

객체지향 분석 단계에서 동적 행위 검증을 위한 정보 트리 확장에 관한 연구

*고상복, 김기한, 이경환

중앙대학교 컴퓨터공학과 소프트웨어공학 연구실

A Study on the Expansion of Information Tree for the Dynamic Behavior Verification in O-O Analysis Process

Sang-pok Ko, Gi-han Kim, Kyung-whan Lee

SE Laboratory, Dept. of Computer Science & Engineering, Chung-Ang University

요약

객체지향 방법의 초기 단계인 객체지향 분석은, 문제 기술서에서 사용자 요구 사항을 실 세계의 개념과 도메인 지식을 통하여 객체 관점에서 모델링하고, 객체 지향 요구 명세를 생성하는 절차이다. 기존에 제안된 객체 모델링 방법론에서 분석 단계의 정적 모델링은 시멘틱 모델 등의 풍부한 시멘틱을 제공하여 모델과 모델링의 많은 부분들을 정형화할 수 있었다. 그러나 대부분의 방법론들의 분석 과정은 동적 모델의 정형화가 미흡하다. 본 논문에서는 시스템의 정적인 구조를 검증할 수 있는 정보 트리 모델에 동적인 행위 정보를 포함하도록 확장하고, 검증 절차를 제시함으로써 시스템의 동적 행위를 검증할 수 있게 하였다. 또한, 객체 지향 요구 명세의 동적인 행위에 대한 검증은 확장 정보 트리를 사용하여 use case와 시나리오를 추적하면서 검증한다.

1. 서론

소프트웨어 개발에 있어서 주어진 문제를 확실하게 이해하고 사용자의 요구사항을 정확하게 추출하는 것은 분석 단계에서 이루어져야 하는 사항들이다[1]. 특히 객체 모델링에 대한 연구가 활발히 진행되면서 분석의 결과를 객체 단위로 표현함으로써 다음 단계로의 자연스러운 모델링이 수행된다는 점과, 변화의 가능성이 가장 적은 부분인 객체를 중심으로 분석함으로써 추후에 발생하는 변화로 인한 파급 효과를 최소화 할 수 있다는 장점이 있다[2]. 객체지향 분석은 구조적 분석에 비해, 좀더 실세계에 근접하는 장점이 있으나, 다음과 같은 몇 가지 이유에 의해, 객체지향 분석은 문제 기술서와 상이점이 존재할 수 있다 첫째, 대부분의 객체 지향 분석 방법은 객체, 동적, 기능 모델의 세 가지 다른 관점으로 시스템을 분석하기 때문에 모델들 사이에 불 일치성이 존재할 수 있다. 둘째로, 객체지향 요구명세는 객체 지향 분석 동안에 도메인 지식을 사용하여 문제 기술서로부터 불필요한 정보를 제거하고 필요한 부분을 추가하면서 작성되기 때문에 사용자 수준과 개발자 수준의 표현에 대한 차이가 생길 수 있다. 마지막으로 객체지향 방법은 객체지향 분석 동안에 엄격한 정의와 제약조건을 부여하지 않기 때문에 객체지향 요구 명세는 모호성피 불 일치성을 가질 수 있게 된다

본 논문의 목적은 객체지향 분석 과정에서 문제 기술서과,

요구 명세서의 일관성을 높이기 위해 동적 행위 검증을 수행하는 확장 정보 트리의 구축 절차와 검증 방법을 제안한다. 본 논문은 2장에서 관련 연구로서 동적 모델의 검증과 Yeom의 검증 방법을 살펴보고, 3장에서 객체지향 분석 과정에서의 정보 트리의 확장을 통한 동적 행위 검증 과정을 살펴보고, 4장에서 결론을 맺는 방식으로 구성되어 있다

