

A Study on an Automatic Model Creation Tool for Applying Structured UML Models in Software Development

Seungmo Jung[†] · Woojin Lee^{††}

ABSTRACT

Recently, large-scale software development has been using the highly readable Unified Modeling Language (UML) development method. The use of standardized UML models in software development improves software quality by resolving unclear communication. However, in existing software development, a code-centric development method is applied rather than a model-centric development method. As a result, problems such as increased model creation work time arise because UML models must be manually handled when developing model-centric software using existing software. In addition, depending on the developer's understanding of the model, the ability to use modeling tools, and the complexity of the software structure, the time required to create the model increases further. The increase in model creation work time is a factor that increases the overall software development time. Therefore, this paper proposes an automatic model creation tool for structurally applying UML models to the development of naval combat system software. The Automatic Model Creation Tool provides features that automatically generate model structures and UML models needed for modeling tasks. Using the method proposed in this paper, it has the advantage of structurally applying UML models through automation functions and efficiency in reducing model creation work time.

Keywords : Unified Modeling Language, Model-Centric Development, Modeling Tool, Automatic Model Creation, Naval Combat System Software

소프트웨어 개발에 구조화된 UML 모델을 적용하기 위한 자동 모델 생성 도구에 관한 연구

정 승 모[†] · 이 우 진^{††}

요 약

최근 대규모 소프트웨어 개발에 가독성이 높은 Unified Modeling Language (UML)을 활용한 개발 방법이 적용되고 있다. 소프트웨어 개발에 표준화된 UML 모델을 활용하면 불명확한 의사소통을 해결할 수 있어 소프트웨어 품질을 향상 시킨다. 하지만 기존 소프트웨어 개발에서는 모델 중심의 개발 방식이 아닌 코드 중심의 개발 방식이 적용되어 있다. 이로 인해 기존 소프트웨어를 활용하여 모델 중심의 소프트웨어를 개발 할 경우 수동으로 UML 모델을 적용해야 하기 때문에 모델 생성 작업 시간 증가 등의 문제가 발생한다. 또한, 개발자들의 모델 이해도 및 모델링 도구의 사용 능력, 소프트웨어 구조의 복잡도에 따라 모델 생성 작업 시간은 더욱 증가한다. 모델 생성 작업 시간의 증가는 결국 전체 소프트웨어 개발 시간을 증가시키는 요인이 된다. 이에 본 논문에서는 합성 전투체계 소프트웨어 개발에 UML 모델을 구조적으로 적용하기 위한 자동 모델 생성 도구를 제안한다. 자동 모델 생성 도구는 모델링 작업에 필요한 모델 구조 자동 생성과 UML 모델들을 자동으로 생성하는 기능을 제공한다. 본 논문에서 제안한 방법을 활용하면 자동화 기능을 통해 UML 모델을 구조적으로 적용할 수 있는 장점과 모델 생성 작업 시간 감소의 효율화를 가진다.

키워드 : 통합모델링언어, 모델 중심 개발, 모델링 도구, 모델 자동 생성, 합성전투체계 소프트웨어

1. 서 론

대규모 소프트웨어는 매우 크고 복잡한 소프트웨어 시스템 을 가리키며 수십 또는 수백 개의 다양한 기능들이 포함되어 있다. 대규모 소프트웨어의 다양한 기능을 개발하기 위해 기능들을 그룹화하고 그룹화된 기능들을 여러 개발자들에게 할 당되어 동시에 개발된다. 여러 개발자들이 다양한 기능의 소프트웨어를 서로 협력하여 개발할 때 개발자들 간의 명확한

※ 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2018-NR031059).

† 준 회 원 : 경북대학교 컴퓨터학부 박사과정

†† 정 회 원 : 경북대학교 컴퓨터학부 교수

Manuscript Received : August 26, 2024

First Revision : October 24, 2024

Accepted : November 26, 2024

* Corresponding Author : Woojin Lee(woojin@knu.ac.kr)

의사소통은 중요한 요소이다[1, 2]. 소프트웨어 개발 과정 중 불명확한 의사소통이 발생하면 시스템에 예기치 못한 오류가 발생할 수 있다. 이러한 오류를 해결하기 위해 개발기간이 증가되고 개발기간 증가에 따른 개발 비용도 증가된다. 대규모 소프트웨어 개발에 불명확한 의사소통을 해결하고 소프트웨어 품질을 향상시키기 위해 모델 중심의 개발 방법이 적용되고 있다[3, 4]. 모델 중심의 개발 방법이란 문서화된 코드 중심의 개발 방법이 아닌 가독성이 높은 모델을 활용하여 개발하는 방식으로 Unified Modeling Language (UML) 모델 객체를 활용한 소프트웨어 개발 방법이 표준으로 널리 사용되고 있다 [5]. 표준화된 UML 모델 객체들을 이용하여 소프트웨어를 개발하기 때문에 개발자들 간의 의사소통을 원활하게 도와주는 장점을 가진다[6, 7].

함정 전투체계(Naval Combat Management System/CMS)를 개발하기 위한 대규모 소프트웨어인 함정 전투체계 소프트웨어 개발에서도 원활한 의사소통에 장점을 가진 모델 중심의 개발 방법이 적용된다. 함정의 두뇌와 같은 역할을 수행하는 함정 전투체계는 함정에 탑재된 센서장비, 무장장비 등을 연동/제어하여 작전에 필요한 정보들을 종합적으로 수집한다. 그리고 그 수집된 정보들을 활용하여 전술상황평가, 교전 등의 기능을 효율적으로 수행할 수 있도록 지원하는 일련의 자동화된 무기체계이다[8, 9]. 함정 전투체계의 효율적인 기능을 수행하기 위해 다양한 기능의 소프트웨어들이 유기적으로 실행되고 있다. 함정 전투체계 소프트웨어 개발에 모델 중심의 개발 방식을 적용하기 위해 풍부한 기능을 가진 모델링 도구를 사용한다.

