

다양한 발사체계모의 소프트웨어 개발을 위한 프레임워크의 제안

정아정*, 정영환*, 이원식*, 위성혁*, 이승영*

*LIGN엑스원

e-mail:ajeong.jeong@lignex1.com

A Proposal on Application Framework for Developing Various Launcher System Simulators

A-Jeong Jeong*, Young-Hwan Jeong*, Won-Sik Lee, Sung-Hyuk Wi*,

Seung-Young Lee*

*LIGNex1

요 약

다양한 유도무기체계의 개발로 이를 시험하거나 훈련하기 위한 발사체계모의 소프트웨어가 반복되어 개발되고 있다. 이러한 소프트웨어는 개발비는 적으나 높은 신뢰성을 요구한다. 따라서 소프트웨어의 품질 향상과 개발 및 유지보수 비용 절감을 위해서는 소프트웨어 재사용이 필요하다. 본 논문에서는 다양한 접근을 통해 발사체계모의 소프트웨어 개발지원을 위한 프레임워크의 개발방안에 대해 제안한다. 여러 체계마다 공통으로 사용될 수 있는 기능을 추출하고 이들의 집합을 프레임워크로 정의한다. 본 프레임워크를 발사체계 시뮬레이터 개발에 이용 시 개발 및 유지보수 비용절감 및 소프트웨어 품질 향상을 기대할 수 있다.

1. 서론

오늘날 다양한 전장상황에 대응하기 위해 유도무기체계가 개발되고 있다. 유도무기체계 개발에는 다양한 장비와 이를 시험하기 위한 시험장비 및 훈련장비 등이 포함되며, 그 중에서 발사체계 모의 소프트웨어는 필수적으로 개발된다. 발사체계 모의 소프트웨어의 경우, 개발 인원과 개발비가 작기 때문에 기능이 유사한 부분이 있음에도 불구하고 매년 새로이 개발되어 왔다. 이 후로도 개발인원이나 개발비가 늘어나는 것은 현실적으로 힘들기 때문에, 적은 비용으로 신뢰성 높은 소프트웨어를 제공하기 위해 프레임워크를 제안한다. 발사체계 모의 소프트웨어는 체계마다 상이한 부분도 많이 있으나 공통화 할 수 있는 부분 또한 존재한다. 우리는 공통화 할 수 있는 부분을 추출하여 프레임워크 형태로 만들어 추후 발사체계 모의 소프트웨어 개발 시에 이용하고자 한다.

본 논문에서는 프레임워크 개발방법을 도입하여 발사체계 모의 소프트웨어의 재사용을 통해 개발 편의를 도모하고, 검증된 프레임워크 사용으로 품질 향상을 하고자

발사체계 프레임워크를 제안한다. 관련 연구에서는 발사체계 프레임워크를 개발하기 위해 프레임워크 개발 방법을 소개하고, 본문에서는 발사체계 프레임워크의 제공 기능 분석하며, 결론으로 마무리 짓는다.

2. 관련연구

2.1 어플리케이션 프레임워크 [1]

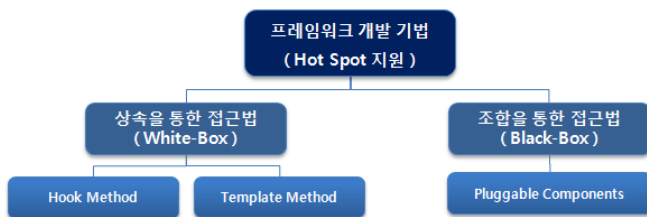
어플리케이션 프레임워크는 응용소프트웨어를 개발할 때 도움을 주는 재사용 가능한 디자인과 코드의 집합이다. 어플리케이션 프레임워크는 Common Spot 과 Hot Spot 으로 나눌 수 있다. Common Spot은 모든 응용 프로그램에서 공통으로 이용할 수 있는 부분을 개발하기 위한 방법을 의미하고, Hot spot은 어플리케이션 특성에 따라 바뀔 수 있는 부분을 개발하기 위한 방법을 의미한다. Hot Spot을 개발하기 위한 접근방법으로는 크게 상속을 통한 접근법(White-Box)과 조합을 통한 접근법(Black-Box)으로 나눌 수 있다(그림 1). 각 접근 방법은 다음과 같다.

- White-Box > Hook Method : 추상 클래스를 상속하고 부모클래스의 추상 메소드를 재정의하는 방법.

- White-Box > Template Method : 알고리즘 뼈대를 정의하는 방법으로 프로세스 흐름과 통합된 로직을 제공.

- Block-Box > Pluggable Component : 인터페이스 구현해 놓고 사용자는 구체 클래스를 정의하여 조합하여 사용하는 방법.

여기서 White-Box와 Block-Box를 모두 이용해 개발한



(그림 1) 프레임워크 개발 기법

프레임워크를 Gray-Box 프레임워크라 한다. 최대한 기존에 개발된 내용을 재사용하며, 프레임워크 개발 시간을 줄이기 위해 발사체계 프레임워크는 Gray-Box를 지향한다.

3. 본론

3.1 발사체계 프레임워크 제공 기능 분석

기존에 개발되어 있는 발사체계 모의 소프트웨어를 참고하여 발사체계 프레임워크에서 제공할 기능을 분석한 결과는 다음과 같다.

