객체지향 분석 모델의 일관성 및 오류 점검 기법에 관한 연구

한미정[1] 김유경 박재년
숙 명 여 자 대 학 교 컴퓨터과학
{mjhan, ykkim, jnpark}@cs.sookmyung.ac.kr

Checking consistency and correctness in object-oriented analysis model

Mi-jung Han, Yu-kyung Kim, Jae-nyun Park
Dept. of Computer Science, Sook-Myung Women's University

요 약

분석 모델은 개발자의 언어로 작성되어 정확성과 일관성이 있는 개발 산출물로 관리되어야 한다. 본 논문은 이를 지원하기 위해 정형명세를 위한 New FST를 정의하고, 분석 모델의 정확성과 일관성 검증 규칙을 제안하며, 사례 연구를 통하여 적용하여 이 규칙들을 시험하였다.

1. 서론

시스템 개발 초기에 수행되는 분석은 소프트웨어 생명 주기의 모든 단계에 영향을 미치는 중요한 단계이다. 초기에 개발된 산출물들이 정확하지 못할 경우, 이를 수정하기 위한 비용이 설계와 구현, 시험 단계 등의 후기 개발 단계로 진행될수록 급격히 증가된다는 사실을 고려할 때, 분석 단계에서 정확한 산출물을 개발하는 것은 매우 중요하다. 표준 객체지향 방법론인 UML(Unified Modeling Language)은 생명주기의 초기인 정련 단계에서 요구사항 명세와 분석 등을 수행하여 쓰임새 모델과 분석 모델을 개발한다. 쓰임새 모델은 고객의 언어로 기술되어 중복되거나 쓰임새 모델의 내부 요소들 사이에 일관성이 부정확할 수도 있다. 반면 분석 모델은 사용자가 아닌 개발자의 언어로 작성되어 중복성이 없어야 하고, 모델의 내부 요소들 사이의 상호 일관성을 유지하면서 요구사항을 정제해야 한다[4]. 특히 분석 모델에서 클래스도는 시스템의 정적 뷰를, 쓰임새도는 동적 뷰를 나타내므로, 이들 사이의 일관성을 확인하는 작업은 반드시 필요하다[1].

본 논문은 객체 지향 분석 모델의 정확성과 상호 일관성을 검증하는 방법을 제공하여, 좋은 품질의 분석 산출물을 개발하는 지침으로 사용하고자 한다. 2장 관련 연구에서는 본 논문에서 다루는 UML의 쓰임새 모델과 분석 모델을 설명하고, 오류 검증의 자동화에 필요한 정형 명세 기법들을 조사하며, 3장은 분석 모델을 정형 명세하기 위한 New FST 정의와 정확성과 일관성 검증 규칙을 제안한다. 4장은 사례연구를 통하여 제안한 방법들을 시험해 보고, 기존의 연구들과 비교한다. 5장에서는 결론과 향후 연구방향을 제시한다.

2. 관련 연구

2.1 UML의 분석 모델

사용자의 요구사항을 기술한 쓰임새 모델을 입력으로 하는 UML 분석 모델은 클래스도와 교류도로 사용하여 요구사항을 구조화시킨다. 쓰임새 모델은 쓰임새도와 교류도, 사건 흐름 기술서 등을 사용하여 사용자의 요구사항을 파악하고, 기술한다. 분석 모델은 교류도와 클래스도 등을 사용하여 앞서 파악한 요구사항을 구조화시켜, 기능의 재사용성과 사용자 요구사항 변경 등의 유지보수를 용이하게 한다. 클래스도는 분석 클래스와 그들 사이의 관계로 구성된다. 분석 클래스는 경계 클래스와 엔티티 클래스, 제어 클래스 등의 3가지 스테레오 타입이 있고, 속성과 책임으로 구성된다. 책임은 설계 모델의 메소드이다. 클래스 사이의 관계는 연관 관계와 일반화 관계 등으로 표현된다. 교류도는 시간적 순서를 강조한 순차도와 구조를 강조한 협력도로 구성된다. 두 다이어그램 모두 객체와 메시지를 사용하여 객체들 사이에 협력을 기술하고 있다.