하지만 표준 UML 모델 객체를 특정 도메인 특성을 가진 함정 전투체계 소프트웨어에 수동으로 적용하는데 어려움이 발생한다. 소프트웨어 개발 초기 통일성 있는 UML 모델을 적용하기 위해 문서 방식의 모델 적용 가이드라인을 제공하지만 개발자 마다의 모델 이해도 및 모델링 도구의 사용 능력에 따라 달리 적용될 수 있다. 또한, 코드 중심의 기존 소프트웨어를 활용 할 경우 실행 가능한 클래스 모델로 변경하는 것에는 한계가 발생한다. 모델링 도구에서도 역공학 기능을 통해 레거시 코드를 입력으로 사용하여 클래스 모델로 변경하는 기능을 제공 하지만 설계용 으로만 변경할 뿐 실행 가능한 클래스 모델로 변경하는 기능은 제공하지 않는다. 이로 인해 레거시 코드에서 설계용 클래스 모델로 자동 변경되어도 개발자가 실행 가능하도록 수동으로 수정 해야 하기 때문에 모델링 작업 시간 증가의 단점을 가지고 있다.

이에 본 논문에서는 함정 전투체계 소프트웨어 개발에 UML 모델을 구조적으로 적용하기 위한 자동 모델 생성 도구 (AMCT : Automatic Model Creation Tool)를 제안한다. 자동 모델 생성 도구는 UML 모델링 작업을 하기 위해 필요한 모델 구조 및 모델들을 자동으로 생성하는 기능을 제공한다. 본 논문에서 제안한 자동 모델 생성 도구기능을 활용하면 UML 모델을 구조적으로 적용할 수 있는 장점과 모델 생성 작업 시간

감소의 효율화를 가진다.

본 논문의 구성은 다음과 같다. 2장에서는 자동 모델 생성 도구를 개발 하는데 필요한 관련 연구에 대해서 간략히 설명하고 3장에서는 자동 모델 생성 도구의 개발 방법에 대해서 설명 한다. 그리고 4장에서는 개발된 자동 모델 생성 도구를 함정 전투체계 소프트웨어 개발에 일부 적용하여 기능과 성능을 입증한다. 마지막으로 5장에서는 결론으로 이 논문을 마무리한다.

2. 관련 연구

2.1 UML 모델 자동 생성

소프트웨어 시스템의 설계와 구조를 시각적으로 표현하기 위해 사용되는 UML은 표준화된 모델링 언어의 장점으로 인해 소프트웨어 개발에 많이 적용된다. 또한, 도메인 특성을 가진 소프트웨어 개발에 쉽게 적용하기 위해 많은 도구들이 개발되고 있다[10-14]. 하지만 대부분 UML 모델에 대한 코드 생성 및 검증에 목적을 가지고 있고 원하는 형태의 UML 모델 생성에 대한 도구들은 거의 없다. 특히, 레거시 코드를 활용하여 클래스 모델로 변환할 경우 모델링 도구의 대부분이 설계용 클래스 모델로만 변환하는 기능을 제공할 뿐 개발자 수정 없이 실행 가능한 클래스 모델로 변환하는 기능은 제공하지 않는다. 본 논문에서 제안한 자동 모델 생성 도구는 모델 접근 언어를 활용하여 도메인 특성에 맞는 UML 모델 생성 기능과 실행 가능한 클래스 모델 변환 기능을 제공한다.

2.2 모델링 도구의 모델 접근 언어

UML은 시스템의 전반적인 개발 과정에서 사용하는 표준화된 객체지향 통합 모델링 언어로 다양한 소프트웨어 기술 및 표준을 유지 관리하고 있는 국제 표준화 기구인 Object Management Group (OMG)에서 관리한다[15, 16]. UML의 구성요소는 시스템에 대한 추상화된 개념을 표현하는 Things, 사물 간의 연결과 상호작용을 표현하는 Relationships, 사물 및 관계를 시각적으로 표현하는 Diagrams로 구성되어 있으며 UML 모델들을 소프트웨어 개발에 적용하기 위해 다양한 모델링 도구가 있다. 다양한 모델링 도구 중에서 Rational Rhapsody는 UML의 장점을 살릴 수 있도록 고안된 실무형 개발 프로세스로서 UML 표기법을 시각화하여 모델로 제공하고 모델에 대한 코드를 자동으로 생성시켜 주어 설계와 구현을 모델링 도구에서 직접 할 수 있는 장점이 있다[17-19]. Fig. 1은 모델링 도구인 Rational Rhapsody의 구조 및 프로세스를 보여준다.

Rational Rhapsody의 기본 프로세스는 ML3, ML2, ML1, ML0 계층을 통해서 표준 UML 모델을 제공하고 제공된 모델로부터 소스코드를 생성한다. 소프트웨어 개발자들은 모델링 도구에서 제공된 UML 모델들을 사용하여 소프트웨어를 개발하고 작성된 UML 모델로부터 소스코드를 생성하여 실행파일

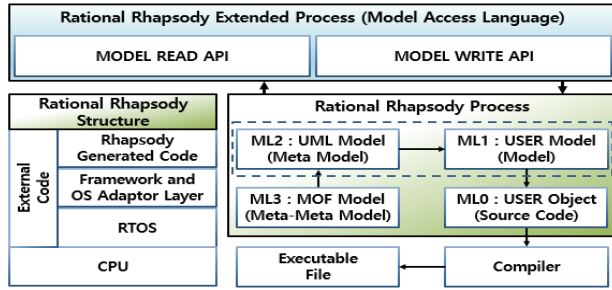


Fig. 1. Rational Rhapsody Structure and Process

로 변경한 후 시스템에 탑재한다. 모델 접근 언어는 모델링 도구에서 제공하는 API로 모델링 도구에서 제공하는 모든 모델에 접근 가능하도록 만들어 준다. 모델 접근 언어는 크게 MODEL READ API와 MODEL WRITE API를 제공한다. MODEL READ API는 모델에 접근하여 모델 정보를 읽어오는 기능을 제공하며 MODEL WRITE API는 개발자가 원하는 형태로 모델을 만들어 사용할 수 있는 기능을 제공한다. 본 논문에서 개발된 자동 모델 생성 도구는 MODEL WRITE API를 활용하여 도메인 특성을 반영한 UML 모델들을 자동으로 생성한다.