2. 관련 연구

요구 명세 검증에 대한 연구는 요구 명세를 구성하는 분석 모델들 사이의 일관성과 완전성을 검증하는 방법과 요구 명세와 문제 기술서 사이의 일관성과 완전성을 검증하는 방법들로 구분된다[3]. 2.1절에서는 후자의 방법인 요구 명세와 문제 기술서 사이의 일관성과 완전성을 검증하는 Yeom의 검증 방법을 선택한다

2.1 Yeom의 검증 방법

이 방법에서는 요구 명세의 일관성과 완전성을 검증하기 위해서 정보 트리, 상속 그래프, 그리고 변이 추적 테이블이라는 그래픽 표기법을 사용하였다. 정보 트리는 시스템의 정적인 구조를, 상속 그래프는 시스템에 존재하는 클래스들 사이의 상속성을, 변이 추적 도는 하나의 시나리오에 대한 시스템의 시간의존적 동적 행위를 표현하였다.

그래픽 표기법들에 대해서 간략히 소개하면 다음과 같다. 정

보 트리는 상속 관계를 제외한 시스템의 정적인 구조를 표현하기 위해서 사용된다 이 트리의 구조는 루트에서 하부 노드까지 시스템, 클래스, 오퍼레이션, 속성, 그리고 제약 조건 노드들을 계층적 형태로 구성된다.

이 방법의 특징은 정보 트리를 구축하는 데 사용되는 정형 요구 명세 구조를 좀더 구체화하였고, 요구명세의 동적 행위를 검증하려는 시도를 하였다

상속 그래프는 시스템은 구성하는 클래스들 사이의 상속 관계, 즉, 하부 클래스는 상부 클래스의 특징을 상속받는 관계를 표현하기 위해서 사용된다 이 그래프의 루트는 시스템을 나타내며, 루트를 제외한 모든 클래스들은 상위 클래스들을 가지는 하위 클래스가 되며 자식 클래스들을 가질 때는 상위 클래스가 되기도 한다.

변이 추적 테이블은 시스템의 시간 의존적 동적 행위 표현하기 위해서 사용된다 이 테이블에서 수직선들은 각 객체를 나타내며 된 방향 화살표를 가진 수평선은 메시지를 나타낸다. 시간은 수직선을 기준으로 아래로 가면 증가한다.

여기서 제안된 그래픽 표기법들을 사용한 검증 방법에 대한 문제점은 다음과 같다

첫째, 시스템의 전체적인 동적 행위를 검증할 수 없다.

문제 기술서는 시간에 따라서 변하는 동적 행위를 나타내는 시나리오식으로 기술되어 있지 않다. 그러므로, 문제 기술서와 비교 검증하는 것은 부분적인 동적 행위의 검증은 가능하지만, 전체적인 동적 행위를 검증할 수 없다. 그리고, 제안된 변이 추적 테이블로서는 시스템의 전체적인 동적 행위 검증을 하지 못한다

둘째, 체계적인 정형 요구 명세 변환이 어렵다

요구 명세에서 형식 요구 명세로 변환하는 단계에서 대수 명세를 기본으로 한 형식 요구 명세 언어의 복잡성으로 인하여 체계적인 정형 요구 명세 변환이 어렵다

셋째, 시스템의 정적인 구조와 동적인 행위 구조의 통합이 이루어지지 않아서 분석 이후 단계들 고려한 시스템의 정적 및 동적 구조를 통합된 형태를 표현하지 못하고 있다.

3. 정보 트리 확장을 이용한 동적 행위 검증 방법

객체지향 소프트웨어의 동적 행위는 시스템에 존재하는 객체들 사이의 메시지 흐름으로 구성된다. 따라서, 요구 명세의 동적 행위 검증은 시스템에 존재하는 객체들 사이의 메시지 흐름을 추적하여 비교함으로써 검증할 수 있다[5]. 그러므로, 이 장에서는 이러한 메시지 흐름을 추적 비교할 수 있도록 요구 명세를 동적 행위 확장 정보 트리로 변환하는 체계적인 각각의 절차를 기술하고, 그 확장된 정보 트리를 사용하여 동적 행위를 검증하는 방법에 대해서 기술한다. 검증 방법의 전체적인 절차는 그림 1과 같다.

