

한국형 비행임무지원체계 개발을 위한 프레임워크 설계

강정훈^o, 양주석, 김성원, 오영민
한국 국방 연구원

e-mail : kjhoun@kida.re.kr^o, { executor97, swmax, ntlover }@naver.com

Framework Design for Korean Flight Mission Support System Development

Jung Hun Kang, Ju Seok Yang, Sung Won Kim, Young Min Oh
Korea Institute for Defense Analyses

요 약

효과적인 조종사의 임무수행을 지원하는 소프트웨어인 비행임무지원체계가 현대 항공작전의 필수 요소로 주목 받고 있다. 첨단 항공기를 위한 비행임무지원체계의 개발을 위해서는 많은 시간과 비용이 소요되며 높은 개발수준을 요구하고 있으나, 국내 실정은 기본 프레임워크조차 없는 단일 응용 프로그램 개발 수준에 불과하다. 따라서 차기 항공기를 지원하는 향상된 한국형 비행임무지원체계를 개발함에 있어, 재사용이 가능하며 품질을 향상시켜줄 수 있는 프레임워크의 개발은 매우 중요한 과제이다. 이에 본 논문에서는 한국형 비행임무지원체계의 개발에 적합한 개발 프로세스를 제시하고, 이에 따른 비행임무지원체계의 프레임워크의 설계 및 개발 방향을 제시하고자 한다.

1. 서론

최근 군에서는 최첨단 전자장비를 갖춘 항공기들이 다양하게 도입되어 사용되고 있다. 이러한 첨단 항공기들의 운용을 위해서는 비행에 필요한 경로, 목표, 수행 임무, 연료 계산 및 무장 장착이 정확하게 계획되어야 하는데, 이때 사용되는 것이 비행 임무 지원 체계이다. 비행 임무지원체계는 각각의 항공기의 특성을 고려하여 비행에 필요한 경로나 목적, Radio 채널, 무장 장착 및 연료 급유 등 항공기 내/외부에서 준비해야 할 일들을 결정(Planning)하고, 리허설(Rehearsal) 기능을 통해 이륙을 하지 않고도 비행 경로에 대한 위험 지역이나 상승/하강시기 등을 미리 시연해 볼 수 있다. 또한, 비행 후에는 저장된 비행 데이터를 기반으로 디브리핑(Debriefing)함으로써 비행 도중 발생된 일이나 실제의 비행 경로 등을 추적해 볼 수 있는 체계이다.

하지만 현재 사용되고 있는 비행임무지원체계는 대부분이 해외에서 도입된 것이며, 유지 보수 및 필요 사항에 대한 개선 요구 시, 막대한 비용과 더불어 즉각적인 수정이 불가능하다는 문제점을 가지고 있다. 또한 체계 플랫폼이 해외 환경을 기반으로 개발되어 있어 국내 항공정보 DB 나 군 정보체계와의 연동 운영 및 상호 호환성이 보장되지 않기 때문에 많은 문제점들이 발생되고 있다[1].

항공기의 경우, 기본적인 기능 블록의 구조에 공통점이 많기 때문에 이러한 공통 부분을 프레임워크로 만들고 각 기종 별 특징적인 부분만을 추가로 개발하여 조합하는 방법을 사용함으로써 개발기간을 단축시키고 임무지원 체계의 품질도 향상시킬 수 있다. 따

라서 본 연구에서는 위와 같은 문제점을 극복할 수 있는 방안으로 한국형 비행 임무지원체계 개발을 위한 프레임워크를 제안한다.

