

## UML 2.0 기반 객체지향 모델링 프로세스 및 지침

강현구, 천두완, 김수동

승실대학교 대학원 컴퓨터학과

{h9kang, dwcheun}@otlab.ssu.ac.kr, sdkim@ssu.ac.kr

### Process and Guidelines for Object-Oriented Modeling with UML 2.0

Hyun Koo Kang, Du Wan Cheun and Soo Dong Kim

Dept. of Computer Science, Soongsil University

#### 요 약

객체 관련 표준화 기구인 OMG 가 UML 을 산업 표준으로 채택한 이후, UML 은 산업계나 학계에서 널리 사용하게 되었다. 그래서, UML 은 각종 과제나 프로젝트를 수행할 때 청사진을 제공하는 대표적인 모델링 언어로 자리매김하게 되었다. UML 을 사용한 경험이 증가하고, 소프트웨어 모델링의 문제를 알게 됨에 따라, UML 에 대한 새로운 요구 사항 나왔고, 개정판으로 UML 2.0 이 발표되었다. 그러나 UML 2.0 을 사용한 범용 프로세스의 명세가 부족하고, 다이어그램에 대한 지침도 부족하다. 본 논문에서는 UML 2.0 에서 사용할 수 있는 범용 프로세스를 제시하고, 각 프로세스에서 필요한 주요 다이어그램에 대한 핵심 지침을 제안하였다.

#### 1. 서론

1997년에 객체 관련 표준화 기구인 OMG가 UML을 산업 표준으로 채택한 이후, UML은 산업계와 학계에서 널리 사용하게 되었다 [1]. 그래서, UML은 각종 과제나 프로젝트를 수행할 때 청사진을 제공하는 대표적인 모델링 언어로 자리매김하게 되었다. UML 사용이 증가하고, 소프트웨어 모델링의 문제를 알게 됨에 따라, UML에 대한 새로운 요구 사항 나왔고, 개정판으로 UML 2.0이 발표되었다 [2][3][4][5]. 그럼에도 불구하고 시스템을 분석하고 설계할 때 아직까지 UML 1.4에 의존해서 모델링을 하고 있는 실정이다. 왜냐하면 UML 2.0을 사용한 범용 프로세스의 설명이 부족하여 전체 과정을 수행하기 힘들고 다이어그램에 대한 모델링 지침이 부족하여, 사용자가 모델링 한 산출물 간의 차이가 발생하기 때문이다.

따라서 이 논문에서는 UML 2.0을 기반으로 시스템을 개발하기 위한 범용 프로세스를 제시하고, 각 프로세스에서 사용되는 주요 다이어그램에 대한 핵심 지침을 제안한다.

본 논문에서의 구성은 다음과 같다. 다음 장에서는 UML 2.0에 관한 간단한 소개를 하고, UML 2.0용 범용 프로세스, 주요 다이어그램에 대한 핵심 지침을 3장과 4장에서 설명한다. 마지막으로 5장에서 결론 및 향후 연구 과제를 제시한다.

#### 2. 기반 연구

UML 1.4를 기반으로 한 객체지향 프로세스는 다음과 같다. 먼저 요구사항을 명세하고, 유즈 케이스 모델링(Use case Modeling), 개념적인 객체 모델링(Conceptual Object Modeling), 개념적인 동적 모델링(Conceptual Dynamic Modeling), 상세 객체 모델링(Detailed Object Modeling), 상세 동적 모델링(Detailed Dynamic Modeling), 사용자 인터페이스

설계, 코딩, 문서화, 유지 및 보수를 함으로써 전체 프로세스를 완료한다 [6].

위에서 언급한 프로세스에서 쓰일 수 있는 UML 1.4 기반의 다이어그램은 다음과 같다. 유즈 케이스 모델링에서는 유즈 케이스 다이어그램과 유즈 케이스 명세(Use case Description)를 이용하여 각 기능을 정의 한다. 객체 모델링에서는 도메인의 객체를 찾아내고 객체간의 관계를 맺는 클래스 다이어그램을 그린다. 동적 모델링은 유즈 케이스 다이어그램과 유즈 케이스 명세를 이용하여 시퀀스 다이어그램을 그린다 [6][7].

