

A Study on the Model Driven Development of the Efficient Combat System Software Using UML

Seung-Mo Jung*, Young-Ju Lee**

Abstract

In this paper, we propose a Model Driven Development using UML(Unified Modeling Language) in an efficient Combat System Software development methods. UML is managed by the OMG (Object Management Group) as a Unified Modeling Language. Recently, In developing the software for the system, development time is contrary to a short, while it must meet a number of requirements of our customers. If you develop a non-efficient software early, the structure of the software become more complex in proportion to the number of requirements. As a result, a serious problem in the system, such as an increase in defective products due to the lowering of the reliability and communication problems between the developer has occurred. To overcome those problems, the aim of this paper is to develop a reusable high efficient Combat System Software by applying Model-Centric Development (Not Code-Centric Development). If Combat System Software development method using the UML proposed in this paper is used, easy communication among developers can help reduce the serious errors in systems. Also, it has the advantage that the future needs of software maintenance/repair become easy by using high readability object Model.

▶ Keyword : UML, MDD, MDA, Modeling, Combat System Software

1. Introduction

초기 소프트웨어는 간단한 기능만 탑재하고 고객들의 요구 사항도 많지 않아 간단한 구조를 가지고 있었다. 이러한 경우, 이미 구축된 시스템에 새로운 요구사항들을 추가하기 위한 코드 수정은 어렵지 않아 시스템의 심각한 문제점들은 발생하지 않았다. 또한 개발기간도 길어서 수많은 시험을 통해 오류가 없는 소프트웨어를 개발할 수 있었다. 하지만 최근 한 시스템의 소프트웨어를 개발함에 있어 고객들의 수많은 요구사항들을 만족시켜야 한다. 이에 반해 개발주기는 예전과 비교하여 더욱 짧아졌다. 짧은 개발주기에 수많은 요구사항을 만족시키기 위해 소프트웨어의 구조는 더욱 복잡해졌다. 만약 코딩중심의 개발

방식에서 소프트웨어의 구조가 복잡해지면 개발자들 간의 의사소통 문제와 신뢰성 저하로 인한 불량품의 증가와 같은 문제점들을 발생시킨다. 최근 전투체계 소프트웨어 개발함에 있어서도 이와 같은 문제점에 직면하였다. 예전의 직렬적인 개발방법에서 점차 병렬적으로 소프트웨어를 개발하고 있다. 즉 하나의 전투체계를 만들고 다음 사업이 순차적으로 진행되는 것이 아니라 두, 세 개의 사업이 병렬적으로 진행되고 있다. 반면에 개발기간은 점점 더 짧아지고 있다. 이러한 문제점들을 해결하기 위해 최근 효율적이고 안정성이 있는 소프트웨어를 개발하기 위한 연구가

-
- First Author: Seung-Mo Jung, Corresponding Author: Seung-Mo Jung
 - *Seung-Mo Jung(seungmo1025@naver.com), SW Team(Naval), Hanwha Systems Co.
 - **Young-Ju Lee(doodug@daum.net), Induction electrons 3 Team, Defense Agency for Technology and Quality.
 - Received: 2016. 08. 09, Revised: 2016. 09. 07, Accepted: 2016. 10. 01.
 - This paper will expand the paper ("A Study on the Development of the Efficient Combat System Software Using UML") presented at the 54th Summer Conference Korea Society of Computer and Information 2016.

활발히 진행되고 있다. 초기 개발부터 소프트웨어를 효율적으로 개발하여 재사용성이 높은 효율적인 소프트웨어를 개발하기 위해 많은 연구를 하고 있다. 현재까지 연구된 효율적인 소프트웨어 개발 방법에는 여러 방법들이 있겠지만 그 중, 그래픽 객체들을 사용하여 가독성을 높인 UML(Unified Modeling Language)을 이용한 모델 기반 개발 방법이 주목 받고 있다 [1][2][3].

모델 기반 개발 방법이란 표준화된 모델 중심의 개발 방식으로 작성된 모델로부터 프로그램 코드뿐만 아니라 다양한 문서를 자동으로 생성하는 개발방식이다. 소프트웨어 개발에 모델 기반 개발 방법을 적용하게 되면 개발자들이 쉽게 인지할 수 있는 UML 객체들을 이용하여 제품 개발을 하기 때문에 시스템의 이해와 같은 의사소통을 원활히 하게 해준다. 이로 인해 소프트웨어 설계 및 구현 시 오류를 줄일 수 있고 신뢰성이 높은 제품을 생산할 수 있다. 또한, 특정 컴파일러를 이용한 설계 환경이나 하드웨어와 같은 시스템에 종속적이지 않기 때문에 소프트웨어의 재사용이 가능하다. 그리고 모델을 자동으로 소스 코드로 변환 시켜 주는 툴을 사용하기 때문에 다수의 개발자로부터 생산된 제품의 성능을 일정하게 유지 시킬 수 있다 [4][5].

본 논문에서는 효율적인 전투체계 소프트웨어를 개발하는 방안의 하나로 전투체계 소프트웨어에 모델 기반 개발 방법을 적용하는 개발방안을 제시한다. 코딩 중심의 개발방식이 아닌 모델 중심의 개발방식을 적용하여 재사용성이 높은 효율적인 전투체계 소프트웨어를 개발하는데 그 목적이 있다. 만약 전투체계 소프트웨어에 모델 기반 개발 방법을 적용하게 되면 표준화된 모델로 부터 자동으로 소스코드를 생성시켜 코드의 통일성과 개발자들 간의, 개발자와 고객 간의 의사소통이 명확해져 더욱 신뢰성이 높은 제품을 생성할 수 있다. 이를 위해, 전투체계 소프트웨어의 한 기능을 담당하는 모듈에 모델 기반 개발 방법을 적용하였다. 그리고 설계된 모델로부터 응용코드를 생성시켜 기존 코드와 성능을 비교하면서 이 접근방법의 가능성을 보여준다. 모델 기반 개발 방법을 적용하기 위해 사용한 UML툴로는 자동 코드 생성 능력이 있는 IBM사의 Rational Rhapsody이다. Rational Rhapsody는 수많은 모델링 툴 중에서 자동코드 생성 기능을 비롯한 여러 가지 기능이 가장 뛰어난 모델링 툴 중 하나로 손꼽히고 있다. 자동코드 생성기능을 가진 UML 소프트웨어 툴을 사용한다면 추상적인 수준의 디자인부터 실행까지 원활한 개발과정을 가능하게 해 주어 개발에 더욱 집중할 수 있다.[6][7].