2. 관련 연구

2.1 PFPS (Portable Flight Planning System)

PFPS[2]는 F-16 뿐만 아니라 범용적인 사용 목적으로 개발된 프로그램으로써, 기본 플랫폼에 비행임무에 필요한 필수 모듈들이 구현 및 연동되어 있으며, 확장성을 위한 개발 툴킷을 제공하고 있어 사용자의 요구에 따라 체계를 확장시킬 수 있도록 설계된 체계이다. 그러나 체계 연동의 구현 범위가 제한적이며, 기본 구조에 제한적으로 추가, 확장시킬 수만 있는 구조로써, 기초적인 기능만 추가가 가능하다는 단점이 있다. 또한 작성된 추가 확장 모듈은 프로그램 내부를 직접적으로 제어하는 것이 아니라 인터페이스를 통해 데이터 값만 전달되는 체계로 동작되기 때문에, 작성된 모듈의 재사용이 거의 불가능하며, 재사용이 가능하다 하더라도 그 응용 범위는 매우 제한적이다.

2.2 JMPS (Joint Mission Planning System)

JMPS[2] 구축 프로젝트는 기존의 비행임무계획체계들을 교체하고, 항공기, 무기, 센서들의 임무 수행 원에 대해 잘 구조화된 비행 계획 도구를 제공할 목적으로 설계된 비행임무지원체계이다. JMPS 프로그램은 프레임워크, 공통 소프트웨어 컴포넌트, UPC(Unique Planning Component)로 구성되어있다. 이렇게 정의된 JMPS 구조는 각종 플랫폼과 무기체계에서 공통의 컴포넌트들을 사용하도록 하여 소프트웨어의 중복개발을 방지하고 상호 운용성을 증진시키도록 설계되어

이다. 프레임워크는 윈도우 기반으로 작동하는 기본 소프트웨어이며, 공통 소프트웨어 컴포넌트는 특정 기종과 상관 없는 일반적인 비행경로 계획 및 3D 가상 비행, 그리고 전자지도, 공역, 장애물 및 항법시설 등의 비행 정보를 시현/관리한다. UPC는 항공기별 고유 형상 소프트웨어로 비행 특성, 특수무장, 데이터 링크 등의 자료를 이용한 임무계획을 지원한다. 비록 JMPS가 다른 체계들과의 상호운용성을 기본 목적으로 구현되었으나, 현재 정보 DB, 작전 DB 등의 한국군에서는 운용하고 있는 체계들과 연동하여 운용되고 있지 못하며, 국내 도입된 타 항공기용 비행임무계획 체계들과도 상호 운용이 불가능 하다.

2.3 T-50 MPS(Mission Planning System)

T-50는 국내에서 개발된 고동훈련 항공기로서 T-50 MPS는 한국국방연구원이 2002년부터 단독 개발한 국내 유일의 비행 임무계획체계이다. 현재 공군에서 운용 중인 정보 DB와 연동하여 경로 중에 미사일 등의 위협 요소의 위치와 비행 참조자료를 제공함으로써, 최적의 비행경로를 선정하고, 항공기 특성 데이터를 이용하여 비행시간, 위험회피 등을 계획할 수 있는 장점이 있으나, 단일기종 플랫폼 구조상 확장성이 부족하고 개발된 모듈의 재사용성을 보장하지 못한다는 단점 또한 상존하고 있다.

따라서 국내의 실정에 맞는 비행임무지원체계의 프레임워크 기반의 개발 및 운용을 통해 실질적인 항공 소프트웨어 개발의 생산성을 높이고 재사용을 통한 비용 절감을 구현함으로써 구조화되고 효율적인 한국형 비행임무지원체계의 프레임워크의 필요성은 절실하다.