UML 2.0을 기반으로 한 프로세스에서 사용하는 다이어그램은 다음과 같다. 요구 분석 단계에서는 시스템의 기능을 알기 위한 유즈 케이스 다이어그램, 도메인의 객체들을 개념적으로 표현한 클래스 다이어그램, 사용자와 시스템 간의 상호활동을 표현한 활동 다이어그램, 특별한 생명주기에 유용한 상태 다이어그램을 사용한다. 설계 단계에서는 클래스의 내부적인 연결을 보기 위한 클래스 다이어그램, 클래스 간의 메시지 전달을 표기한 시퀀스 다이어그램, 소프트웨어의 구조를 큰 단위로 나누는 패키지 다이어그램, 클래스의 복잡한 생명 기록을 알아내기 위한 상태 다이어그램, 소프트웨어의 물리적인 레이아웃을 그리기 위해 배치도를 사용한다. 그 외에 정적 모델링을 하는 컴포넌트 다이어그램(Component Diagram), 복합 구조 다이어그램(Composite Structure Diagram), 기능적 모델링을 하는 활동 다이어그램(Activity Diagram), 동적 모델링을 하는 의사소통 다이어그램 (Communication Diagram), 상호연동 개요 다이어그램 (Interaction Overview Diagram), 타이밍 다이어그램(Timing Diagram)가 있다 [8].

UML1.4와 UML2.0은 객체 지향 프로세스에서 사용하는 다이어그램을 제시하지만 전체 프로세스가 아닌 일부 프로세스에 대해 언급하고 있고, 프로세스와 관련된 다이어그램의 용도에 대해서 언급하고 있다.

### 3. UML 2.0 범용 프로세스

본 절에서는 UML 2.0을 사용한 범용 프로세스는 그림 1에 서와 같이 9가지 단계로 제안한다 [9].

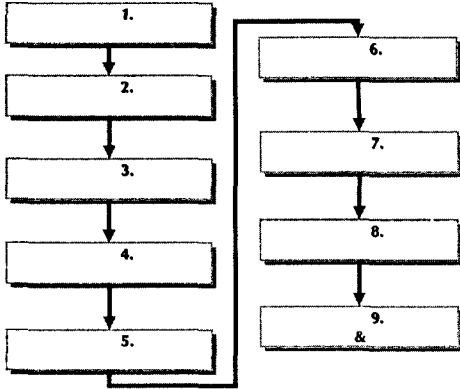


그림 1. UML 2.0 범용 프로세스

요구 사항 정의에서는 개발할 시스템의 요구 사항을 기능적 요구사항과 비기능적 요구사항으로 나누고, 개발할 범위를 정의한다.

기능적 모델링에서는 사용자가 어떤 기능을 원하는지 분석한다. 객체 모델링에서는 도메인에 있는 객체를 찾아내고, 그 객체들간의 관계를 구조도로 표현한다. 동적 모델링에서는 기능적 모델링에서 파악한 기능의 수행 방법을 설계 한다.

아키텍처 설계에서는 시스템을 구성하는 부분들과 그 부분들간의 관계, 구조를 설계한다. 데이터 베이스 설계에서는 객체 모델링에서 나온 산출물인 클래스들을 테이블에 매핑시킨다. 이때 관계 형 테이블, 객체 지향형 테이블, 객체 관계 형 테이블 등이 사용된다. 사용자 인터페이스 설계에서는 시스템 기능별 흐름에 대해 설계 하는 것으로 구현 단계에서 사용자 인터페이스 구현에 도움이 되며, 고객에 의한 시스템 기능의 흐름을 확인할 수 있다. 구현에서는 전 단계에서 나온 산출물을 이용하여 고객이 원하는 플랫폼에 맞게 실제 코드를 생성한다. 통합, 시험에서는 사용자 인터페이스 계층, 논리 계층, 데이터 계층의 클래스를 결합하고, 그 한 결합한 결과물의 오류를 찾아내서 수정하는 단계이다.

위에서 언급한 각 단계에서는 UML 2.0이 제공하는 다이어그램을 사용할 수 있다. 표 1에서는 9가지 단계 중에서 다이어그램이 사용되는 단계 2,3,4,5에 대해서 정리한다.

표 1. 각 단계 별 사용 다이어그램

단계	다이어그램
기능적 모델링	유즈 케이스, 활동 다이어그램
객체 모델링	클래스 다이어그램
동적 모델링	시퀀스 다이어그램, 의사소통도, 상호연동개요 다이어그램
아키텍처 디자인	컴포넌트 다이어그램, 배치 다이어그램, 패키지 다이어그램, 복합 구조 다이어그램

### 4. 주요 다이어그램에 대한 핵심 지침

이번 장에서는 각 단계에서 사용되는 주요 다이어그램에 대한 핵심 지침에 대해서 서술하려고 한다. 앞에서 제시한 9가지 단계가 모두 중요하지만, UML 2.0에서 추가되거나 수정된 다이어그램에 대한 내용을 언급하는 논문이므로, 다이어그램이 많이 사용되는 단계 2,3,4에 대해서, 그리고 UML 1.4와 다른 부분에 대해서만 중점적으로 서술한다.

#### 4.1. 기능적 모델링

기능적 모델링을 하기 위해서 유즈 케이스 다이어그램을 사용하고, 선택적으로 활동 다이어그램을 사용한다.