본 논문의 구성은 다음과 같다. 2절에서는 효율적인 전투체계 소프트웨어를 개발하기 위한 개발환경에 대해서 설명하고 3절에서는 효율적인 전투체계 소프트웨어 개발에 모델 기반 개발 방법을 적용하는 개발방법에 대해서 설명하였다. 그리고 4절에서는 성능분석을 통해 기존 코드와 모델링 작업한 코드의 메모리 용량, 처리 속도 비교 및 정성평가 통해 제안한 방법에 대한 가능성을 입증하였다. 마지막 5절에서는 결론 및 추후 연

구과제로 이 논문을 마무리 하였다.

II. Combat System Software Development Environment base on MDD

이 절에서는 효율적인 전투체계 소프트웨어를 개발하는데 사용한 개발 환경에 대해서 설명한다. 모델 기반 개발 방법을 적용하기 위해 UML과 MDA(Model-Driven Architecture)기반의 개발방식에 대해서 간략하게 설명하고 사용한 모델링 툴인 Rational Rhapsody에 대해서 설명하였다. 또한, 소프트웨어 개발에 필요한 디자인패턴과 휘저 모델에 대해 설명하였다.

1. UML과 MDA기반 개발방식

모델 기반 개발 방법은 코드중심의 개발방식과는 달리 모델 중심의 개발방식으로 모델을 통해 프로그래밍 소스코드를 자동으로 생성시켜 준다. 기존의 코드중심의 개발방식은 의사소통의 도구가 프로그래밍 언어이다. 이런 이유로 아무리 쉽게 코딩을 하더라도 프로그램을 개발하는 사람들 간의, 개발자와 고객들 간의 소통은 어려웠다. 반면에 모델중심의 개발방식은 의사소통의 도구가 표준으로 정의되어 있는 모델이기 때문에 개발자들 간의, 개발자와 고객과의 소통의 어려움을 해결할 수 있다. 모델 기반 개발 방법은 MDA를 기반 기술로 한다[8]. MDA는 2001년 OMG(Object Management Group)에서 표준 스펙으로 정의되었고 OMG에서 지속적으로 작업해온 모델링 관련 표준화 작업의 성과를 바탕으로 모델 중심의 시스템 구성을 위한 표준 기술이다[9][10][11]. MDA는 플랫폼 독립적인 모델(PIM:Platform Independent Model)작성, 플랫폼 종속적인 모델(PSM:Platform Specific Model)작성으로 구성된다 [12][13]. 플랫폼 독립적인 모델과 플랫폼 종속적인 모델은 UML을 이용하여 작성하며 UML은 요구분석, 시스템설계, 시스템 구현 등의 시스템 개발 과정에서, 개발자간의 의사소통을 원활하게 이루어지게 하기 위하여 표준화한 모델링 언어이다. 이는 객체지향설계를 위한 표준 언어로서 소프트웨어 산출물을 가시화, 명세화, 구축, 문서화하는 데 사용된다.

본 논문에서는 UML의 다이어그램 중 클래스다이어그램, 상태머신다이어그램 등을 사용하였다. 또한, 응용소프트웨어의 플랫폼 독립적인 모델 작성과 플랫폼 종속적인 모델 작성을 위해 휘저와 디자인패턴을 적용하였고 OS(Operating System)의 플랫폼 독립적인 모델 작성과 플랫폼 종속적인 모델 작성은 모델링 툴인 Rational Rhapsody에서 제공하는 프레임워크를 사용하였다.

2. Rational Rhapsody

소프트웨어 개발에 모델 기반 개발 방법의 개념을 적용하기 위해 사용한 모델링 툴은 Rational Rhapsody이다. Rational

Rhapsody는 UML의 장점을 최대한 살릴 수 있도록 고안된 실무형 개발 프로세스로서, UML을 가장 잘 적용할 수 있다고 알려져 있다. 또한, UML 표기법을 시각화 하여 모델로 제공하고 모델에 대한 코드를 자동으로 생성시켜 주어 설계와 구현을 모델링 틀에서 직접 할 수 있는 장점을 가지고 있다. Rational Rhapsody는 OS에 독립적으로 모델을 만들 수 있고 상태머신 다이어그램(State Machine Diagram)으로 객체에 대한 행위를 모델링 할 수 있다. 또한, 애니메이션 기능이 있어서 모델에서 직접 테스트를 할 수 있는 장점을 가진다. Rational Rhapsody에서 자동으로 코드를 생성하기 위해서는 Windows나 VxWorks등과 같은 OS가 있는 시스템인 경우에는 OS 종속적인 모델인 OXF(Object eXecution Framwork)를 사용하여야 한다. RTOS에서 제공된 API을 OS Adaptor Layer에서 wrapping 하여 개발자가 어느 운용체제에서든 동일한 인터페이스로 보이게 만들어 준다. 그리하여 개발자는 운용체제마다 다른 API에 신경을 쓰지 않고 모델에 집중하여 개발 및 구현 할 수 있다. 개발자는 모델자체에 집중할 수 있게 되어 프로그램의 완성도를 높일 수 있고 개발 시간 또한 단축하게 만들어 준다. Fig. 1은 OS가 있는 시스템인 경우의 소프트웨어 구조를 표현한 그림이다.

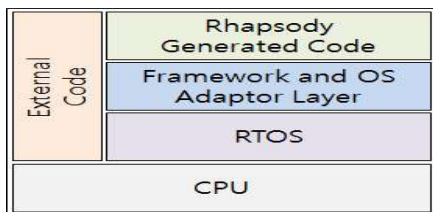


Fig. 1. Software Architecture (With Operating System)

그리고 OS가 없는 시스템인 경우에는 하드웨어 종속적인 모델인 IDF(Interrupt Driven Framework)를 사용한다. IDF는 OS가 없는 작은 시스템의 OS를 대체하는 역할을 한다. Fig. 2는 OS가 없는 시스템인 경우의 소프트웨어 구조를 표현한 그림이다.