3.1 수정 요구 명세로의 변환

수정 객체 모델은 정형 요구 명세로 변환을 용이하게 하기

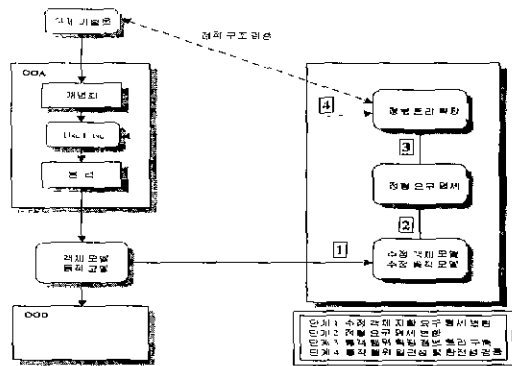


그림 1. 검증 방법의 전체 절차

위에서 객체 모델을 수정 객체 모델로 변환하는 것이고, 수정 동적 모델은 동적 모델은 시스템의 시간 의존적인 동적 행위를 기술한다 하지만, 그 모델은 객체 하나에 대한 상태, 상태변이, 그리고 메시지들을 나타내기 때문에 객체들 사이의 메시지 흐름을 나타내지 못한다 그래서, 이러한 메시지 흐름을 표기하기 위하여 동적 모델을 수정 동적 모델로 변환하여 사용한다.

3.2. 정형 요구 명세 변환

정형 요구 명세란 언어의 단어, 구조, 의미 등이 정형적으로 정의되고, 정형논리에서 일반적으로 사용되는 수학적 기초를 갖는 언어에 의해 표현되는 명세이다 수정 객체 지향 요구 명세를 정형 요구 명세로 변환하는 이유는 첫째, 수정 객체 모델과 수정 동적 모델을 통합해야 하기 때문이며, 둘째로 정형 방법을 사용하여 객체 지향 요구 명세를 분명하게 표현하기 위해서이다 본 논문에서 사용하는 정형 요구 명세 언어는 Yeom[4]에서 사용된 언어를 수정하였다.

3.2.1 정형 요구 명세 구조

객체 지향 소프트웨어 개발은 객체를 기반으로 하기 때문에 소프트웨어 시스템은 객체들의 집합으로 정의할 수 있다. 그래서, 정형 요구 명세는 클래스 단위 명세로 구성되며 그 구조는 다음과 같다.

```
class spec CLASS NAME is
begin
relations          R1,R2,R3, . . . ,Rm
attributes         A1,A2,A3 . . . ,An
operations         OP1,OP2,OP3, . . . ,OPp
operations sequence OS1, OS2, OS3 . . . , OSq
end
[ Ri,1 ≤ i ≤ m, Ai,1 ≤ i ≤ n, OPi,1 ≤ i ≤ p, OSi,1 ≤ i ≤ q ]
```

3.2.2 오퍼레이션 상호작용 명세

클래스 내의 오퍼레이션들은 가능한 순서들을 지정할 수 있다 즉, 시스템에 존재하는 모든 오퍼레이션들에 대해서 가능한 순서가 존재한다는 것을 의미한다 그러므로, 오퍼레이션 상

호작용 명세는 클래스 내부의 오퍼레이션의 가능한 순서를 참조하면서 오퍼레이션에 연관된 객체와 메시지를 추가하여 전체적인 시스템의 동적 행위를 표현할 수 있게 하였다. 오퍼레이션 상호작용 명세의 구조는 다음과 같다.

