

# 객체 지향 모델에서 일관성 검증을 위한 관계성 분석 도구 설계 및 구현

이소영\*, 조영석\*, 허계범, 최영근  
광운대학교 컴퓨터 과학과

## Design and Implementation of a Relationship Analysis Tool for Consistent Verification within Object-Oriented Model

So-Young Lee, Young-Seok Cho, Kwae-Bum Heo, Young-Keun Choi  
Dept. of Computer Science Kwangwoon University

### 요 약

본 논문에서는 객체 지향 소프트웨어 개발 생명주기에서 생성된 객체 모델들간의 일관성 유지를 위한 관계성 분석 도구 설계 방법을 제시한다. 이 방법에서는 객체 지향 모델의 산출물들간의 관계성을 분석하여 소프트웨어가 개발되고 동시에 부목화 되어 저장된다. 그리고 소프트웨어 유지 보수 및 재사용이 가능한 검색 시스템으로 되어 있으며, 객체 지향 CASE 시스템의 정보 저장소 기능을 부분적으로 지원한다. 따라서 시스템 분석기 및 설계자에게 유용한 지침을 제공하고, 개발자에게 산출물들간의 일관성 검증을 쉽게 하도록 하여 소프트웨어 재사용 및 유지보수가 용이하다.

### 1. 서론

객체 지향 기술은 자인스러운 모델링을 통해 분석 및 설계과정에서 효과적으로 사용되어지며, 소프트웨어 구현이 편리하고 재사용이 뛰어나 유지보수가 쉬운 장점이 있다. Bach, Rumbaugh, Jacobson의 방법론을 통합한 객체 지향 표준 방법론인 UML(Unified Modeling Language)은 기존의 방법들을 보다 확장된 기법으로 명료하고 일관된 높은 표현력을 제공하고 있다. 그리고 객체 지향 시스템에서 CASE 도구의 중요성은 날로 증기하고 있으며, 현재 상업적인 CASE 들이 많이 발표되고 있다[1][2]

그러나 UML은 소프트웨어 개발을 위한 프로세스를 정의하고 있지 못하며, 소프트웨어 구조를 쉽게 표현하는 명세서(Specification) 정보가 부족한 실정이다. 객체 지향 CASE 시스템들은 개발적인 저장 및 추출기능은 가지고 있으나, 분석 단계에서부터 일관성 있는 개발을 사용하며 실행 결과를 통한 반복 작업과, 시스템 분석가 및 설계자에게 유용한 정보를 제공할 수 있는 산출물들간의 일관성 검증을 위한 관계성 정보를 제공하지 못한다.

따라서 본 논문에서는 객체 지향 소프트웨어 개발 생명주기에서 명세서들을 제시하고, 생성된 객체 모델들간의 일관성 유지를 위한 관계성 분석 도구 설계 방법을 제시하여 시스템 분석가 및 설계자에게 유용한 지침을 제공하고, 개발자에게 일관성 검증은 쉽게 하여 소프트웨어 개발 및 유지보수가 용이하도록 한다.

### 2. 관련연구

#### 2.1 객체 지향 CASE TOOL 시스템

객체지향 시스템에서 CASE 도구의 중요성은 날로 증가하고 있다. 현재 객체 지향 기법을 위한 CASE 도구로는 Rational Rose, OMTool, Select OMT등이 대표적이다. 그러나 현재 CASE 시스템의 정보 시각소에서 지원되는 주요 기능은 클래스 객체의 개별적인 저장 및 추출 기능, 상속, 복합, 참조 관계의 인스턴스를 저장하는 기능에 불과하여 아직 불충분한 상태이다[4][5]. 그리고 시스템 분석가 및 설계자에게 유용한 정보를 제공할 수 있는 산출물들간의 일관성 검증을 위한 관계성 정보를 제공하지 못한다.