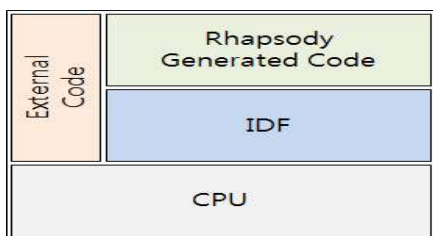


Fig. 2. Software Architecture (Without Operating System)

본 논문에서는 Windows환경에서 Rational Rhapsody의 프레임워크인 OXF를 사용하여 개발하였다. Rational Rhapsody는 각 도메인 특성에 맞게 확장해서 사용할 수 있는 기능을 제공한다. 즉 OMG에 정의된 UML 표기법에 맞게 작성된 틀에서 사용하는 도메인에 맞게 확장해서 사용할 수 있다. Rational

Rhapsody는 기본적으로 Meta Model을 통해서 UML 표기법을 모델로 만들고 Meta-Meta Model을 통해 모델에 대한 코드를 생성시킨다. Fig. 3은 Rational Rhapsody의 기본적인 프로세스를 설명한 그림이다.

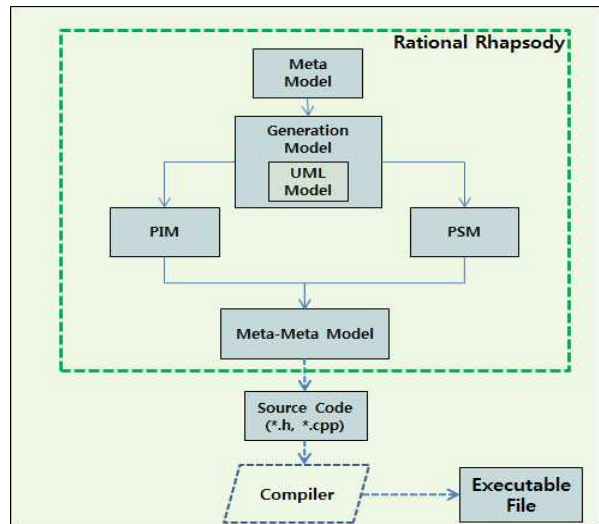


Fig. 3. Rational Rhapsody Process

하지만 전투체계 소프트웨어에 모델 기반 개발 방법을 적용하기 위해 스테레오타입 설정, 전투체계 통신 클래스 모델작성 등 전투체계 도메인에 맞게 확장하여서 사용하였다. 스테레오타입 설정은 전투체계에 적합하게 Rational Rhapsody 틀의 Properties를 설정하였고 통신클래스 모델작성은 전투체계 메시지 정의 틀에서 자동으로 클래스 모델을 생성하여 Rational Rhapsody틀에서 사용할 수 있도록 확장하였다. Fig. 4는 Rational Rhapsody의 확장된 프로세스를 설명한 그림이다.

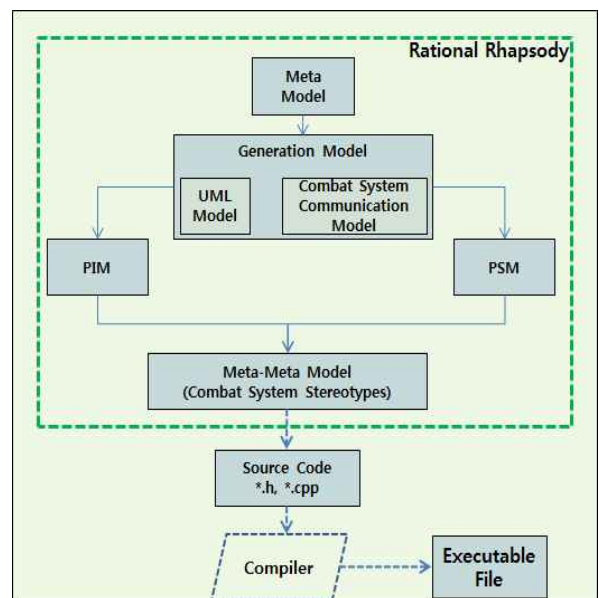


Fig. 4. Rational Rhapsody Extended Process

3. Design Pattern

디자인 패턴은 소프트웨어 설계 시 특정 상황에서 자주 만나는 문제를 해결하기 위해 사용할 수 있는 재사용 가능한 솔루션을 말한다. 디자인 패턴을 소프트웨어 개발에 적용시키면 객체들의 유연성을 높일 뿐만 아니라 가변요소에 대해서 확장을 쉽게 할 수 있어 효율적인 소프트웨어를 개발할 수 있다 [14][15]. 수많은 Design Pattern 방법들이 존재하지만 전투체계 소프트웨어에 특징에 따른 적합한 Design Pattern을 선별하였다. 본 논문에서 적용한 디자인 패턴은 다음과 같다.

- Strategy Pattern : 고객의 요구사항에 따라 기능의 변화가 빈번하므로 이를 해결하기 위해 사용.
- Adapter Pattern : 전투체계 소프트웨어에서 사용하는 통신 방법의 변화가 있으므로 이를 해결하기 위해 사용.
- Template Method Pattern : 전투체계의 장비 상태정보를 체크하는 알고리즘의 흐름은 비슷하지만 상태정보가 가변적일 경우 이를 해결하기 위해 사용.
- Factory Pattern : 수많은 클라이언트 클래스가 생길 수 있어 클래스들을 관리하기 위해 사용.

4. Feature Model

휘처란 Domain Language로 사용자나 개발자가 식별할 수 있는 시스템의 구분이 되는 특징을 의미한다. 이러한 휘처를 통해 나온 휘처 모델은 사용자나 개발자의 의사소통 매체로 사용할 뿐만 아니라 공통점과 가변요소를 식별하는데 많이 사용하는 기법이다[16].

본 논문에서 휘처 모델을 사용한 것은 전투체계 시스템의 한 모듈인 기록 및 재생 모듈에서 이전 사업들을 비교하여 공통점과 가변요소를 좀 더 효율적으로 도출하기 위해 사용하였다.

III. Combat System Software Development Method base on MDD

이 절에서는 효율적인 전투체계 소프트웨어를 개발하는 방법에 대해서 설명한다. 개발방법을 설명하기 위해 적용한 모듈은 기록 및 재생 모듈이다. 전투체계 소프트웨어는 수많은 모듈로 구성되며 모듈간의 서로 획득한 정보들을 주고받으며 메시지 기반으로 기능을 수행한다. 기록 및 재생 모듈은 전투체계 소프트웨어의 한 기능을 담당하는 모듈이다. 기록 및 재생 모듈의 주요기능은 전투체계의 무장/센서에서 획득한 레이더비디오 정보, TV비디오정보, 전술정보 등을 원하는 시점에 기록하여 같은 상황을 보기위해 재생하는 기능을 수행한다. 개발방법에는 요구분석, 설계, 구현, 시험의 4단계로 구성된다.