3. 프레임워크 기반 개발 프로세스

3.1 프레임워크의 개념 및 특징

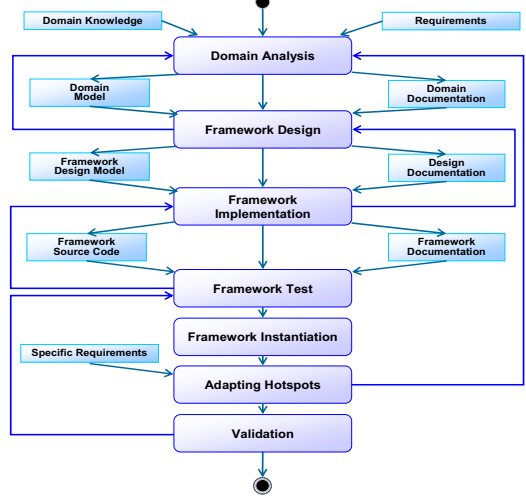
소프트웨어 자산을 재사용하여 소프트웨어 개발의 생산성을 향상시키기 위해, 클래스는 상속을 통하여 기존에 정의되어 있던 클래스의 모든 속성과 메소드를 재사용할 수 있다. 그러나 재사용 대상의 크기가 작으면 재사용의 횟수는 증가하지만 재사용을 통한 비용절감의 효과가 작으며, 반대로 그 크기가 크면 재사용을 통한 비용 절감 효과가 크지만, 재사용의 가능성이 매우 제한적인 문제점을 갖고 있다. 따라서 이를 해결할 수 있는 방안으로 프레임워크가 소개되었으며, 서로 연관된 문제들을 해결하기 위한 추상적인 디자인이 포함된 프로그램들로 정의하고 있다[3].

프레임 워크의 주요 특징들로는 모듈성(Modularity), 재사용성(Reusability), 확장성(Extensibility)등이 있다 [4]. 프레임워크의 모듈성은 상세한 구현의 내용을 캡슐화 하여 디자인과 구현을 변경할 때 그 영향을 국지화 시켜 소프트웨어의 품질을 향상시키는 장점이 있으며, 프레임워크는 반복적으로 적용 가능한 일반적인 컴포넌트를 정의함으로써 재사용성을 높인다. 또한 프레임워크는 개별 프로그램마다 바뀔 수 있는 부분(Hot-spot)에 훅 메소드(Hook methods)를 제공함으로써 확장성을 제공한다[4]. 훅 메소드는 추상적인 메

소드로 프레임워크를 확장할 때는 이 메소드의 내용을 완전하게 구현하여 특정 프로그램의 기능을 구현한다.

3.2 프레임워크 개발 프로세스 정의

대표적인 프레임 워크 개발 프로세스로는 Hot-Spot-Driven Development[5,6]가 사용되고 있다. 이는 동일한 도메인에서 많이 사용되고 있는 프로그램들을 분석하여 공통요소들을 식별하여 프레임워크화 하고 프로그램마다 변화가 있는 부분(Hot-Spot)을 식별해 나가는 작업을 반복적으로 적용하여 개발하는 방식이다. 그러나 한국형 비행임무계획체계를 개발하기 위한 경우에는 기존에 개발된 프로그램들은 프레임워크 대상으로 분석하기에는 단순하며, 현 체계에 요구되는 요소들을 적용하는데 한계가 있다. 또한 한국형 비행 임무계획체계 개발에 필요한 군사정보, 무장, 정보통신, 기상 등 여러 자료 공유가 필수적인데, 개발 중 식별되는 기술, 데이터, 소프트웨어 등의 표준화를 설정하여, 향후 재사용성 및 상호운용성이 보장되도록 설계되어야 한다. 따라서 한국형 비행임무계획체계 도메인에 가장 적합한 프레임워크 개발프로세스를 정의하고 적용하기 위해서는 프레임워크개발을 위한 프로세스 정의[7]를 참고하여 한국형 비행임무지원체계를 위한 프레임워크의 개발에 적합한 프로세스를 (그림 1)과 같이 크게 7 가지 단계로 나누었다.