① 원격통제명령 처리

- 운용통제기로부터 원격 전원 및 S/W 인가/해제 명령 수신 시 전원 및 S/W를 인가/해제할 수 있어야 한다.

② 장비 상태 점검

- 모의 장비 상태를 점검 하여 운용통제기로 송신할 수 있어야 한다.

③ 시나리오 정보 수신 및 응답

- 운용통제기로부터 시나리오 정보를 수신하고 설정할 수 있어야 한다.

- 시나리오 배포 완료 후 시나리오 설정 응답 송신 기능을 제공해야 한다.

④ 시작 / 종료 수신 및 응답

- 운용통제기로부터 시험 시작과 종료 명령을 수신하고, 그에 대한 응답을 송신할 수 있어야 한다.

- 수신한 시작/종료 명령에 따라 발사 체계 모의를 시작/종료 할 수 있어야 한다.

⑤ 발사체계 기능 모의

- 발사체계의 상태를 모의 할 수 있어야 한다.
- 모의 정보를 운용통제기로 송신할 수 있어야 한다.

⑥ 발사체계 연동 모의

- 운용통제기와의 인터페이스를 제공하여야 한다.
- 외부 체계 인터페이스를 제공하여야 한다.

여기서 ①, ②, ③, ④는 다른 모의 소프트웨어에서도 사용 가능한 Common Spot 이므로 본 논문에서는 다루지 않는다. 본 논문에서 개발을 제안하고자 하는 프레임워크는 Hot Spot 인 발사 체계 모의 부분으로 이와 관련된 ⑤, ⑥에 대해 논한다.

3.2 발사체계 기능 및 연동 모의 세분화

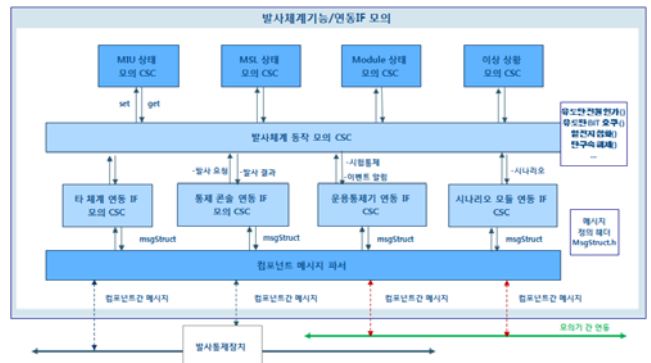
- Block-Box : 3.1의 ⑤, ⑥ 기능을 풀어쓴 내용은 표 1과 같다. 세분화된 기능은 서로 분리되어 독립적으로 동작하는 모듈 형태로 구현된다. 이 기능을 적용한 발사체계 프레임워크 내부 흐름도는 그림 2와 같다. 외부로부터 받은 메시지를 공통으로 파싱할 수 있는 모듈을 두어 메시지 파싱 기능을 모듈화 하여 재사용할 수 있도록 구현한다. 또한 체계 연동 인터페이스는 매 체계마다 달라질 수 밖에 없으므로 분리하고, 본 프로그램을 제어하는 운용통제기와 시나리오 모듈과는 주고받는 메시지가 거의 일정하기 때문에 모듈형태로 구현한다.

- White-Box : 발사체계 동작모의 모듈의 경우는 주요 기능인 전원인가/BIT요구/열전지 점화/탄구속 해제 등에

대해서는 인터페이스 형태로 제공하되 매번 달라질 수 있으므로 상속을 이용하는 형태로 구현한다. MIU/MSL/Module/이상상황 모의의 경우도 마찬가지로 인터페이스를 제공하되 매 체계마다 달라질 수 있으므로 상속을 이용하는 형태로 구현한다.

<표 1> 발사체계 기능 세분화

기능	기능 세분화	동작 내용
발사체계 기능모의	유도탄 연동기 모의	유도탄연동기 상태 모의 유도탄연동기 동작 모의
	장입유도탄 모의	장입유도탄 상태 모의 장입유도탄 동작 모의
	모듈제어함 모의	모듈제어함 상태 모의 모듈제어함 동작 모의
	이상상황 모의 발사동작모의	발사절차 이상상황모의 발사체계상태 이상상황모의 발사동작모의
발사체계 연동모의	통제콘솔 연동모의	메시지 수신 처리 메시지 송신 처리
	운용통제기 연동모의	메시지 수신 처리 메시지 송신 처리
	시나리오모듈 연동모의	메시지 수신 처리 메시지 송신 처리
	타 체계 연동모의	메시지 수신 처리 메시지 송신 처리



(그림 2) 발사체계 프레임워크 내부 흐름도

4. 결론

본 논문에서는 Hot Spot 접근 방식을 이용한 발사체계 프레임워크 개발에 대해 제안하고, 제공해야 할 기능을 연구하였다. 기대 결과로는 개발 시간을 단축하고 유지보수, 장비 규격화에 필요한 문서 작업 및 정적/동적 분석에 드는 비용을 줄일 수 있다. 추후에는 제안한 프레임워크를 구현하여 앞으로 개발되는 발사체계 모의 소프트웨어 개발에 적용할 수 있다.

참고문헌

[1] Xin Chen, "Developing Application Frameworks in .NET", Apress, 2004