- 1단계 : 요구분석 단계에서는 기존 전투체계의 사업별 기록 및 재생 모듈들의 요구사항들을 휘처 단위로 분석하여 공통

점과 가변요소를 식별하였다.

- 2단계 : 설계 단계에서는 전투체계에 적합한 디자인 패턴과 식별된 휘처를 사용하여 공통점과 가변요소에 대한 클래스 다이어그램을 설계하였다.
- 3단계 : 구현 단계에서는 클래스 다이어그램의 행위를 스테이트 머신 다이어그램으로 작성하였고 작성된 모델로부터 코드를 생성하였다. 그리고 컴파일러와 연동하여 실행파일을 생성하였다.
- 4단계 : 시험단계에서는 모델링 툴의 애니메이션 기능을 사용하여 모델 시험 및 기능 시험을 하였다.

Table 1은 개발방법의 절차를 간략하게 정리한 표이다.

Table 1. Development Process of Combat System Software

Process	Discription
Requirement Analysis	Feature Model Design of Combat System Software
	Commonalities and Variable Factor of Combat System Software
Design	Class Diagrams Design of Combat System Software
	Application of Design Patterns
Implementation	Creating a State Machine Diagrams of Combat System Software
	Code Generation
Test	Model Test of Combat System Software
	Functional Test of Combat System Software

1. Step1 : Requirement Analysis

요구사항 분석 단계에서는 휘처모델을 통해 공통점과 가변요소를 식별하였다. 기존 개발되었던 시스템들의 요구사항을 분석하여 휘처들을 식별하여 추후 추가되는 부분에 대해서 대비하였다. 이를 통해 새로운 시스템 개발할 때 수정을 최소화하는 재사용성이 높은 소프트웨어를 개발하는 것이 그 목적이다. 휘처 모델에서 Mandatory 휘처는 시스템에서 필수적으로 구현해야 하는 부분이고 Optional 휘처는 시스템에 따라서 선택되거나 선택되지 않는 휘처로 표현한다. Alternative 휘처는 여러 휘처 가운데 하나만 선택되는 것을 의미한다. 기록 및 재생 모듈에서는 Radar Source, TV Source, Domain, TacInfoList는 시스템에 따라서 선택되거나 선택되지 않아도 되서 Optional 휘처로 표시하였고 Combat System Communication Method, Standard Communication Protocol, Operating System은 여러 휘처 중 하나만 선택되므로 Alternative 휘처로 표시하였다. 그리고 꼭 필요한 기능들은 Mandatory 휘처로 표현하였다. Fig. 5는 기존 여러 사업에서 개발되었던 기록 및 재생 기능을 휘처 단위로 분석하고 휘처 모델로 표현한 그림이다.

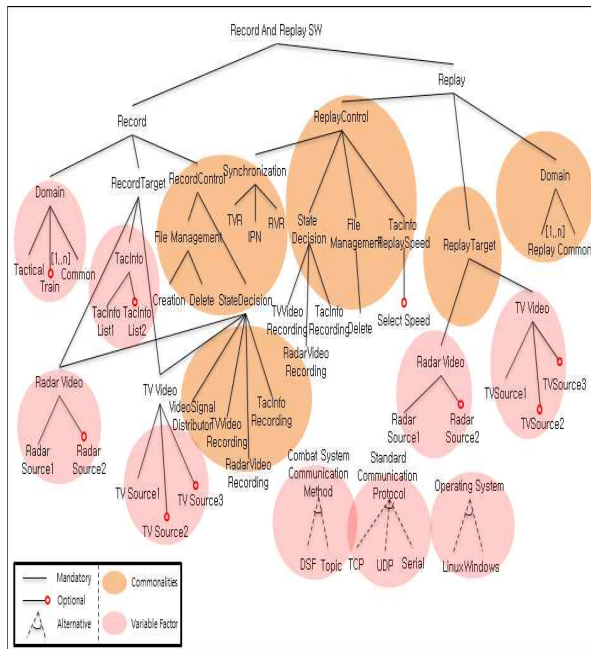


Fig. 5. Feature Model of Combat System Software

회처 모델을 기반으로 공통점과 가변요소를 구분하였다. 회처 모델에서 Mandatory 회처는 필수적으로 구현해야 하는 공통적인 부분이고 Optional 회처와 Alternative 회처는 가변적 요소를 가지고 있는 부분이다. Table 2는 회처 모델을 참고하여 기록 및 재생 모듈의 공통점과 가변요소를 정리한 표이다.

Table 2. Commonalities and Variable Factor of Combat System Software

Classification	Identification
Commonalities	Record Control Algorithm
	Replay Control Algorithm
	Record File Management
	Status Information Management
Variable Factor	TV Video Source Selection
	Radar Video Source Selection
	Tactical Information Selection
	Distribution Status Information
	Communication Method

2. Step2 : Design

설계 단계에서는 회처 모델과 디자인패턴을 사용하여 클래스 다이어그램 설계해 주었다. 회처 모델은 기능적(서비스) 및 비 기능적 특성 등을 표현하는데 비해, 클래스 모델은 개발 단계 수준 상 상위 수준에 해당한다[17]. 따라서 회처 모델의 서비스부분을 기반으로 클래스 모델들을 도출하였다. 이를 기반으로 클래스들의 상호 작용을 나타내는 다이어그램인 클래스

다이어그램을 설계했다. 하지만 모델링 툴을 사용하여 클래스 다이어그램을 설계했다고 해서 모델 기반 개발 방법을 적용한 것은 아니다. 플랫폼 독립적인 모델과 플랫폼 종속적인 모델을 구분하여 클래스 다이어그램을 설계해 주어야 한다. 이를 해결하기 위해 전투체계에 적합한 디자인 패턴과 공통점 및 가변요소를 바탕으로 플랫폼 독립적인 모델과 플랫폼 종속적인 모델을 작성하였다. 이렇게 설계 할 경우 추후 확장되는 기능에 대해 수정요소를 줄일 수 있고 수정요소만 시험하면 되기 때문에 좀 더 신뢰성을 가진 안정적인 소프트웨어가 될 수 있다.