#### 2.2 UML

UML(Unified Modeling Language)은 기존의 방법들을 보다 확장된 기법으로 명료하고 일관된 높은 표현력을 제공하고 있으며, 분석에서 설계, 구현으로의 자연스러운 대응(mapping), 그리고 사용자외 개발자간의 의사소통이 원활한 장점이 있다. 그러나 UML은 소프트웨어 개발을 위한 프로세스를 정의하고 있지 못하며, 소프트웨어 구조를 쉽게 표현하는 명세서(Specification) 정보가 부족한 실정이다.

### 3. 클래스 관계성 분석 및 설계 방법

#### 3.1 시스템 개요

이 절에서는 그림 1에 제시되어 있는 것처럼 객체지향 소프트웨어를 개발하기 위한 단계는 요구분석단계, 분석단계, 설계단계, 구현단계, 테스트단계로 나누어지며 각 단계는 점진적이며 반복적으로 수행된다.

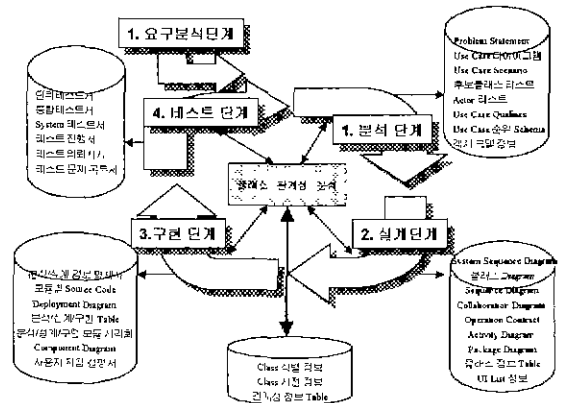


그림 1. 프로세스 개발 과정의 관계성 저장 정보

각 단계별의 업무를 정의하면 다음과 같다. 요구분석단계는 구축하고자 하는 시스템이 어떤 시스템이며 타당성 여부를 파악하여 시스템구축을 위한 사전 조사를 하게 된다. 분석 단계는 시스템의 문제정의를 작성하고 여기에서 Use Case를 식별

하고 식별된 Use Case를 세분화하며, Actor를 추출하는 작업, Use Case 시나리오 작성, System Sequence Diagram 작성하는 등의 작업을 하게 된다. 설계단계는 Use Case 시나리오로부터 Class를 식별하고, Sequence Diagram 작성을 통한 Operation 추출, Class Relationship 작성 등을 하게 된다. 구현단계는 분석/설계/구현 모듈의 Table화 작업, 각 모듈별 Diagram 편집등을 표현하게 된다. UML 다이어그램과 명세서와 클래스간의 기각 관계성을 나타낸다.

본 논문에서는 객체지향 시스템의 생명주기에서 UML의 산출물들과 클래스 관계성 정보 설계 방법을 제시한다. 이러한 관계성 정보의 테이블화의 시의적절은 시스템을 관리하기 위한 좋은 수단. 업무간의 통신 수단, 교육과 참고 문헌으로의 역할, 품질 보증과 유지보수의 개발 감리 강화를 제공, 시스템 개발 방법 및 순서의 표준화가 되며, 재사용에 유용하게 이용할 수 있는 등 그의 많은 장점을 갖고 있다.

3.2 검증 방법

이 절에서는 클래스 사전 정보, UML 산출물들간의 관계성, 클래스와 UML 산출물의 관계성을 테이블화 하였다.

첫째, 객체지향 프로그램에서의 프로그램 이해단위인 클래스의 정보를 사전화하여 표 1과 같은 테이블로 작성한다.

둘째, 다이어그램과 명세서의 대한 관계성 정보로 저장하기 위해서 기준정보를 설정하고 관계정보들간의 관계와 관계성과 분석, 설계, 구현정보를 나타낼 수 있도록 테이블화 시킨다.

셋째 표 2와 같이 산출물들간의 관계가 강한 관계인지, 약한 관계를 분석하여 정보를 테이블화 하여 저장하고, 작성된 내용을 가지고 자동적으로 관계성 검증을 할 수 있도록 그래프형식으로 나타내 준다.

넷째, 그림 5와 같이 분석,설계,구현에 대해서 재사용 및 유지보수를 위해서는 모든 다이어그램과 명세서들의 정보를 시각화하여 보여 주기 위해서 트리 구조로 나타내 준다.