2.3 모델링 기반 해양 전투 시스템 소프트웨어 개발

함정전투체계 소프트웨어 개발은 기존 코드 중심의 개발 방식에서 변화하여 모델링 도구를 활용한 모델 중심의 개발 방식을 기본적으로 적용하고 있다[20]. Fig. 2는 모델링 기반 함정 전투체계 소프트웨어 개발을 도식화하여 보여준다. 소프트웨어 개발자들은 요구사항 분석(소프트웨어 기능 정의), 모델 적용 가이드라인(모델 적용에 필요한 문서화된 가이드라인), 메시지 정의(소프트웨어 모듈간에 기능을 수행하기 위해 송/수신 하는 메시지들을 정의), 재사용이 가능한 레거시 코드(기존 시스템에서 재사용이 가능한 코드)를 참고하여 모델링 작업을 한다.

모델링 작업에 필요한 다이어그램은 유스케이스 다이어그램, 액티비티 다이어그램, 클래스 다이어그램, 시퀀스 다이어그램이다. 유스케이스 다이어그램은 시스템과 사용자 간의 상호작용을 표현하는 다이어그램이고 액티비티 다이어그램은 시스템 또는 비즈니스 프로세스의 동작을 시각적으로 표현하는 다이어그램이다. 클래스 다이어그램은 시스템을 구성하는 클래스 간의 관계를 표현하는 다이어그램이고 시퀀스 다이어

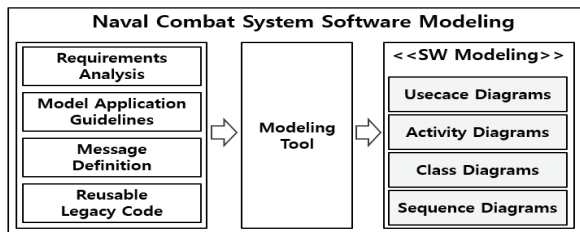


Fig. 2. Software Development base on Modeling

그램은 시간 순서에 따라 정렬된 객체 상호작용을 표현하는 다이어그램이다. 소프트웨어 개발자들은 모델링 도구인 Rational Rhapsody에서 제공된 UML 모델들을 활용하여 함정 전투체계 소프트웨어 개발에 필요한 다이어그램들을 작성한다.

3. 제안 방법

3.1 자동 모델 생성 도구 개요

표준화된 UML 모델들을 특성화된 해양 전투체계 도메인 기반 소프트웨어 개발에 통일성 있게 적용하기 위해 모델 적용 가이드라인을 활용한다. 모델 적용 가이드라인은 모델 적용 범위 등 모델을 적용하기 위한 설명이 문서화 방식으로 표현되어 있다. 각 개발자들은 배포된 모델 적용 가이드라인을 참고하여 모델링 작업에 필요한 모델 구조 및 모델들을 수동으로 생성하여 소프트웨어 개발에 활용한다. 또한, 레거시 코드를 재사용할 경우에는 레거시 코드를 실행 가능한 클래스 모델로 수동으로 생성하여 사용한다. 하지만 모델링 작업에 필요한 모델들을 수동으로 생성할 경우 모델 생성 작업 시간 증가 및 모델을 활용한 소프트웨어 개발 작업 시간을 증가시키는 문제점을 발생시킨다. 모델을 적용하는 소프트웨어가 많을수록 개발 시간은 더욱 증가한다. 이러한 문제점을 해결하기 위한 자동 모델 생성 도구는 모델 생성에 필요한 필수 정보들만 입력으로 받아 자동으로 UML 모델들을 생성해 주는 기능을 제공한다. 이러한 자동화 기능을 활용하면 모델 생성 작업 시간이 감소 되기 때문에 소프트웨어 개발 시간을 감소시키는 장점을 가진다.

Fig. 3은 자동 모델 생성 도구의 개요를 도식화한 그림이다. 소프트웨어 개발자들이 소프트웨어 구조 정보 및 모델에 대한 기본 정보를 입력하면 입력된 정보들을 자동으로 추출하여 모델 구조 및 정의 모델을 자동으로 생성하는 기능을 수행한다. 함정 전투체계에서는 소프트웨어들 간의 메시지를 송/수신 하기 위해 Message Oriented Management Analysis Tool (MOMAT)을 사용하여 광범위한 메시지에 대한 정의 및 관리를 한다[21]. 이벤트 모델 자동 생성 기능은 MOMAT에 정의된 메시지 정보들을 자동 추출하여 수행한다. 또한, 재사용이 가능한 레거시 코드를 활용하기 위해서 레거시 코드를 파싱하여 필요한 정보들을 추출한 후 실행 가능한 클래스 모델로 자동 생성하는 기능을 수행한다. 자동 생성된 모델들은 모델링 작업에 활용한다.

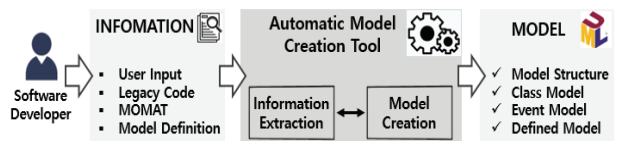


Fig. 3. Overview of Automatic Model Creation Tool

3.2 자동 모델 생성 도구 개발 방법

자동 모델 생성 도구 개발에 필요한 요구사항 분석과 요구 사항에 따른 기능을 수행하기 위한 자동 모델 생성 도구 설계 및 구현 과정에 대해서 설명한다.