본 논문에서는 기록 및 재생 모듈의 클래스 다이어그램 설계를 다음과 같이 해 주었다.

- 전투체계 통신클래스 : Adapter Pattern을 적용하였다. 통신방식이 변경할 경우 수정을 최소화 하도록 설계하였다. 전투체계 소프트웨어에서는 메시지 기반으로 소프트웨어가 수행되기 때문에 통신클래스가 가장 중요하다. 통신클래스의 가장 큰 역할은 메시지를 수신 받아서 처리하고 그 결과를 메시지로 송신한다. 만약, 메시지가 변화하게 되면 메시지 정의부터 처리에 영향을 미치기 때문에 가장 중요한 부분이다. 이런 이유로 Adapter Pattern을 적용하여 통신변화에 따른 변화를 최소화 하였다. 통신변화에 따른 API가 변경될 경우 Adapter클래스만 변경하면 다른 클래스들은 수정이 안하도록 하였다. 이로 인해 개발자가 수동으로 수정하여 발생하는 오류를 최소화 할 수 있다.
- 장비제어클래스 : Template Method Pattern을 적용하였다. 장비제어라는 큰 흐름은 상위 클래스에서 정의하고, 서브 클래스의 변화는 부분만 정의하였다. 이를 통해, 복사코드의 사용을 최소화하였고, 현재 관리하고 있는 레이더비디오 녹화기, TV비디오 녹화기, 진술정보 외에 추가되는 장비에 대해서도 수정이 용이하도록 설계하였다. 만약 추가되는 장비가 있다면 서브 클래스를 확장해서 수정하면 된다.
- 표준통신클래스 : Strategy Pattern을 적용하였다. 기존코드에서는 사용하는 표준통신클래스를 바로 정의해서 사용했지만 이 부분에 대해서 추가요구사항에 대비하여 표준통신방식을 분리하여 캡슐화 하였다. 이를 통해, 각 장비에서 필요한 표준통신방식을 쉽게 사용할 수 있고, 추후 표준통신방식변경에 대한 코드 수정을 최소화 되도록 설계하였다.
- 객체관리클래스 : Factory Pattern을 적용하였다. 클래스들의 객체 생성을 담당하는 클래스를 만들어 주어 각 클래스에서 객체를 생성하지 못하도록 설계를 하였다.

Fig. 6은 기록 및 재생 모듈의 클래스 다이어그램을 설계한 그림이고 Table 3은 클래스 다이어그램 중 주요 클래스 기능에 대해 설명한 표이다.

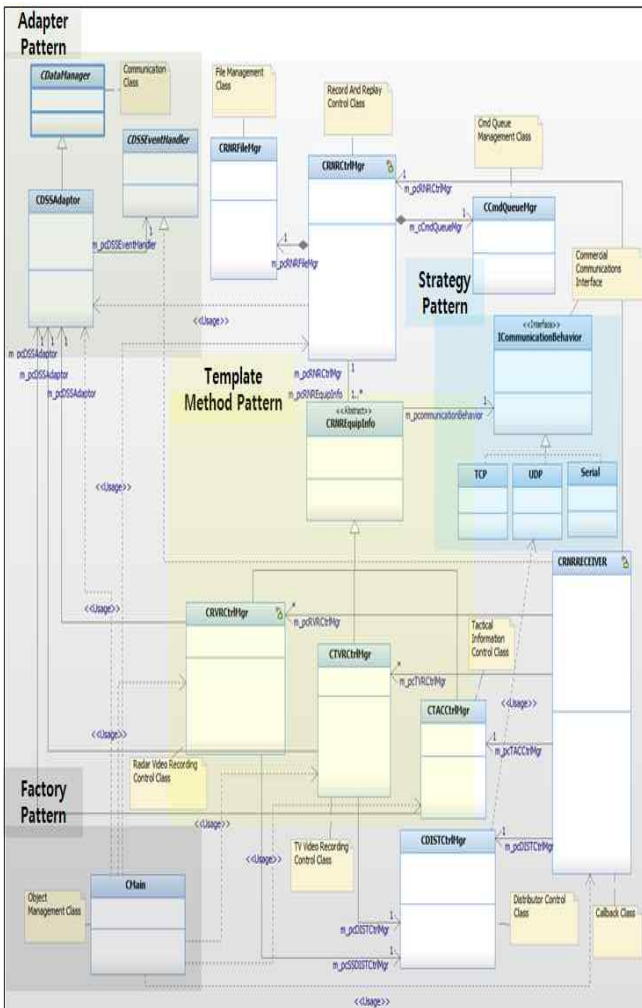


Fig. 6. Class Diagrams Design of Combat System Software

Table 3. Description of Class Diagram

Class Name	Class Description
CRNRCtrlMgr	Record And Replay Control Class
CRNREquipInfo	Equipment Management Abstract Class
CRVRCtrlMgr	Radar Video Recording Control Class
CTVRCtrlMgr	TV Video Recording Control Class
CTACCtrlMgr	Tactical Information Control Class
CDISTCtrlMgr	Distributor Control Class
ICommunicaton Behavior	Communications Interface
CMain	Object Management Class
CRNRFileMgr	File Management Class

3. Step3 : Implementation

구현단계에서는 주요 클래스의 상태머신 다이어그램(State Machine Diagrams)을 작성하였다. 상태머신 다이어그램은 클래스내의 Operation(함수)에 대한 실제 동작을 구현한 다이어그램이다. Event의 발생에 따라서 각 State가 천이되고 State에 따라서 Action을 실행한다. 상태머신 다이어그램은 클래스가 어떻게 동작하는지에 대해서 설계해주면 된다. 한 시스템을 모델로 설계 할 경우에는 클래스 다이어그램으로만 가지고 설계

하는 것은 한계가 있다. 클래스 다이어그램에서는 행위를 모델화 할 수 없기 때문이다. 보통 모델링 툴로 시스템을 모델링할 경우, 클래스 다이어그램에서 끝나는 경우가 많다. 그리고 각 클래스의 행위는 클래스 다이어그램의 Implementation 기능을 사용하여 표현한다. 왜냐하면 상태머신 다이어그램으로 구현할 경우 각 State를 식별하기에는 많은 시간과 노력이 필요하기 때문이다. 하지만 상태머신 다이어그램의 모델로 구현하지 못하면 모델링 툴의 강력한 기능인 애니메이션 기능을 사용하지 못한다. 애니메이션 기능은 모델을 검증하는 기능으로 각 State의 천이에 따라서 모델이 제대로 설계 및 구현이 되어있는지를 확인하는 단계이다. 클래스 다이어그램으로만 모델링하게 되면 이런 기능은 전혀 사용하지 못한다. 비록 상태머신 다이어그램으로 설계할 경우에는 많은 노력이 필요하지만, 상태머신 다이어그램을 사용할 경우 더 가독성이 높은 모델로 설계할 수 있다. 이 단계에서는 기록 및 재생 모듈의 주요기능을 상태머신 다이어그램으로 작성하였다. 기록 및 재생의 주요기능은 다음과 같다.