유즈 케이스의 경우, UML 1.4와 크게 달라진 것이 없으므로 이전에 제시된 지침을 따른다.

활동 다이어그램의 경우, 활동(Activity)에 대한 개념의 변화와 모델링에 있어서 새로운 요소가 추가되었기 때문에 활동 다이어그램에 대한 지침을 제안한다.

- UML 1.4에서 활동 안에 다른 활동이 포함으로 표현되었던 것(Nested Activity)을 UML 2.0의 확장 영역(Expansion Region)을 이용하여 표현하고, 키워드(Keyword)와 핀(Pin)을 이용하여 입력, 출력 값에 대해 상세하게 기술한다.
- 활동간의 연결에서 시그니처(Signature)나 인자(Parameter)가 필요할 경우 핀(Pin)을 활동에 추가한다 [2].
- 인터넷 쇼핑몰에서 정기적으로 확인해야 하는 회원의 사 이버 머니 와 같이 기능적 요구사항에서 경제적인 문제와 관련되어 중요하게 취급해야 하는 시간적 제약이나, 주식 정보 시스템에서 데이터 갱신이 3 초안에 수행해야 하는 것과 같이 비기능적 요구사항에서 언급되는 시간적 제약은 시간 신호(Time Signal)을 이용하여 명세 한다.
- 간단한 수신 신호 혹은 전송 신호가 있을 경우에는 UML 2.0에서 제공하는 수신 신호(Accept Signal), 전송 신호(Send Signal) 표기법을 사용하고, 신호처리에 관한 것이라도 여러 가지 절차가 있을 경우 활동으로 표현한다.

#### 4.2. 객체 모델링

객체 모델링에서 이용되는 주요 다이어그램을 이용하여 지침을 제시한다. 객체 모델링을 하기 위해서 사용하는 가장 일반적인 방법으로 클래스 다이어그램을 이용한다

UML 2.0에서 클래스 다이어그램은 크게 변한 것은 없다. 그래서 대부분은 UML 1.4에서 정의된 지침을 따르면 된다. 그러나, 관계에서 다중성은 지속적인 클래스간 관계가 영구적인 경우에 표현한다. 또한 의존(Dependency)관계에서 스테레오 타입(Stereo Type)으로 표현할 때 일시적인 연결의 뜻만 표현한다면 표시하지 않는다.

#### 4.3. 동적 모델링

MDA 기법에서 모델 변환(Model Transformation)을 용이하게 위해서 UML 2.0의 교류 다이어그램(Interaction Diagram) 부분에서 실행을 위한 상호 작용을 표현하기 위한 구성 요소(Constructor)가 다양하게 제공되었기 때문에 이번 장에서 상세하게 서술한다. UML 1.4에서 교류 다이어그램에는 시퀀스 다이어그램, 협동 다이어그램(Collaboration Diagram)이 있다. UML 2.0에서는 협동 다이어그램에서 이름이 변경된

의사소통 다이어그램, 시퀀스 다이어그램, 새롭게 추가된 상호연동 개요 다이어그램과 타이밍 다이어그램이 있다. 이 중에서 시퀀스 다이어그램과 상호연동 개요 다이어그램의 지침에 대해서만 설명한다.

UML 2.0으로 오면서 시퀀스 다이어그램에서 크게 달라진 것은 while 문이나 if-else 문, switch-case 문과 같은 반복 문, 조건 문, 선택 문 등으로 나타나는 것은 상호연동 프레임(Interaction Frame)으로 표현한다는 점이다. 상호연동 프레임의 범위와 크기 결정에 대한 지침을 다음과 같이 제안한다.

- 유즈 케이스 명세(Use case Description)에서 행위자(Actor)와 시스템 간의 메시지 전달(Message Passing)관계를 기준으로 상호연동 프레임의 후보를 결정한다.
- 후보 상호연동 프레임에서 일어나는 수행이 반복(Loop)이나 분기(Branch)가 없고 순차적으로 일어나는 경우 후보 상호연동 프레임에서 제거한다.
- 후보 상호연동 프레임에서 일어나는 수행이 반복적, 대안(Alternative)이나 선택적(Optional), 병렬적(Parallel)으로 일어나는 수행이지만 수행하는 행동(Operation)이 모두 합해서 두 개 이하일 경우에 UML 1.4 에서 사용했던 표기법을 사용한다.
- 위에 지침에 따라서 결정된 상호연동 프레임에서 수행되는 행동이 모두 합해서 10 개 이상일 경우, 새로운 다이어그램을 만들어서 상호연동 프레임 안에서 수행되어야 할 행동에 대해 정의하고, 상호연동 프레임이 사용된 다이어그램에서는 'ref' 연산자(Operator)를 이용하여 새로운 다이어그램을 참조한다.