(그림 1) 프레임워크 개발 프로세스

①Domain Analysis에서는 비행 임무지원체계에서 공통적으로 필요한 기능들을 식별(Domain Scoping)하는 단계이다. 일반적으로는 대표적인 객체들을 식별하고 로직을 파악하는 것이 중요하지만, 프레임워크를 개발하는데 있어서는 프로그램으로 개발되었을 때의 중요 기능 블록들을 식별하는 것이 먼저 선행되어야 한다. 이 단계에서 필요한 입력 요소로는 충분한 도메인 지식과 체계 요구사항들이 필요하다. 또한 프레임워크가 개발된 이후, 프레임워크가 진화하는 과

정에서 각 기능 블록의 요구사항을 분석하는 과정이 필요한데 먼저, **Domain Model** 을 통해 체계를 구성하고 있는 기본적인고 공통된 클래스들을 추출하고 클래스들의 관계를 정의하는 작업이 필요하다. 이와 동시에 정의된 각 요소 별 기능을 세부적으로 정리하는 단계가 요구되는데 **Domain Documentation** 을 통해 그 역할을 할당할 수 있다.

②**Framework Design** 에서는 **Domain Analysis** 단계의 산출물을 기반으로 프레임워크에 대한 설계를 진행한다. 각 클래스들의 메소드 중에서 각 기능요구 사항에 따라 변경해야 할 부분(**Hot-Spot**)을 식별하는 것이 중요하다. 특히, 이 **Hot-Spot** 은 프레임워크를 사용하여 실제 비행임무지원 체계를 개발할 때 항공기에 따른 특정 정보 요구 사항(항공기 및 무장 상태 등)을 임무가 부여된 항공기에 맞게 조정해 주어야 하는 부분으로, 국방 표준 체계 연동을 통해 각 항공기 별로 적합한 상태의 확인과 무장체계의 장착 가능성 및 무장 특성 파악, 항공기의 비행 성능 등 임무 부여 요구에 따라 조정될 필요가 있다.

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

④**Framework Test** 단계에서는 프레임워크가 사용자의 요구사항을 만족하는지 그리고 비행임무 지원체에서 요구하는 기능들을 개발할 수 있는지에 대해 시험하며, 이 시험이 완료되면 유효성 검사를 거쳐 최종 산출물인 한국형 비행임무지원체계를 위한 프레임워크가 나오게 된다. 프레임워크 개발 후에 추가적으로 변경을 요하는 경우 먼저 ⑤프레임워크를 실체화(**Instantiation**)하고 요구 기술에 맞는 특수한 요구사항을 적용하여 ⑥**Hot-Spot** 지점의 클래스들을 구체화시키는 과정이 추가될 수도 있다. 요구사항에 맞는 **Hot-Spot** 의 추가를 통해 발생하는 새로운 개발 사항이 발생하는 경우, 다시 1 단계인 **Domain Analysis** 로 되돌아가 개발된 프레임워크를 진화시킴으로써 보다 기능 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 **Framework** 에서 요구하는 특수한 요구사항들에 대해 결함이 없는지, 혹은 유효한 요구사항인지 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 무결성도

를 높일 수 있는 프레임워크 설계 모델을 제시할 수 있다.

4. 한국형 임무지원체계를 위한 프레임워크 기본설계

이 장에서는 한국형 비행 임무지원체계의 도메인을 분석하고 앞 장에서 정의한 프레임워크 개발 프로세스에 따라 프레임워크 설계를 진행하도록 한다.

4.1 Domain Analysis

Domain Analysis 를 진행하기 위해서는 비행업무의 진행 과정을 먼저 이해할 필요성이 있다. 비행을 위한 주요 단계로는 비행계획, 비행준비 및 리허설, 비행 후 디브리핑으로 크게 나눌 수 있다. 이러한 비행 임무지원체계는 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 **Mission** 에 의해 요구되는 데이터들이 결정된다. 따라서 **Mission** 은 임무지원체계 전반에 영향을 주는 주요 요소이며, 진행하는 주체가 된다. 비행임무지원체계는 체계 내의 모든 모델들을 관리하게 되며, 체계의 주요 요소인 **Mission** 의 부여 및 발생하는 이벤트의 우선 순위 결정 등의 처리 권한을 갖게 된다.