- 기록 및 재생 모듈은 기록 및 재생 대상 장비에 대한 상태 정보를 관리하여 전시하여야 한다.
- 기록 및 재생 모듈은 기록파일을 리스트로 관리하며 추가 및 삭제가 가능하도록 하여야 한다.
- 기록 및 재생 모듈은 명령 실행 시 운용자에게 이를 알리고 정지 상태로 천이하여야 한다.
- 기록 및 재생 모듈은 제어하는 장비의 상태를 수신 받지 못하거나 예상하지 못한 상태가 올 경우에는 오류 상태로 천이 하여야 한다.

주요기능에 맞게 각 State를 식별하였고 State가 천이하는데 필요한 Event와 State의 행위를 명시하는 Action부분을 설계해 주었다.

- CRNRCtrlMgr 클래스에서는 기록 및 재생 통제 및 제어하는 클래스로 각 명령에 따라서 State를 천이하도록 만들었다. 또한, 기록 및 재생 명령을 처리하기 위해 큐를 사용하여 FIFO 방식으로 처리하였고 명령에 대한 유효성 검사, 장비상태정보를 취합해서 보고하는 기능을 설계해 주었다. 또한, 기록파일에 대한 추가/삭제 및 관리 기능들을 상태머신 다이어그램으로 설계해주었다.
- CRVRCtrlMgr 클래스에서는 레이더비디오 녹화기 제어 클래스로 장비의 상태정보에 따라서 오프라인 상태, 대기상태, 기록상태, 재생상태, 일시정지 상태로 천이하도록 설계하였다. 처음 상태는 오프라인상태로 장비들한테 상태정보를 받기 전에 상태로 정의하였다. 그리고 제어 및 통제하는 장비의 상태를 수신받기 시작하면 대기상태로 천이된다. 이것은 기록 및 재생을 시작할 수 있는 상태이다. 기록 및 재생을 사용하는 사용자의 명령에 따라 기록상태, 재생상태, 일시정지 상태로 천이한다. 또한, 각 상태에 따른 명령의 유효성 검사, 장비 명령에 대한 ACK처리, 상태정보 메시지 관리 등을 상태머신 다이어그램으로 설계해 주었다.

Fig. 7은 CRNRCtrlMgr 클래스에 대한 상태머신 다이어그램을 설계한 그림이고, Fig. 8은 CRVRCtrlMgr 클래스에 대한 상태머신다이어그램을 설계한 그림이다.

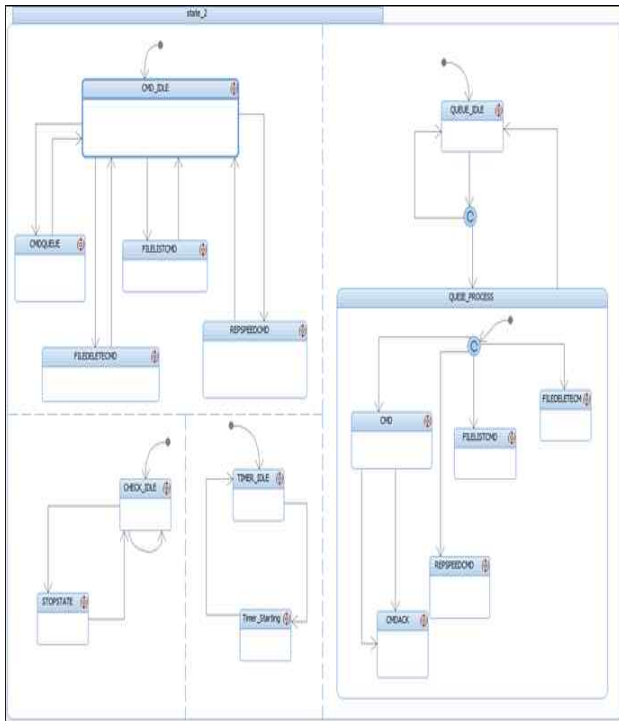


Fig. 7. State Machine Diagrams for CRNRCtrlMgr Class

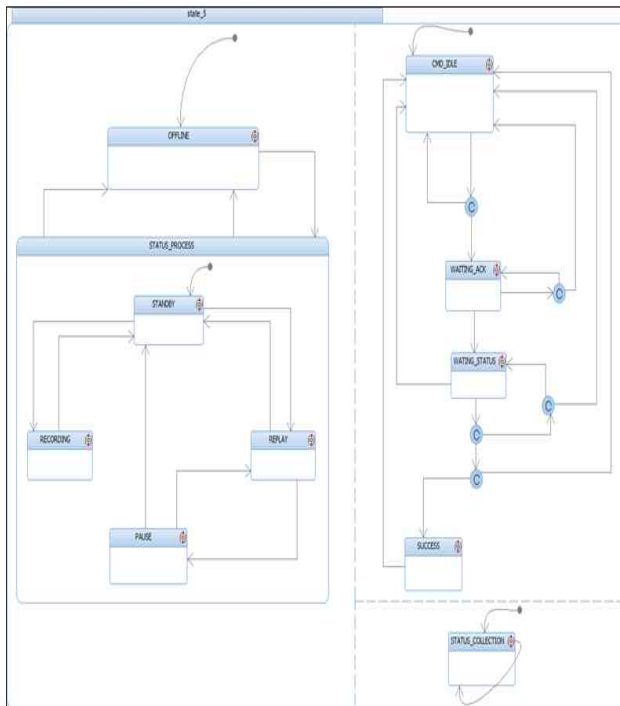


Fig. 8. State Machine Diagrams for CRVRCtrlMgr Class

상태머신다이어그램을 설계한 이후에는 설계 및 구현된 모델로부터 코드를 자동으로 생성하여 Compiler를 연동시켜 실행

파일을 만들어 주었다.