1) 자동 모델 생성 도구 요구사항

요구사항 분석 단계는 자동 모델 생성 도구 개발에 필요한 요구사항에 대해서 분석하는 단계이다. 자동 모델 생성 도구의 기능을 수행하기 위해 4가지 자동 생성 요구사항들을 수립하였고 Fig. 4는 자동 모델 생성 도구에 대한 요구사항을 도식화한 그림이다. 자동 모델 생성 도구는 모델 구조 자동 생성 기능, 클래스 모델 자동 생성 기능, 이벤트 모델 자동 생성 기능, 정의 모델 자동 생성 기능을 포함하고 있다. 모델 구조 자동 생성 기능에서는 소프트웨어 구조의 기본 정보를 입력으로 받아 UML 모델들을 작업할 수 있는 구조를 자동으로 생성한다. 구조 자동 생성에는 해양 전투체계 도메인 소프트웨어 개발에 필요한 유스케이스 다이어그램, 액티비티 다이어그램, 클래스 다이어그램, 시퀀스 다이어그램을 작업 할 수 있는 트리구조를 자동 생성한다. 모델링 작업을 할 수 있는 구조를 자동으로 생성하면 수동으로 생성하는 것보다 다양한 소프트웨어에 대한 통일성 있는 구조를 생성할 수 있다. 클래스 모델 자동 생성 기능에서는 실행 가능한 클래스 모델을 자동으로 생성한다. 헤더파일에 정의된 정보들을 파싱하여 변수, 함수 등 클래스 모델의 뼈대를 생성하고 소스 파일에 정의된 정보들을 파싱하여 함수의 내용을 추가한다. 또한, 여러 클래스들의 관계를 생성하고 실행 파일을 생성할 때 필요한 설정 부분도 자동으로 생성한다. 이러한 과정을 통해 설계용 클래스 모델이 아닌 실행 가능한 클래스 모델을 자동으로 생성한다. 이벤트 모델 자동 생성 기능에서는 통신과 관련 된 모델을 자동으로 생성한다. MOMAT에 정의된 정보들을 입력으로 활용하여 광범위한 메시지에 대한 송신 모델과 수신 모델을 생성한다. 생성된 송/수신 모델은 액티비티 다이어그램, 시퀀스 다이어그램 모델링 작업에서 메시지 송/수신 부분을 표현할 때 사용한다. 마지막으로 정의 모델 자동 생성 기능에서는 사업에서 필요한 액터와 스테레오 타입 모델을 정의하여 생성한다. 이 모델은 유스케이스 다이어그램과 시퀀스 다이어그램 모델링 작업에서 활용한다.

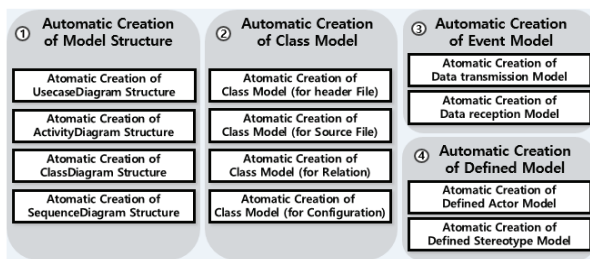


Fig. 4. Requirement for Automatic Model Creation Tool

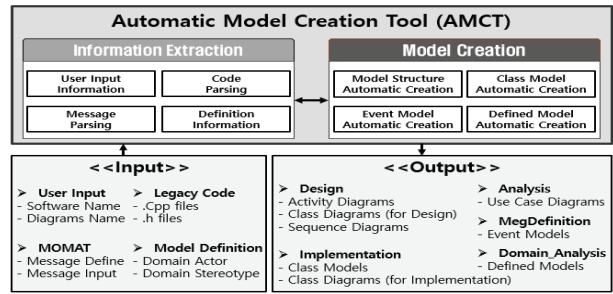


Fig. 5. Automatic Model Creation Tool Process

2) 자동 모델 생성 도구 설계

자동 모델 생성 도구는 입력 정보들을 활용하여 모델 구조 및 모델링 작업에 필요한 모델들을 자동으로 생성하는 기능을 수행한다. Fig. 5는 자동 모델 생성 도구의 프로세스를 도식화한 그림이다. 입력 정보에는 소프트웨어 구조에 대한 개발자 입력 정보, 모델로 변환하고 싶은 레거시 코드 정보, MOMAT에 정의된 메시지 정보, 사업에 정의된 정보들이다. 자동 모델 생성 도구는 입력 정보들에서 필요한 정보들을 자동 추출하여 모델 구조, 클래스 모델, 이벤트 모델, 정의 모델을 자동 생성한다. 모델 구조 자동 생성은 개발자의 입력 정보인 소프트웨어 이름과 다이어그램 이름 정보 들을 추출하여 모델 작성이 필요한 트리구조를 자동 생성한다. 클래스 모델 자동 생성은 개발자의 입력 정보, 레거시 코드 정보, 이벤트 정보들을 활용하여 모델 생성에 필요한 정보들을 파싱하여 실행 가능한 클래스 모델을 자동 생성한다. 이벤트 모델 자동 생성은 메시지 정의 틀인 MOMAT에 입력된 메시지 정보들을 파싱하여 송/수신 이벤트 모델들을 자동 생성한다. 정의 모델 자동 생성은 사업 개발에 필요한 모델 정보들을 활용하여 모델 작성에 필요한 액터 모델과 스테레오 타입 모델을 자동 생성한다.

3) 자동 모델 생성 도구 구현

자동 모델 생성 도구의 기능을 수행하기 위해 필요한 입력 정보 들은 소프트웨어 구조 정보, 생성이 필요한 다이어그램 이름, 모델 변환이 필요한 레거시 코드 정보, 메시지 송/수신 정보, 사업에 필요한 모델 정보들이다. 소프트웨어 구조 정보는 다양한 소프트웨어들의 형상 관리를 하기 위해 분리한 소프트웨어 집합체로써 자체 운용자 인터페이스를 통해 모델 구조 생성에 필요한 정보들을 취합한다. 생성이 필요한 다이어그램 이름은 모델링 작업이 필요한 다이어그램들의 이름으로 자체 운용자 인터페이스를 통해 필요한 정보들을 취합한다. 모델 변환이 필요한 레거시 코드 정보는 기존의 코드 중심 소프트웨어의 헤더파일과 소스파일 정보를 파싱하여 클래스 모델 생성에 필요한 정보들을 취합한다. 메시지 송/수신 정보는 메시지 정의 틀인 MOMAT에 입력된 정보들을 파싱하여 필요한 정보들을 취합한다. 사업에 필요한 모델 정보들은 사업을 설계할 때 정의한 모델 정보 파일을 파싱하여 모델 생성에 필요한 정보들을 취합한다. 모델 생성 도구는 취합된 입력 정보

