검증을 위한 상태차트

3.2 체계아키텍처 분석 및 설계

체계 아키텍처 분석은 주요 체계 기능을 체계/부체계 구조적 관점에서 식별하고, 해당 기능을 위한 후보 솔루션들을 정의한다. 그리고 체계 기능 분석 모

델에서 식별된 요구 기능들을 충족하는 최우수 솔루션을 결정하기 위해서 MOE(Measure of Effectiveness)를 기반으로 절충분석(Trade-off)을 수행하고, 이 분석 결과에 따른 주요 기능 최적 솔루션을 결정한다. MOE는 정확성, 주요 성능, 비용, 정비성 등이 될 수 있다.

각 MOE는 우선순위와 중요도에 따라 가중치가 부여되고 이 것은 정량적 혹은 정성적인 방법을 통해 평가되어 최적 솔루션이 결정된다. 후보 솔루션을 도출하기 위해서 먼저 시행되어야 하는 것은 활동 다이어그램에서 작성된 액션 즉 체계 기능을 중심으로 물리적 구조 관점에서 분류하여 그룹화 하여야 한다. 그림 7은 활동다이어그램의 액션을 그룹화한 예제이다.



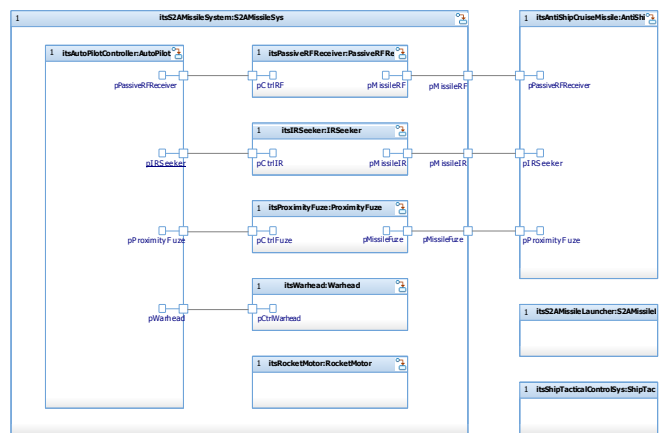
(그림 7) 체계기능의 물리적 구조로 분류

체계의 기능을 물리적 구조 관점에서 분류 시 응집력을 높이기 위해 기능인터페이스가 가장 적은 것들로 분류를 한다. 이후 효과 척도를 기준으로 최적 솔루션을 부체계로 정의한다. 이후 과정은 앞서 체계의 각 유스케이스에 대한 액티비티 다이어그램, 시퀀스 다이어그램, 상태 차트를 정의한 것과 같이 식별된 부체계를 대상으로 각 다이어그램을 반복 작성하면서 부체계에 대한 기능 정의를 수행한다. 아키텍처 설계는 부체계 기능이 분해되고, 정의되어 부체계에 할당되고, 부체계간 물리적 인터페이스가 정의되면 비로소 완료된다. 그림 8은 체계 아키텍처 예제이다.

4. 결론

본 논문은 SysML을 통해 시스템 엔지니어링 설계 프로세스인 요구사항 분석, 기능 분석, 아키텍처 설계 활동을 수행하는 과정을 유도하기 시스템을 예제로 제공하였다. SysML은 시스템 엔지니어링 통합 모델링 언어로서 다양한 다이어그램을 통해 시스템을 조망하는 여러 관점(View)을 제공하는 유용한 언어이다.

앞으로도 시스템에서 소프트웨어의 비중은 점차로 증가할 것이다. 소프트웨어 설계와 시스템 설계의 일관성을 유지하고, 통합된 모델을 적용 한다는 측면에서 SysML을 사용하여 MBSE를 수행하는 것은 매우 중요하게 고려되어야 할 점이다. SysML은 요구사항 관리, 요구사항과 설계 내용의 추적, 시스템의 계층적 표현 및 구조 표현, 시스템 거동 및 시나리오와 시스템 분석을 지원하는 모델과 표준화 방법을 제공한다. 그러나 UML 및 SysML을 사용하는 많은 영역에서 그 사용하는 방법과 표현법이 다양하다. 이는 프로세스의 부재에서 오는 영향일 수도 있고, UML 및 SysML의 사용되는 대상 시스템이 달라 발생하는 불일치 일 수 있다. 본 논문을 통해 이러한 다양한 사용법이 조금은 개선되고 일관성을 가지길 희망한다.



(그림 8) 체계 아키텍처

참고문헌

[1]S. Friedenthal, R. Steiner, and A. C. Moore, "Practical Guide to SysML: The Systems Modeling Language", Elsevier Science, 2008.
 [2]DAU, "DAG Guidebook", p 210, 2012.
 [3]ISO/IEC. "ISO/IEC 15288: Systems Engineering - System Life Cycle Processes", , p 22, ISO, 2008.
 [4]Bruce Powel Douglass, "Learning the ROPES", Embedded Design India, 2001.
 [5]Hans-Peter Hoffmann , "SysML-Based Systems Engineering Using a Model-based Development Approach", Proceedings of INCOSE 16th International Symposium, 2006.
 [6]Bruce Powel Douglass, "Systems Engineering Best Practices with the Rational Solution for systems and Software Engineering", Systems Engineering Best Practices Deskbook, v.3.1.2, 2011
 [7]Jaedeok Jang, "On System Modeling Framework for System Design Processes Using UML, Ajou University Master Thesis, 2004.