다섯째 표 3과 같이 클래스와 산출물들간의 관계가 있는지, 클래스

와 클래스간의 관계는 잡조관계인지 상속관계인지 집합관계인지의 정보를 테이블화 하여 저장한다

여섯째, 표 4와 같이 클래스와 산출물, 클래스와 클래스간의 유사성을 표현하였다. 클래스의 산출물간의 관계성 표현은 클래스 C#1을 산출물에서 사용하면 1, 그렇지 않으면 0의 값으로 표현하고, 클래스들간의 유사성 측정은 다음의 수식을 따른다[4].

클래스 데이터 멤버의 유사성 측정은  $SIM_{dm}(A, B) = (dm(A/B))^2 / dm(A) \times dm(B)$ 로 측정하고  $dm(A)$ 는 클래스 A를 구성하는 데이터 멤버의 개수를 나타낸다. 클래스 함수 멤버 유사성은  $SIM_{fm}(A, B) = (dmf(A/B))^2 / dmf(A) \times dmf(B)$ 로 측정한다. 클래스들간의 유사성 측정은  $SIM(A, B) = P \times SIM_{dm}(A, B) + (1-P) \times SIM_{dmf}(A, B)$ 로 측정하며  $dmf$ 는 델타함수를 나타낸다.

본 논문에서 도표의 표기는 다음과 같다. 요구사항명세서(RD), 시스템구축계획서(SDP), 프로젝트개발계획서(PDP), Problem Statement(PS), UseCase 다이어그램(UCD), Use Case Scenarios(UCS), Flow of Event(FOE), UseCase

표 1 클래스 관계성 사전 정보

필드 이름	데이터 형	설명
ID	일련번호	
ClassName	문자열	클래스 명
stereotype	문자열	스태레오 타입
Attribute	문자열	속성
Method	문자열	메소드
BaseClass	문자열	상속받은 클래스
Link	문자열	참조 클래스
Type	문자열	속성요형, 메소드 리턴형
IntValue	중수	초기값
ParameterNo	중수	파라미터수
ReferCNT	중수	참조수
FNTOverload	문자열	함수중복
FNTVirtual	문자열	가상함수
Friend	문자열	프렌드 선언
Diagram	문자열	관련된 다이어그램
Spec	문자열	관련된 명세서
memo	문자열	비고

표 2 UML 산출물들간의 관계

산출물명	UML 산출물										클래스										클래스간의 관계성									
	UCD	UCS	FOE	UCLL	ACFI	ASL	SDP	SDP	SDP	SDP	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9	Class10	Class11	Class12	Class13	Class14	Class15	Class16	Class17	Class18	Class19	Class20
UCD	A																													
UCS		A																												
FOE			A																											
UCLL				A																										
ACFI					A																									
ASL						A																								
SDP							A																							
Class1								A																						
Class2									A																					
Class3										A																				
Class4											A																			
Class5												A																		
Class6													A																	
Class7														A																
Class8															A															
Class9																A														
Class10																	A													
Class11																		A												
Class12																			A											
Class13																				A										
Class14																					A									
Class15																						A								
Class16																							A							
Class17																								A						
Class18																									A					
Class19																										A				
Class20																											A			

표 3 UML 산출물과 클래스, 클래스의 클래스간의 관계성

클래스명	UML 산출물										클래스										클래스간의 관계성									
	UCD	UCS	FOE	UCLL	ACFI	ASL	SDP	SDP	SDP	SDP	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9	Class10	Class11	Class12	Class13	Class14	Class15	Class16	Class17	Class18	Class19	Class20
Class1	A																													
Class2		A																												
Class3			A																											
Class4				A																										
Class5					A																									
Class6						A																								
Class7							A																							
Class8								A																						
Class9									A																					
Class10										A																				
Class11											A																			
Class12												A																		
Class13													A																	
Class14														A																
Class15															A															
Class16																A														
Class17																	A													
Class18																		A												
Class19																			A											
Class20																				A										