4. Step4 : Test

시험단계에서는 Rational Rhapsody에 있는 애니메이션 기능을 사용하여 시험하였다. 애니메이션 기능은 모델 유효성 검증을 가능하게 하는 Rational Rhapsody의 주요 기술이다. 애니메이션 기능을 사용하게 되면 모델들을 테스트를 할 수 있어, 더욱 더 신뢰성 있는 소프트웨어를 개발할 수 있다. 기록 및 재생 모듈의 클래스 다이어그램 설계 및 상태 머신 다이어그램 작성이 끝나고 모델에 대한 검증을 하였다. 또한, 고객의 요구 사항을 다 만족하는지 기능 시험도 하였다. 애니메이션 기능을 사용하여 Event를 발생시키고 그것에 대한 State 천이가 제대로 되는지, State의 Action이 정확하게 작동하는지 시험해 보았다. Fig. 9는 모델 시험 및 기능 시험에 대한 내용의 그림이다.

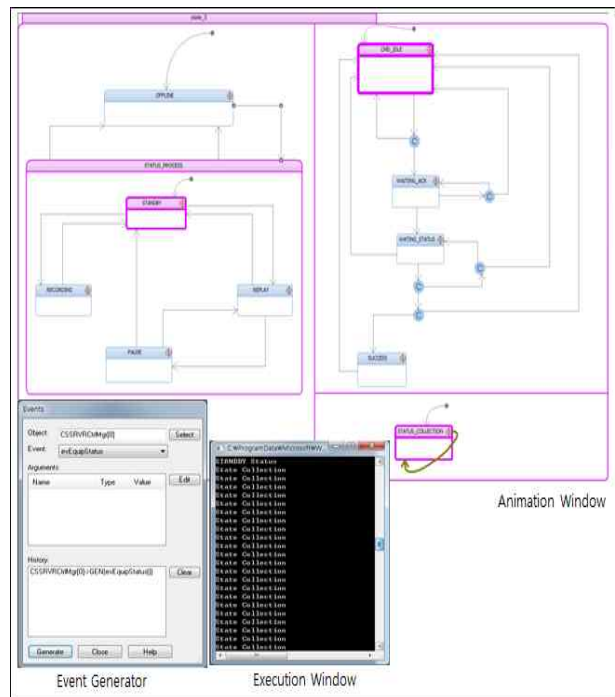


Fig. 9. Model Test & Function Test of Combat System Software

IV. Performance Analysis

성능검증을 위해 기존 전투체계 시스템의 기록 및 재생 모듈의 Legacy 코드와 Modeling 코드의 메모리(Code Memory, Data Memory) 사이즈, 처리속도 비교 및 정성평가를 해 보았다. 메모리 사이즈에서는 Modeling 코드가 Legacy코드보다 증가되었다. 하지만 Rational Rhapsody의 Modeling 코드에서는 OXF라는 프레임워크가 있어서 정확한 비교를 하기 위해서는

모델링 코드에서 OXF의 메모리 사이즈를 제한하게 되면 약간의 증가만 보였다. 이는 Windows OS에서 처리하는 속도로 보았을 때 그 영향성은 미비하다. Fig. 10은 메모리 사이즈의 결과를 보여준다.

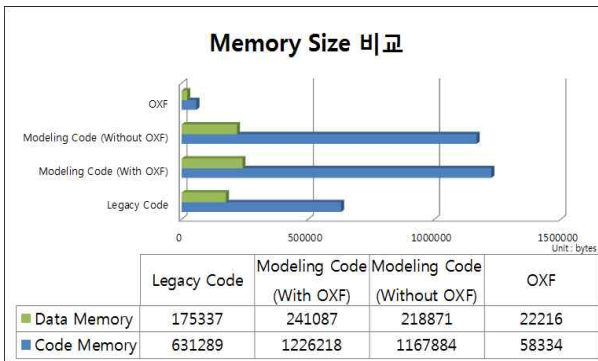


Fig. 10. Test Result for Memory Size

기능적인 면에서는 2시간 기록 후 기록된 파일을 재생했을 경우의 처리 속도를 비교해 보았다. 기록 및 재생 모듈에서는 수많은 정보를 기록하기 때문에 재생 했을 때 처리 속도가 나 빠지면 기록 했을 때의 시점과 재생 했을 때의 시점이 달라지기 때문에 기능적으로 큰 문제를 발생시킨다. 따라서 기록 및 재생 모듈의 가장 큰 기능인 Legacy 코드의 처리 속도와 Modeling 코드의 처리속도를 비교해 보았다. 5분, 10분, 20분, 40분, 60분단위로 재생을 해서 각 시간을 측정해 보았다. 처리 속도면 에서는 5분 재생 했을 때는 2ms 차이가 났고 60분 재생 했을 때는 24ms 차이가 났다. 시험결과 재생 처리 속도면 에서는 Legacy 코드와 Modeling 코드의 차이가 거의 없었다. Fig. 11은 처리속도의 결과를 보여준다.

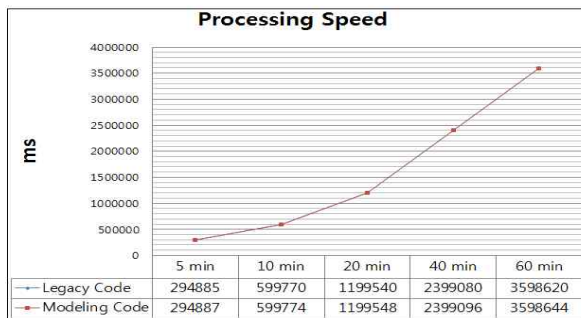


Fig. 11. Test Result for Processing Speed

정성적인 평가 면에서 Legacy 코드와 Modeling 코드를 비교하여 예측해 보았다. Programming Language의 정성적 평가 기준은 신뢰성, 효율성, 유지보수성, 이식성 이다[18]. 신뢰성면 에서는 정해진 기능을 유지하는 정도에 대한 품질 특징으로 Legacy코드와 비교 했을 때 차이가 거의 없었다. 하지만 새롭게 추가되는 기능에 대해서는 Modeling 코드가 더 좋을 것이라 예측된다. 효율성 면에서는 Legacy 코드가 수많은 시험을

하였기에 Modeling 코드보다는 좋을 것이라고 예측된다. 하지만 Modeling 코드도 많은 시험을 하고 좀 더 최적화 하면 더욱 좋아 질것이라고 생각한다. 유지보수성에서는 Modeling코드가 휘쳐 모델과 디자인패턴을 사용하여 설계하였기 때문에 더 좋을 것이라고 예측된다. 이식성에서는 Legacy 코드나 Modeling 코드에서 OS 변경에 따라 사용하는 방법을 가지고 있어 차이가 없는 것으로 판단된다.