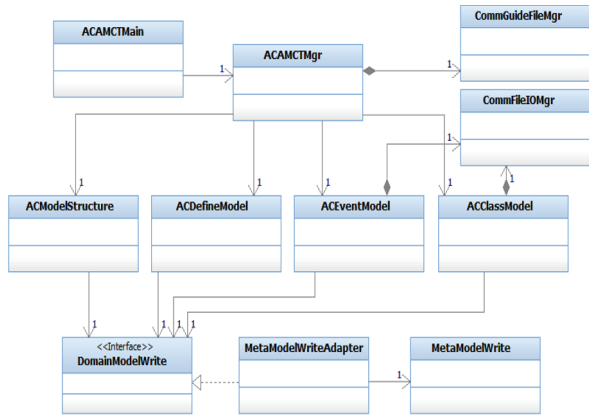


Fig. 6. Class Diagram of Automatic Model Creation Tool

들에서 모델 생성에 필요한 정보들을 자동으로 추출하여 모델 구조 자동 생성, 클래스 모델 자동 생성, 이벤트 모델 자동 생성, 정의 모델 자동 생성 기능을 수행한다. Fig. 6은 자동 모델 생성 도구 구현에 대한 클래스 다이어그램을 도식화하여 보여 준다.

자동 모델 생성 도구의 기능을 구현하기 위한 클래스 다이어그램에서 주요 기능을 담당하는 클래스는 아래와 같다.

- a) ACAMCTMgr: 자동 모델 생성 도구의 기능을 관리하는 클래스로 운용자 인터페이스, 파일을 통해서 모델 생성에 필요한 정보들을 수집하여 각 기능 담당 클래스로 필요한 정보들을 전달 해주는 기능을 수행한다.
- b) ACModelStructure: 모델링 작업에 필요한 모델 구조를 자동으로 생성하는 기능을 담당하는 클래스로 운용자 인터페이스에 입력된 정보를 바탕으로 소프트웨어의 구조에 따라 Package와 NestedPackage 들을 자동으로 생성하는 기능을 수행한다. 또한, Usecase Diagram, Activity Diagram, Class Diagram, Sequence Diagram 등 모델링 작업을 위해 필요한 기본 구조를 자동으로 생성하는 기능을 수행한다.
- c) ACClassModel: 실행 가능한 클래스 모델을 자동으로 생성하는 기능을 담당하는 클래스로 헤더파일과 소스파일에 파싱된 정보들을 추출한다. 추출된 정보들을 기반으로 Attribute 모델과 Operation 모델을 생성하여 최종적으로 클래스 모델로 변환하는 기능을 수행한다. Attribute 모델에는 Type, Name, Comments 등을 포함하고 Operation 모델에는 Type, Name, Parameter Type, Parameter Name, Comments, Contents 등을 포함한다.
- d) ACEventModel: 이벤트 모델을 자동으로 생성하는 기능을 담당하는 클래스로 메시지 정의 툴인 MOMAT에서 파싱된 송/수신 메시지 정보들을 활용하여 송신 메시지 이벤트 모델과 수신 메시지 이벤트 모델로 변환하는 기능을 수행한다.
- e) ACDefineModel: 정의 모델을 자동으로 생성하는 기능을 담당하는 클래스로 사업에 정의된 Actor와 Stereotype

정보들을 활용하여 Actor와 Stereotype의 특성을 가진 프로파일 모델로 변환하는 기능을 수행한다.

- f) MetaModelWrite : 모델 접근 언어인 Model Write API를 통해 Meta Model에 접근하여 필요한 UML 모델을 최종적으로 생성하는 기능을 담당한다.
- g) CommFileIOMgr : 파일 정보를 읽어서 파싱하는 기능을 담당하는 클래스로 헤더파일에서 클래스명, 변수타입, 변수명, 변수주석, 함수타입, 함수명, 매개변수타입, 매개변수명을 파싱하고 소스파일에 함수의 내용을 파싱한다. 또한, 소프트웨어 간의 통신이 필요한 송/수신 메시지에 대한 정보도 파싱한다.

4. 실험 결과

4.1 자동 모델 생성 도구에 대한 기능 시험

자동 모델 생성 도구의 기능적인 면을 시험하기 위해 함정 전투체계 소프트웨어 중 전술정보 기록 및 재생 소프트웨어를 대상으로 시험하였다. 전술정보 기록 및 재생 소프트웨어는 전술상황을 수행하면서 다양한 소프트웨어에서 발생하는 전술정보들을 기록하였다가 사후분석/전술상황평가를 위해서 기록된 전술정보를 재생하는 기능을 담당한다. 구현이 완료된 자동 모델 생성 도구를 실행하면 Fig. 7과 같이 모델 생성에 기본적인 정보들을 수집하기 위한 자체 운용자 인터페이스가 전시된다. 개발자들은 전시된 운용자 인터페이스를 통해서 모델 생성에 필요한 기본적인 정보들을 입력한다. Project Name은 사업의 이름 정보를 입력하며 CSCI-CSC-CSU 구조는 소프트웨어 형상 관리를 위한 구조를 입력한다. Computer Software Configuration Item (CSCI)는 소프트웨어 형상항목을 나타내며 Computer Software Component (CSC)는 소프트웨어 구성품을 나타낸다. 또한, Computer Software Unit (CSU)는 시험할 수 있는 최소 단위의 단위 소프트웨어를 나타낸다. 그리고 Diagram Name은 소프트웨어 개발에 모델 중심의 개발 방식을 적용하기 위해 필요한 다이어그램 이름을 입력한다. 정보 입력이 끝나면 Model Structure Automatic Creation, Class Model Automatic Creation, Event Model Automatic Creation, Defined Model Automatic Creation 버튼을 클릭

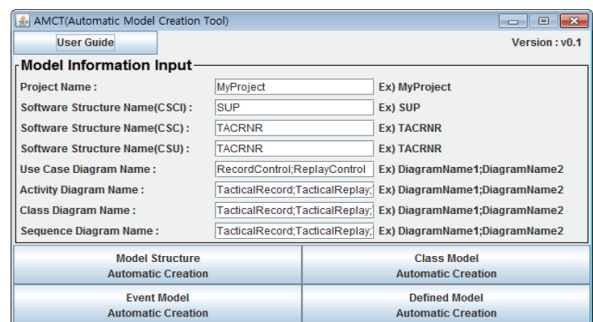


Fig. 7. User Interface of Automatic Model Creation Tool

하여 모델링 작업에 필요한 모델들을 자동 생성한다.