2.2 정형 명세 기법

정형 명세 기법(Formal Specification Technique)은 정확한 의미론을 정의하여 객체 지향 방법론을 구성하는 있는 표기와 다이어그램들의 의미를 깊게 고찰하기 위해 사용되며, 크게 메타 모델 또는 알고리즘 형식으로 명세된다. 개발자가 정형 명세를 이해하는 것은 어렵지만, 도구 기반으로 산출물들을 검증하고 검토하는데 용이하므로, 분석 모델의 정확성과 일관성을 검토하기 위해 반드시 필요한 작업이라 하겠다. 현재 Object-Z와 OOLOTOS, VDM++ 등이 있으며, 대부분 OMT 방법론을 기반으로 하고 있다. 반면 비정형 명세 기법(Informal Specification Technique)은 이해하기는 쉽지만, 정확한 의미론이 부족하여 개발자가 표기와 용법 등을 해석하는데 혼돈을 가져 올 수 있다. 그러나 대부분의 객체 지향 방법론이 비정형 명세 기법을 사용하고 있다.

UML은 메타 모델과 자연어를 함께 사용하고 신뢰성을 유지하기 위해 정형 명세 기법의 일부를 도입하여 완전한 의미론을 갖추고 있다[5]. 그러나 정형 명세 기법의 메타 모델은 풍부한 표기에 초점을 두어 비교적 이해하기 쉽고, 모델링 도구로의 구현은 용이하지만, 구성 요소들 사이의 정확한 의미론의 해석은 부족하다[7].

따라서 본 논문은 UML에서 메타 모델의 사용하여 일부 정형 명세 기법을 도입했다는 사실을 고려하여, 알고리즘 형태의 정형 명세 기법을 사용함으로서 메타 모델을 보충하고, UML의 분석 모델을 정의하여 정확한 의미론의 해석을 시도한다.

3. New FST 와 검증 규칙

먼저 분석 모델을 정형명세하기 위한 New FST를 정의하고, 정확성과 일관성 검증 규칙을 제안한다.

3.1 New FST

본 논문에서 제안하는 New FST는 분석 모델의 정확성과 일관성을 검증하기 위한 필요한 요소들을 사용하여 기술한 Definition 이다. 쓰임새도와 클래스도, 교류도 등에 관한 New FST가 Definition 3.1과 Definition 3.2, Definition 3.3 에 있다.

Definition 3.1 An use-case diagram is a three-tuple (U, A, R), where the following conditions are satisfied :

1. U is a finite set of the names of use-cases in an use-case diagram.
2. A is a finite set of the names of actors in an use-case diagram.
3. R is a finite set of the names of relations in an use-case diagram.
4. For each $r_i \in R$, r_i is consisted of a sender, a a receiver and type

Definition 3.2 A class diagram is a two-tuple (C, R), where the following conditions are satisfied :

1. C is a finite set of the names of classes in an class diagram.
2. R is a finite set of the names of relations in an class diagram.
3. For each class $c_i \in C$, c_i is consisted of a attributes stereotype, and responsibilities. The stereotype is boundary, control or entity.
4. For each $r_i \in R$, there are a sender, receiver, and type.

Definition 3.3 An Interaction diagram is a two-tuple (O, M), where the following conditions are satisfied :

1. O is a finite set of the names of the objects in an interaction diagram.
2. M is a finite set of the names of message in an interaction diagram.
3. For each object $o_i \in O$, o_i is consisted of object, type and class. The object may be anonymous.
4. For each message $m_i \in M$, m_i is consisted of an sender, a receiver and a sequence number.

3.2 분석 모델의 정확성 검증 규칙

쓰임새 모델을 포함한 분석 모델의 각 다이어그램의 정확성을 보장하기 위한 규칙들이 <표1.>과 <표2.>, <표3.> 등에 있다

<표1.> 쓰임새도의 정확성 규칙

구성요소		규칙
액터		ruleU1. 액터의 이름은 유일하다.
쓰임새		ruleU2. 쓰임새의 이름은 유일하다.
관계	연관	ruleU3. 액터와 쓰임새 사이에서 존재한다
	일반화	ruleU4. 액터와 액터 사이에서 존재한다.