임무에 따른 각 모델들로부터 요청 받은 데이터는 임무계획(**Mission Plan**)에게로 전달되어 비행 전반에 대한 항로계획, 위협 분석 및 임무데이터 처리 등의 임무를 결정하게 되며, 임무계획 후 임무리허설 (**Mission Rehearsal**)을 통해 비행의 전반적인 확인 및 분석을 시스템 상에서 미리 진행해 볼 수 있다. 이 데이터를 통해 비행이 이루어진 후에는 임무디브리핑 (**Mission Debriefing**)을 통해 비행 당시 수집된 비행 정보를 바탕으로 화면을 재구성하여 비행 당시의 상황을 재현해 볼 수 있게 된다. 지금까지 설명된 **Domain Analysis** 를 바탕으로 구성된 **Domain Model** 을 표시하면 (그림 2) 과 같다.

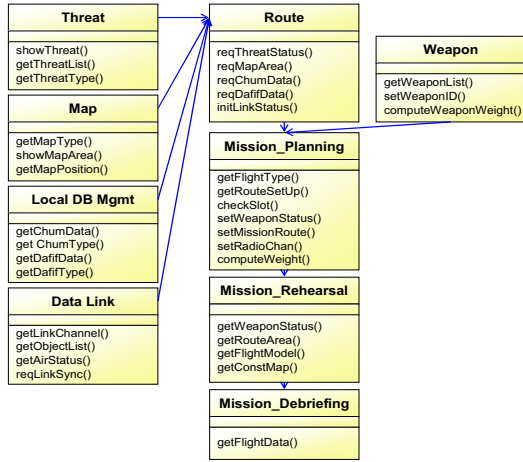


(그림 2) 비행임무지원체계 프레임워크 Domain Model

4.2 Framework Design

Framework Design 단계에서는 **Domain Analysis** 단계의 산출물인 **Domain Model** 과 기본 업무 흐름을 바탕으로 프레임워크를 구성하는 클래스들을 디자인하고 이 클래스들 간의 관계를 정의한다. 이 단계에서 가장 중요한 것은 비행임무지원체계의 ‘변할 수 있는 메소드(**Hot-Spot**)’ 와 ‘변하지 않는 메소드(**Frozen-Spot**)’ 를 구분하는 것으로, 프로그램마다 변할 수 있는 메소드는 **abstract method** 로 정의하여 메소드의 **signature** 만 정의하고 프레임워크를 사용하여 실제 체

제를 개발할 때에는 이 클래스를 상속받아 abstract method 를 구체적으로 구현하면 된다.

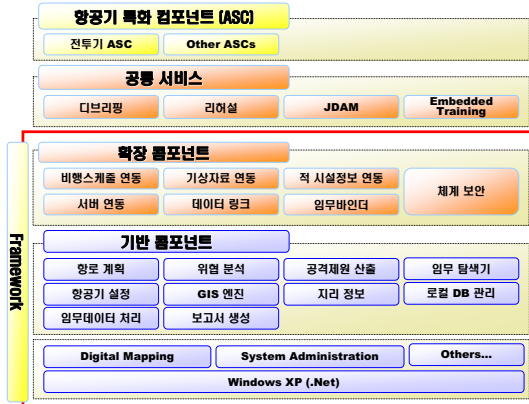


(그림 3) Framework Design 단계의 클래스 다이어그램

(그림 3)의 Mission_Planning 클래스의 경우 항공기 별로 무장을 장착할 수 있는 개수와 슬롯의 위치가 다르므로 항공기의 특성에 따라 각기 다른 설정 상태를 지원해 줄 필요가 있다. getFlightType(), checkSlot() 과 setWeaponStatus() 등의 메소드들은 항공기의 상태, 무장을 장착할 수 있는 슬롯의 상태와 무장의 장착 가능 상태를 확인할 수 있는 메소드들로서, 항공기의 특성에 종속되므로 Hot-Spot 으로 표현해 주어야 한다.