자동 모델 생성 도구를 통해 자동 생성된 모델들이 제대로 생성되었는지 문서화 방식의 모델 적용 가이드라인과 비교하여 검증하였다. 모델링 작업이 필요한 UML 모델 구조는 4개의 모델 패키지와 1개의 프로파일로 구성되어 있다. Analysis 모델 패키지는 소프트웨어 요구사항 모델을 정의하는 영역으로 유스케이스 다이어그램 모델링 작업을 한다. Design 모델 패키지는 소프트웨어 설계 모델을 정의하는 영역으로 액티비티 다이어그램, 설계용 클래스 다이어그램, 시퀀스 다이어그램 모델링 작업을 한다. Implementation 모델 패키지는 소프트웨어 구현 모델을 정의하는 영역으로 구현용 클래스 다이어그램 모델링 작업을 한다. 그리고 모델 변경이 필요한 레거시 코드도 실행 가능한 클래스 모델로 변경한다. MegDefinition 모델 패키지는 이벤트 모델을 정의하는 영역으로 정의된 이벤트 모델들은 액티비티 다이어그램, 시퀀스 다이어그램에서 모델링 작업할 때 활용한다. Domain_Analysis 프로파일은 사업에 정의된 모델들을 정의하는 영역으로 도메인에 필요한 Actor, Stereotype을 정의하여 각 다이어그램 모델링 작업 할 때 활용한다. 자동 모델 생성 도구에서 자동 생성된 모델들은 모델 적용 가이드라인에서 설명된 구조 및 모델들을 모두 생성하였다. Fig. 8은 자동 모델 생성 도구의 실행 결과를 보여주고 Table 1은 자동 모델 생성 도구의 기능에 대해서 간략하게 설명한다.

4.2 자동 생성된 실행 가능한 클래스 모델 기능 시험

자동 모델 생성 도구는 코드 중심의 소프트웨어를 재사용

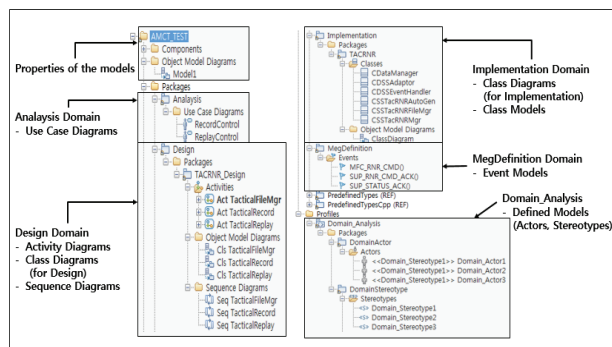


Fig. 8. Execution Result of Automatic Model Creation Tool

Table 1. Features of the Automatic Model Creation Tool

Features	Description
Model Structure	Convert the model structure information defined in the "Project Name" into a model.
Class Model	Parse the code information in the CSCI-CSC-CSU structure and convert it into a model.
Event Model	Parse the event information in the CSCI-CSC-CSU structure and convert it into a model.
Defined Model	Convert the actor and stereotypes information defined in the "Project Name" into a model.

할 경우 실행 가능한 클래스 모델로 자동 변경하는 기능을 제공한다. 모델 중심의 소프트웨어 개발을 하기 위해 모델링 도구를 활용하지만 재사용이 가능한 코드를 설계용 클래스 모델로만 변경할 뿐 실행 가능한 클래스 모델로 변경하는 기능은 존재하지 않는다. 설계용 클래스 모델은 실행을 고려하지 않고 추상적으로 표현한 모델이고 실행 가능한 클래스 모델은 수정 없이 실행 가능한 코드를 생성하는 모델이다. 이러한 이유로 기존 레거시 코드를 활용하려면 수동으로 실행 가능한 클래스 모델로 변경해야 하기 때문에 개발시간 증가의 문제가 발생한다. Fig. 9는 자동 모델 생성 도구에서 자동 생성된 클래스 모델이 실행 가능한 클래스 모델을 생성 했는지 확인하는 시험 환경을 나타낸다. 시험방법은 전술정보 기록 및 재생 소프트웨어에서 재사용이 가능한 레거시 코드를 입력으로 활용하여 자동 모델 생성 도구를 통해 클래스 모델을 생성한다. 그리고 수정 없이 실행 가능한 코드를 생성하여 시험 환경에 탑재하여 기능 시험을 하였다. 기능 시험 결과 기능이 정상 동작하는 것을 확인하였다. Table 2는 자동 모델 생성 도구에서 생성된 클래스 모델에 대한 기능 시험 결과를 나타낸다. 전술 정보 기록및재생에 대한 기능에 대해서 모두 시험하였고 시험 결과 모든 기능을 만족하였다.