<표2.> 클래스도의 정확성 규칙

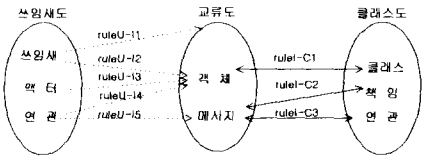
구성요소		규칙
클래스		ruleC1. 클래스의 이름은 유일하다.
관계	연관	ruleC2. 다른 클래스와 연관되지 않은 클래스는 존재하지 않는다.
	일반화	ruleC3. 일반화 관계는 순환되지 않는다.

<표3.> 교류도의 정확성 규칙

구성요소	규칙
메세지	ruleI1. 다른 객체들과 메시지로 연관되지 않은 객체는 존재하지 않는다.

3.3 분석 모델의 일관성 검증 규칙

쓰임새 모델의 쓰임새도는 사용자의 요구사항을 기술한 것으로 분석 모델의 교류도를 통하여 구현된다. 또한 교류도의 객체는 그 구조가 클래스도에 정의되어 있어야 한다. 쓰임새도와 교류도, 교류도와 클래스도의 일관성 검증을 위한 사상이 <그림1.>과 <표4.>에 있다.



<그림1.> 쓰임새도와 교류도, 교류도와 클래스도 사이의 일관성 규칙

<표4.> 일관성 검증 규칙

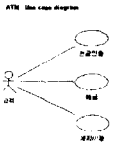
규칙	설명
ruleU-I1	쓰임새에 대응되는 교류도가 존재한다.
ruleU-I2	쓰임새는 교류도에서 제어 객체로 존재한다.
ruleU-I3	액터는 교류도에서 객체로 존재한다.
ruleU-I4	연관은 교류도에서 경계 클래스로 존재한다.
ruleU-I5	연관은 교류도에서 메시지로 존재한다.
ruleI-C1	객체는 클래스도에서 클래스로 존재한다.
ruleI-C2	메시지는 클래스도에서 메시지를 수신하는 클래스의 책임으로 존재한다.
ruleI-C3	메시지는 클래스도에서 연관 관계로 존재한다.

<표4.>의 ruleU-I1~ruleU-I5는 쓰임새도와 교류도 사이의 일관성 검증 규칙이고, ruleI-C1~ ruleI-C3 교류도와 클래스도 사이의 일관성 검증 규칙이다.

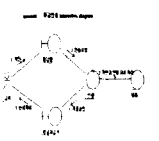
4. 사례 연구 및 비교 분석

4.1 사례 연구

ATM 시스템의 쓰임새 모델과 분석 모델이 <그림2.>과 <그림3.>, <그림4.> 등에 있다.



<그림2.> 쓰임도



<그림3.> 교류도

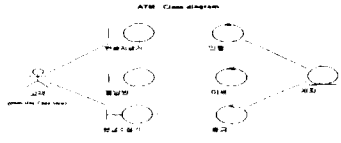


그림4. 클래스도

위 분석 모델을 New FST를 사용하여 정형명세하고, 위에서 도출한 규칙들을 적용하여 정확성과 일치성을 검증한다.

① 정형 명세

<그림2.>의 쓰임새도는 다음과 같이 정형 명세한다.

ATM-Usecase = (U, A, R)

U = {현금인출, 예금, 계좌이체 }

A = {고객}, ATM_R = { r_1, r_2, r_3}

관계	sender	receiver	type
r_1	고객	현금인출	연관
r_2	고객	예금	연관
r_3	고객	계좌이체	연관

<그림3.>의 교류도는 다음과 같이 정형 명세한다.

ATM-현금인출 = (O, M)

O = { :고객, :출납원, :인출, :계좌, :현금지급기}

M = {확인, 인출요청, 확인과 인출 요청 처리, 지급 승인, 현금지급}

객체 O

이름	타입	클래스
	액터	고객
	경계	출납원
	제어	인출
	엔티티	계좌
	경계	현금지급기

메시지 M

name	sequence no	sender	receiver
확인	1	:고객	:출납원
인출요청	2	:출납원	:계좌
확인과 인출 요청 처리	3	:계좌	:인출
지급 승인	4	:인출	:현금지급기
현금지급	5	:현금지급기	:고객

<그림4.>의 클래스도 또한 New FST를 사용하여 정형명세할 수 있다.