4.3 한국형 비행임무지원체계 프레임워크 설계

지금까지 Framework 개발 프로세스에 따른 Domain Analysis 와 Framework Design 을 통해 국내 현실에 적합한 비행임무 지원체계의 프레임워크 설계 방법을 제시하였으며, 이 결과, (그림 4)과 같은 한국형 비행 임무지원체계 프레임워크의 정의를 도출할 수 있다.



(그림 4) 한국형 비행임무지원체계 프레임워크 정의

기본 프레임워크로는 주요 비행임무지원을 담당하는 범용적 기능을 기반 컴포넌트로 묶어 프레임워크

를 구성하며, 비행에 있어 참조가 될 수 있는 데이터 연동이나 정보 획득과 관련된 부분을 확장 컴포넌트로 묶어 기본 프레임워크를 구성하였다. 또한 항공기에 공통적으로 쓰일 수 있으나 필요에 따라 부수적인 역할을 담당하는 디브리핑이나 리허설 등의 기능은 공통 서비스로 묶어 선택적으로 채택할 수 있도록 구성하였고, 상단의 항공기 특화 컴포넌트는 각기 적용되는 항공기의 특성을 묶은 별도의 모듈로써, 하위 모듈에 특정 값을 전달할 수 있도록 별도 설계하였다. 이는 기존 체계(예, 미군의 JMPS)의 기본 모델을 바탕으로 국내 실정에 맞도록 각 컴포넌트를 재 구성하고, 주요 유사 기능을 가지고 있는 컴포넌트를 묶어 프레임워크로 재지정함으로써, 추후 발생될 수 있는 컴포넌트의 개발 및 수정, 그리고 재사용에 따른 효율성을 높일 수 있도록 설계하였다.

5. 결론

본 논문에서는 국내에서 운용하는 차기 항공기를 지원하는 비행 임무 지원체계를 개발하는데 있어서, 재 사용이 가능하며 품질을 향상시키고, 국내 환경에 적합한 체계를 보장해 줄 수 있는 한국형 비행 임무 지원체계의 프레임워크를 제안하였으며, 이에 대한 개발 및 설계 방향을 제시하였다.

향후 연구 방향으로는 제안된 프레임워크의 Model Checking 방법 등을 이용한 각 모듈 별 검증 및 효율적인 유효성 검토 방법 등에 대한 연구가 이루어질 예정이다.

Acknowledgement

본 연구는 정보통신부 및 정보통신연구진흥원의 IT 신성장동력핵심기술개발 사업의 일환으로 수행하였음(ITAA1100080200460001000500200, 항공기 임베디드 시스템 개발)

참고문헌

- [1] 정구돈, 김규석, “비행임무계획체계 발전동향 및 과제”, 주간국방논단 제 1162 호, p1-20, 2007.
- [2] William R. Clemons, “The Case for Consolidating Tactical and Operational Systems”, The Land Warfare Papers, No.52, September 2005.
- [3] Ralph E. Johnson, Brian Foote, "Designing Reusable Classes," Journal of Object-Oriented Programming, vol. 1, pp. 22-35, June/July 1988.
- [4] Mohamed Fayad, Douglas C. Schmidt, "Object-Oriented Application Frameworks," Communications of the ACM, vol. 40, No 10, October 1997.
- [5] Wolfgang Pree, “Design Patterns for Object-Oriented Software Development,” Addison-Wesley, 1994.
- [6] Hafedh Mili, Ali Mili, Sherif Yacoub, Edward Addy, “Reuse Based Software Engineering,” John Wiley & Sons, Inc., 2002.
- [7] M. Morisio, D. Romano, I. Stamelos, “Quality, Productivity, and Learning in Framework-Based Development: An Exploratory Case Study,”IEEE Transactions on Software Engineering, vol. 28, No. 9, September 2002.