상호연동 개요 다이어그램(Interaction Overview Diagram)은 UML2.0에서 새롭게 추가된 다이어그램이다. 이 다이어그램은 활동 다이어그램과 시퀀스 다이어그램의 복합 다이어그램으로 표현되기 때문에 위에서 두 다이어그램에 대해 언급한 지침을 기준으로 그린다. 그렇지만 이 다이어그램에서 시퀀스 다이어그램의 기준은 유즈 케이스가 아니라 활동이기 때문에 활동을 기준으로 시퀀스 다이어그램을 그리는 지침이 필요하다.

- 만일 활동이 진입 동작, 탈출 동작 그리고 하나의 행동으로 이루어져 있다면, 시퀀스 다이어그램으로 표현하지 않는다.
- 활동에서의 진입 동작은 시퀀스 다이어그램에서 처음으로 발생하는 메시지 전달로 표현한다.
- 활동에서의 탈출 동작은 시퀀스 다이어그램에서 마지막으로 수행되는 메시지 전달로 표현한다.
- 활동 다이어그램에서 편을 사용해서 표현된 시그니처 인자는 시퀀스 다이어그램의 메시지 전달에서 처음 수행에서의 인자나 마지막 수행에서의 리턴 값(Return value)로 표현된다.
- 활동 다이어그램에서 분기는 시퀀스 다이어그램의 상호연동 프레임을 이용해서 표현한다. 분기는 대안이나 선택적으로 일어나는 수행이므로, 상호연동 프레임의 연산자는 'alt'를 사용하고, 활동 다이어그램의 분기에서 표현된 전이식은 가드(Guard)를 사용하여 표현한다.
- 활동 다이어그램에서 분할(Fork)과 합류(Join)는 시퀀스 다이어그램의 상호연동 프레임을 이용해서 표현한다. 분할이나 합류는 병렬적으로 일어나는 수행이므로, 상호연동 프레임의 연산자는 'par'를 사용한다.

## 5. 결론 및 향후 연구과제

본 논문에서는 UML 2.0을 기반으로 한 범용 프로세스를 제안하여, 시스템 개발 시 사용자가 원하는 시스템을 UML 2.0을 통해 분석, 설계 하고자 했다. 그리고 각 단계에서 사용되는 주요 다이어그램의 핵심 지침을 제안하여, 여러 명의 사용자가 같은 프로젝트를 수행할 때, UML 2.0의 주요 다이어그램을 설계해서 나오는 산출물 간의 일관성을 유지할 수 있다. 지침을 통해 다이어그램 간의 관점의 차이를 줄일 수 있으며, 개발 기간을 단축하고 품질을 높이는 개발을 할 수 있다. 그리고, 사용자에게 UML 2.0을 사용상 보다 구체적인 지침을 제공하기 때문에 효율적이고 능률적인 설계가 가능하게 된다.

앞으로 수정된 모든 다이어그램에 대해 모델링 지침을 제안하고, 그 다이어그램 간의 일관성을 맞추도록 해야 할 것이다. 그러기 위해서는 모델링 지침의 항목을 좀더 구체화하고 정형화 해야 한다. 마지막으로 UML 2.0으로 모델링할 수 있는 툴과 모델링 한 산출물을 통해 MDA를 구현할 수 있는 자동화 도구나 소프트웨어 개발이 필요할 것이다.

## 6. 참고문헌

- [1] J. Rumbaugh, et al., *The Unified Modeling Language Reference Manual*, Addison-Wesley-Longman, Reading, Mass., 1998.
- [2] OMG, "UML 2.0 Infrastructure – Final Adopted Specification", OMG document ptc/03-09-15, Available at: <http://www.omg.org/cgi-bin/doc?ptc/2003-09-15>. September 2003.
- [3] OMG, "UML 2.0 Superstructure – Final Adopted Specification", OMG document ptc/03-08-02, Available at: <http://www.omg.org/cgi-bin/doc?ptc/2003-08-02>. August 2003.
- [4] OMG, "UML 2.0 OCL – 2nd Revised Submission", OMG document ptc/03-01-07, Available at: <http://www.omg.org/cgi-bin/doc?ad/2003-01-07>. January 2003..
- [5] OMG, "UML 2.0 Diagram Interchange – Draft Adopted Specification", OMG document ptc/03-07-03, Available at: <http://www.omg.org/cgi-bin/doc?ptc/2003-07-03>. July 2003..
- [6] 김수동, 실무자를 위한 소프트웨어 공학, 에드텍, 1994
- [7] Fowler M, Scott K, *UML Distilled Second Edition*, Addison Wesley, 2000..
- [8] Fowler M, *UML Distilled Third Edition*, Addison Wesley, 2004.
- [9] Object Technology Laboratory, *GREEN Methodology Specification, Technical Report*, 2001.