4.3 자동 모델 생성 도구에 대한 성능 평가

모델링 작업에 필요한 UML 모델들을 수동으로 생성할 경우, 개발자들의 모델 이해도 및 모델링 도구의 사용 능력에 따

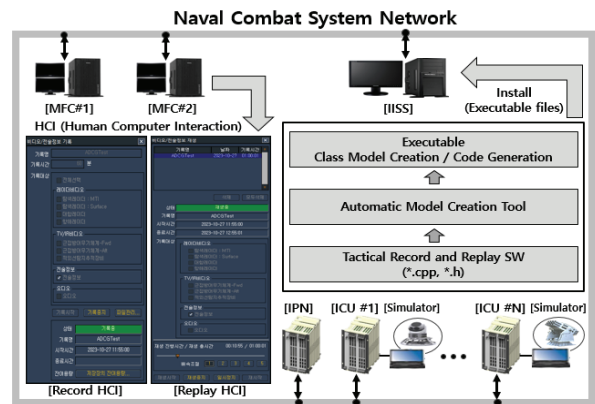


Fig. 9. Test Environment

Table 2. Test results for key functions

Function	Result
Tactical Recording (Recording Start, Recording Stop)	Satisfaction
Tactical Replaying (Replaying Start, Replaying Pause, Replaying Offset, Replaying Restart, Replaying Stop)	Satisfaction
Tactical File Management (File Creation, File Delete, File List)	Satisfaction
Tactical Status Management (Offline, Standby, Recording, Replaying, Pause)	Satisfaction

라 모델링 작업 시간이 증가한다. 특히, 설계용 모델이 아닌 실행이 필요한 모델을 수동으로 생성할 때는 잘못 적용으로 인한 오류를 해결하는데 추가적인 시간이 소요된다. 또한, 개발자들이 담당하는 소프트웨어 구조의 복잡도에 따라서 이러한 시간은 더욱 증가된다. 하지만 자동 모델 생성 도구는 모델링 작업에 필요한 UML 모델들을 자동으로 생성하는 기능을 가지고 있다. 모델 생성에 대한 자동화 기능을 활용하여 모델 생성 작업 시간에 대한 증가 요인 없이 일정하게 모델을 생성할 수 있다. 이로 인해 전체적인 모델 생성 작업 시간을 감소시키는 장점을 가진다. 모델 생성 작업 시간의 감소는 결국 전체 소프트웨어 개발 시간을 감소시키는 효율을 가지기 때문에 중요한 요소이다. 모델 생성 작업 시간은 모델링 작업을 위해 필요한 UML 모델들을 생성하기 위한 작업 시간이다. 모델 생성 작업 시간에는 모델 구조 생성 시간, 실행 가능한 클래스 모델 생성 시간, 이벤트 모델 생성 시간, 정의 모델 생성 시간을 포함한다. 또한, 개발자들의 모델 생성 능력과 담당 소프트웨어 구조의 복잡도에 따른 가중치와 소프트웨어 노드 수도 고려한다. 수동으로 모델을 생성할 경우 모델 생성 가중치는 개발자들의 모델 이해도, 모델링 도구의 사용 능력, 담당 소프트웨어 구조의 복잡도에 따라 증가 하지만 자동 모델 생성 도구는 자동으로 모델을 생성하기 때문에 가중치에 따른 증가 요인은 없다. 모델 생성 가중치는 이벤트 모델 생성 시간과 정의 모델 생성 시간은 공통으로 사용하는 모델이기 때문에 미포함 되었지만 모델 구조 생성 시간과 실행 가능한 클래스 모델 생성 시간에는 포함하였다. 소프트웨어 노드 수는 기능을 수행하는 소프트웨어의 개수로 모델을 적용하는 소프트웨어 개수가 증가할 때마다 모델 생성 작업 시간이 증가하므로 시그마로 표시하였다. 모델 생성 작업 시간 계산식은 Equation (1)과 같다.

$$MCWTime = T_E + T_D + \sum_{k=1}^N \alpha(T_S + T_C) \tag{1}$$

(Model Creation Working Time)

N : Software Node Number α : Model Creation Weights
 T_S : Structure Model Creation Time T_C : Class Model Creation Time
 T_E : Event Model Creation Time T_D : Define Model Creation Time

함정전투체계 소프트웨어를 구성하는 기능 중 체계지원 일부 기능에 자동 모델 생성 도구를 적용하면서 성능을 평가하였다. 체계지원은 함정전투체계의 정보들을 쉽게 분석하기 위한 지원을 담당하는 소프트웨어들로 구성되어 있다. 체계지원을 구성하는 소프트웨어 중 기록 및 재생을 담당하는 4개의 소프트웨어 대상으로 시험하였다. RNRCtrl은 기록 및 재생 통제 기능을 담당하며 TacRNR은 전송정보를 기록 및 재생하는 기능을 담당한다. RecordCtrl은 기록을 하기 위한 운용자 인터페이스 통제 기능을 담당하고 ReplayCtrl은 재생을 하기 위한 운용자 인터페이스 통제 기능을 담당한다. 위 소프트웨어 대상으로 모델링 작업하는데 필요한 모델 구조 생성과 UML 모델 생성에 대해 수동으로 작업 하는 시간과 자동 모델 생성 도구를 활용하여 작업하는 시간을 비교 하였다. Table 3

Table 3. Comparison of Model Creation Working Time

Software	Number of Classes / Lines	Model Creation Working Time (sec)	
		MANUAL	AMCT
RNRCtrl	13 / 16,150	3,600	600
TacRNR	6 / 36,223	7,200	600
RecordCtrl	6 / 9,033	1,800	600
ReplayCtrl	4 / 6,843	1,800	600

은 소프트웨어 클래스 수, 라인 수 및 수동 모델 생성 방식과 자동 모델 생성 도구를 활용한 방식에 대한 모델 생성 작업 시간을 비교한 표이다. 시간을 비교한 결과 자동 모델 생성 도구를 활용한 방식이 수동으로 작업하는 방식보다 모델 생성 작업 시간 감소의 효율화를 보였다. 자동 모델 생성 도구는 자동으로 모델을 생성하는 기능 때문에 일정한 시간대로 측정 되었지만 수동 모델 생성 방식은 클래스 수, 라인 수 및 소프트웨어 복잡도에 따라 시간이 달리 측정되었다. 이러한 결과를 참고 했을 때 자동 모델 생성 도구를 적용하는 소프트웨어 개수가 많을수록 모델 생성 작업 시간 감소의 효율화는 더욱 증가한다.