② 정확성 검증

New FST를 사용하여 쓰임새도 정형 명세에 <표1.>의 쓰임새도의 정확성 규칙을 적용하여 모델을 검토한다. ATM-Usecase = (U, A, R)에서 ruleU1과 ruleU2 규칙을 적용하여 A 튜플에서 액터의 이름이 유일한가, U 튜플에서 쓰임새의 이름이 유일한가를 검사한다. R 튜플에 ruleU3과 ruleU4를 적용하여, R 튜플의 요소의 타입이 연관일 경우에는 sender는 액터 이름, receriver는 쓰임새 이름인가를 확인한다. 일반화일 경우는 sender와 receiver 모두 액터인가를 확인한다. 이와 같이 <표2.>와 <표3.>에서 제안한 규칙을 사용하여 클래스도와 교류도의 정확성을 검사할 수 있다.

③ 일관성 검증

쓰임새도와 교류도의 정형 명세에 <표4.>의 일관성 검증 규칙, ruleU-I1~ ruleU-I5을 적용한다.

ruleU-I1를 적용하여 쓰임새의 집합인 U 튜플의 원소들이 교류도로 존재하는가 검사하고, U 튜플의 원소가 ruleU-I2는 이 교류도의 제어 객체로 존재하는가를 검사한다. ruleU-I3을 적용하여 액터들의 집합인 A 튜플의 원소들이 교류도의 객체로 존재하는가를 검사한다. ruleU-I4과 ruleU-I5를 사용하여 관계의 집합인 R 튜플에서 연관 관계가 협력도의 경계 객체로, 순차도의 메시지로 존재하는가를 검사한다.

이와 같이 교류도와 클래스도의 정형명세에 <표4.>의 일관성 검증 규칙, ruleI-C1~ ruleI-C3을 적용하여 두 다이어그램 사이의 일관성을 검사할 수 있다.

4.2 비교 분석

[2]는 UML을 기반으로 한 설계 모델을 메타 모델을 사용하여 정확성과 일관성을 검증하는 규칙들을 정의하고 있다. 그러나 메타 모델의 정형명세는 도구로의 구현이 쉽지가 않다.

[3]은 OMT의 객체 모형과 동적 모형 사이의 일관성을 진단하고, 동적 모형과 기능 모형 사이의 일관성을 진단하는 방법들을 제시하고 있지만, 도출한 규칙이 비교적 단순하고, 모델의 정확성을 검증하는 단계가 없다.

반면 본 논문에서는 UML을 기반으로 알고리즘 형태의 정형 명세 기법을 사용하여 UML을 보안하고, 도구 기반으로 구현이 용이하다는 점이 장점이라 하겠다.

5. 결론과 향후 연구 관계

분석 모델은 개발자의 언어로 작성되어 정확성과 일관성이 있는 개발 산출물로 관리되어야 한다. 이를 지원하기 위해 본 논문은 분석 모델을 일관성을 검증하기 위해, New FST와 정확성과 일관성 검증 규칙을 제안하고, 사례 연구를 통하여 적용하여 이 규칙들을 시험하였다. 본 논문은 UML 표기들 사이의 숨은 의미를 규칙으로서 분명히 드러내어, 좋은 품질의 분석 모델의 개발에 기여한다고 생각한다.

현재 이러한 검증 규칙을 자동화하기 위한 알고리즘에 관한 연구가 진행 중에 있다. 이를 바탕으로 검증 도구의 자동화 뿐만 아니라, 분석 모델을 자동으로 정형 명세하는 도구에 대한 연구도 진행되어야 한다.

6. 참고 문헌

[1] 박현철, 객체지향 분석 설계 Visual C++ 프로그래밍, 비앤씨, 1999.

[2] 정기원, 조용선, 권성구, "객체지향 설계 방법에서 오류 검출과 일관성 점검 기법 연구", 정보처리 논문지, 제 6권, 제 8호, 1999

[3] 천왕성 "객체지향 분석 모형간의 일관성 진단", 한국과학기술원 전산학과 석사학위 논문, 1997

[4] Ivar Jacabson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addision Wesley, 1998

[5] OMG Unified Modeling Language Specification, Ver. 1.3, June 1999.

[6] Jean-Michel Bruel, Robert B.France, Transforming UML Models to Formal Specifications, UML'98 - beyond the notation, LNCS. Spring, 1998

[7] Andy Evans, Robert France, Kevin Lano, Bernhard Rumpe, The UML as a formal modeling notation, UML'98 - beyond thenotation, LNCS. Spring, 1998