V. Conclusions

본 논문에서는 효율적인 전투체계 소프트웨어 개발방법으로 UML을 이용한 모델 기반 개발 방법에 대해서 설명하였다. 그리고 전투체계 소프트웨어의 한 모듈인 기록 및 재생 모듈에 모델 기반 개발 방법을 적용하면서 개발방법에 대해서도 설명하였다. 또한, Legacy 코드와 Modeling 코드의 메모리 사이즈와 처리속도 비교 및 정성평가를 통해 이 방법에 대한 가능성을 입증하였다. 만약, 본 논문에서 제시한 개발방법을 사용한다면 전투체계 소프트웨어 개발 시 시스템에 대한 오류를 줄이고 향후 추가되는 요구사항에 대한 수정이 용이하다는 장점을 가진다. 이로 인해 기존의 코딩중심의 개발방식 보다 더욱 더 신뢰성이 높은 전투체계 소프트웨어를 개발할 수 있다.

추후 연구 과제에서는 프로파일 확장 등을 통해 전투체계 도메인 특성에 맞는 모델들을 생성하여 전투체계 소프트웨어에 적용하기 위한 더 많은 연구가 필요하다.

REFERENCES

- [1] Moon-Seol Kang, Tae-Hee Kim, "Standardization of object-oriented software development methodology(UML: Unified Modeling Language)," Korea Information Processing Society Review, Vol. 5 No. 5, pp. 64-73, Sep. 1998.
- [2] G. Martin, "UML for Embedded Systems Specification and Design : Motivation and Overview," Proc. Design Automation and Test in Europe Conference and Exhibition 2002, pp.773-775, 2002.
- [3] A. E. Ibrahim, L. Zhao, and J. Kinghorn, "Embedded Systems Development: Quest for Productivity and Reliability,"Proc. ICCBSS 2006, pp.13-16, 2006.
- [4] J. W. Choi and D. J. Lim, "A Study on the Development of Embedded System Software for Ubiquitous Sensor Networks using UML," Proc.

- FeT' 2009 IFAC International Conference, pp.230-233, 2009.
- [5] Modeling tool Rational Rhapsody from IBM, Homepage:<http://www-03.ibm.com/software/products/en/ibmratirhapdesimana>
- [6] M. U. Khan, K. Geihs, F. Gutbrodt, P. Gohner, and R. Trauter, "Model-Driven Development of Real-Time Systems with UML 2.0 and C," Proc.MBD/OMPES2006, pp.33-42, 2006.
- [7] R. Mura, "Code Generation from Statecharts: Simulation of Wireless Sensor Network," Proc. Digital System Design Architectures, Methods and Tools 2008, pp.525-532, 2008.
- [8] Yun-Seok Choi, "A Transformation Technique of PIM to PSM based on UML Profiles for Mobile Applications," Journal of KSCI, Vol. 17, No. 6, pp. 131~144, Jun. 2012.
- [9] Jeong-Il Choi, "A Case Study of SW Development Productivity Improvement by MDD/MDA Technology Application in CBD Project," Journal of the Korea society of IT services, Vol. 5, No. 2, pp.179~188, 2006.
- [10] Seung-Han Lee, Jae-Pyo Park, "A Case Study of Software Development Quality Improvement by Agile Methodology and MDA/MDD Technology," Journal of the Korea Academia-Industrial cooperation Society, Vol. 16, No. 4, pp.2744-2748, 2015
- [11] <http://www.omg.org/mda/>
- [12] Jin Lee, Jungsun Kim, "A MDA-based Application Authoring Tool Architecture for Smart Cross-Platform," Journal of KIISE : Computer Systems and Theory, Vol. 40 No. 3, pp.139-148, Jun. 2013
- [13] <http://i-bada.blogspot.kr/2012/04/mda-model-driven-architecture-mdd-model.html>
- [14] Ji hun Park, Je Min Bae, Kyung Whan Lee, Do Soo Bae, Kyung Jae Ryu, "Object Oriented Development Method Applying Design Patterns to Intranet Application," KIISE Conferences, Vol. 23, No. 2B, pp.1481-1484, 1996
- [15] Yun-Sung Kang, Jun-Hwan Lee, Han-Jin Cho, "Design and Implementation of .NET Remoting Common Framework Applied Design Pattern," J. of Contents Association, Vol. 11, No. 3, pp.36-47, Mar. 2011
- [16] Tung M. Dao, Kyo C. Kang "Integrating Feature-Oriented Software Product Line Engineering with Problem Frames," Journal of KIISE : SA, Vol. 38, No. 3, pp. 124~133, Mar. 2011.
- [17] Dong-Ri Kim, Chee-Yang Song, Dong-Su Kang, Doo-Kwon Baik, "An Ontology - based Transformation Method from Feature Model to Class Model," Journal of KSCI, Vol. 13, No. 5, pp. 53~67, Sep. 2008.
- [18] Sang-Hoon Chi, Kyu-Boem Lee, Sou-Kil Lee, Seok-Kwon Youn, Kyung-Whan Lee, Jeong-Young Song, "Qualitative Evaluation for a Programming Languages," Proceedings of KIIT Summer Conference, pp. 76~79, Aug. 2003.

Authors



Seung Mo Jung received the B.S. degree in Electronic Engineering from Gangneung National University, Korea, in 2006. the M.S. degree in Electronic, Electrical, Control and Instrumentation Engineering from Hanyang University, Korea, in 2010.

He is currently working in Hanwha Systems Co. from 2010. He is interested in Embedded Software, Combat System Software, and Model Driven Development.



Young Ju Lee received the B.S., M.S. degrees in Electrical Engineering from Yonsei University, Korea, in 2001 and 2003, respectively.

He is currently working in Defense Agency for Technology and Quality from 2006. He is interested in Biomedical signal processing, Combat System, and Model Driven Development.