5. 결 론

본 논문에서는 함정 전투체계 소프트웨어 개발에 UML 모델을 구조적으로 적용하기 위한 자동 모델 생성 도구 개발 방안을 제안하였다. 자동 모델 생성 도구는 모델에 대한 기본적인 정보들을 입력으로 활용하여 모델링 작업에 필요한 모델 구조, 실행 가능한 클래스 모델, 이벤트 모델, 정의 모델을 자동 생성하는 기능을 수행한다. 그리고 개발된 자동 모델 생성 도구를 함정 전투체계 소프트웨어 중 체계지원 일부 기능에 적용하여 기능과 성능을 입증하였다. 소프트웨어 개발에 자동 모델 생성 도구를 활용하면 모델링 작업에 필요한 모델들을 자동으로 생성하여 수동으로 작업하는 것보다 통일성 있게 적용할 수 있다. 또한, 모델링 작업에 필요한 UML 모델들을 자동 생성하는 기능 때문에 수동으로 모델링 작업한 것 보다 모델 생성 작업 시간을 감소시키는 장점도 가진다. 모델 생성 작업 시간의 감소는 모델링 작업에 대한 시간을 감소시키기 때문에 전체 소프트웨어 개발 시간을 감소시키는 효율화를 가진다. 추후 연구과제에서는 도메인 특성에 맞는 더 다양한 UML 모델들을 식별하고 자동 생성하여 함정 전투체계 소프트웨어 개발에 유연하게 적용하기 위한 더 많은 연구가 필요하다.

References

[1] C. K. Jang, "Effect of Collaboration Tools on Stakeholders' Communication in Software Development Project," *The*

Graduate School of Information Yonsei University, 2022.

[2] S. K. Kim and G. S. Ryu, "Research for improving quality of SI(System integration) development project," *The Journal of Korea Institute of Information, Electronics, and Communication Technology*, Vol.11, No.3, pp.215-220, 2018.

[3] K. Lee and T. Kim, "UMIICA: A Model- Driven Integrated Development Environment for Industrial Control Applications," *IEEE Access*, Vol.6, pp.43290-43301, 2018.

[4] G. Martin, "UML for Embedded Systems Specification and Design: Motivation and Overview," *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, pp.773-775, 2002.

[5] D. Tikhonov, D. Schütz, S. Ulewicz and B. Vogel-Heuser, "Towards industrial application of model-driven platform-independent PLC programming using UML," *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pp.2638-2644, 2014.

[6] H. Kim, S. Hong, D. Kwon, J. Kim, and J. Ma, "The Study on Automated Artillery Fire Direction for Future Warfare with UML," *Korean Journal of Computational Design and Engineering*, Vol.23, No.4, pp.394-403, 2018.

[7] D. Apostol, P. Rusovan, and M. Marcu, "UML to code, and code to UML, a view inside implementation challenges and cost," *26th International Conference on System Theory, Control and Computing*, pp.140-145, 2022.

[8] S. M. Kwon and S. M. Jung, "Virtualization based high efficiency naval combat management system design and performance analysis," *Journal of the Korea Society of Computer and Information*, Vol.23, No.11, pp.9-15, 2018.

[9] C. G. Yi and Y. G. Kim, "Security Testing for Naval Ship Combat System Software," *IEEE Access*, Vol.9, pp.66839-66851, 2021.

[10] E. Domínguez, B. Pérez, Á. L. Rubio, and M. A. Zapata, "A Systematic review of code generation proposals from state machine specifications," *Information and Software Technology*, Vol. 54, No.10, pp.1045-1066, 2012

[11] M. K. Shiferaw and A. K. Jena, "Code Generator for Model-Driven Software Development Using UML Models," *2018 Second International Conference on Electronics, Communication and Aerospace Technology*, pp.1671-1678, 2018.

[12] E. V. Sunitha and P. Samuel, "Automatic Code Generation From UML State Chart Diagrams," *IEEE Access*, Vol.7, pp.8591-8608, 2019.

[13] F. J. Lucas, F. Molina, and A. Toval, "A systematic review of UML model consistency management," *Information and Software Technology*, Vol. 51, No. 12, pp. 1631-1645,

2009.

[14] M. Usman and A. Nadeem, "A survey of Consistency Checking Techniques for UML Models," *2008 Advanced Software Engineering and Its Applications*, pp. 57-62, 2008.

[15] S. Gotti and S. Mbarki, "UML executable: A comparative study of UML compilers and interpreters," *2016 International Conference on Information Technology for Organizations Development*, pp.1-5, 2016.

[16] D. Torre, "On validating UML consistency rules," *2015 IEEE International Symposium on Software Reliability Engineering Workshops*, pp.59-60, 2015.

[19] J. Zeng, J. Song, B. Jia, and M. Wang, "A test method of utility system management computer based on rhapsody design model," *CSAA/ IET International Conference on Aircraft Utility Systems*, pp.1212-1216,

[20] S. M. Jung and W. J. Lee, "A Study on Applying a Consistent UML Model to Naval Combat System Software Using Model Verification System," *Journal of the Korea Society of Computer and Information*, Vol.27, No.5, pp.109-116, 2022.

[21] K. S. Song, D. S. Kim, and Y. S. Choi, "A Design of Message Oriented Management and Analysis Tool for Naval Combat Systems", *Journal of the Institute of Electronics and Information Engineers*, Vol.51, No.2, pp.197-204, 2014.



정 승 모

<https://orcid.org/0009-0005-8062-2209>

e-mail : seungmo1025@knu.ac.kr

2010년 한양대학교 전자전기

제어계측공학과(석사)

2021년~현 재 경북대학교 컴퓨터학부

박사과정

2010년~현 재 한화시스템 SW1팀(해양) 전문연구원

관심분야 : Combat System Software, Software Modeling,

Model-Driven Development



이 우 진

<https://orcid.org/0000-0002-8075-5248>

e-mail : woojin@knu.ac.kr

1994년 KAIST 전산학과(공학석사)

1999년 KAIST 전산학과(공학박사)

1999년~2002년 ETRI 선임연구원

2002년~현 재 경북대학교 컴퓨터학부 교수

관심분야 : Software Testing, Software Modeling, Embedded

Systems