$\langle OP_1 \rangle \Rightarrow sm(to C_1) rm(from C_2) \langle OP_2 \rangle$
 $OP_1, OP_2 = operation, C_1, C_2 = class$
 $sm = send message, rm = receive message$

3.3. 정보 트리 확장

확장 정보 트리는 시스템의 정적인 구조와 시간 의존적인 동적 행위를 나타내는 그래프 표기이다. 이 트리에서 표현되어지는 동적 행위는 오퍼레이션들 사이의 메시지 흐름으로 표기되어진다. 확장된 정보 트리의 구조는 그림 2와 같으며, 굵은 실선 내의 부분에서 시간 의존적 동적 행위를 나타낸다. 동적 행위를 나타내는 부분은 클래스 내의 오퍼레이션들의 순서와 클래스들 사이의 오퍼레이션들의 순서로 나뉘어진다.

먼저, 같은 클래스 내에 존재하는 오퍼레이션들의 순서는 실선으로 표기한다. 실선 위에는 메시지를 표기하며 화살표를 받은 오퍼레이션을 수행하기 위해 다른 객체에서 보내온 메시지를 표시한다. 다음으로, 서로 다른 클래스에 존재하는 오퍼레이션들의 사이의 메시지 흐름은 점선으로 표기하고, 클래스의 오퍼레이션에서 다른 클래스로부터 메시지를 받거나, 다른 클래스로 메시지를 보낸다면, 어느 클래스로 메시지를 보냈고, 어느 클래스에서 메시지를 받았는지를 정의되어야 한다. 이 조건은 클래스들 사이의 오퍼레이션들의 결과에 의한 메시지들을 잃어버리는 것을 방지할 수 있다. 이러한 방법을 이용하여 메시지의 일관성을 검증할 수 있게된다.

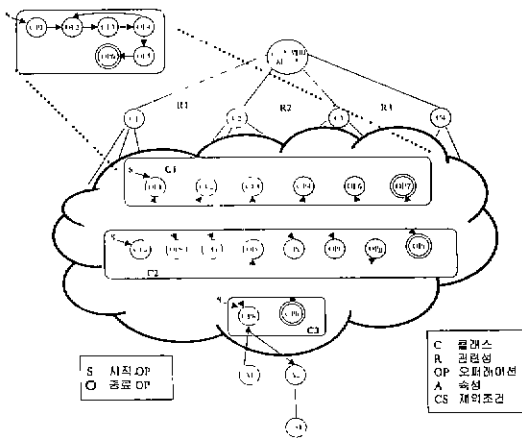


그림 2 확장된 정보 트리의 구조.

3.4 일관성과 완전성의 검증

객체 지향 요구 명세에 대한 일관성과 완전성 검증은 상향식과 하향식 방법으로 구분된다. 먼저 각각의 방법은 다음과 같은 단계를 갖는다

<하향식 방법>

- 1 Use case에서 시나리오를 생성
2. 확장 정보 트리의 메시지 흐름 추적
- 3 일관성 검증
- 4 위의 단계들을 반복하여 완전성 검증

<상향식 방법>

- 1 확장 정보 트리로부터 시나리오를 생성
2. Use case에 존재하는 시나리오와 비교
- 3 일관성 검증
- 4 위의 단계를 반복하여 완전성 검증

4. 결론 및 향후 과제

기존에 제안된 객체 모델링 방법론들은 구조적 방법에서 사용되던 ER 다이어그램, 시멘틱 모델 등의 풍부한 시멘틱을 그대로 사용할 수 있기 때문에 정적 측면의 모델링에서 많은 부분들을 정형화할 수 있다. 그러나 동적 측면의 모델링에서는 객체와 객체간의 상호 작용관계와 시스템 전체의 행위를 모델링하지 못하고, 객체간의 시간적 제약성을 표현하지 못하며 객체간의 병행성이나 동기화도 표현하기 힘들다는 문제점을 가지고 있었다. 또한 기존의 객체지향 방법론들이 모델링에 중점을 두고 제안되어 모델간의 비정형화라는 문제점도 가지고 있다.

본 논문에서는 시스템의 정적인 구조를 검증할 수 있는 기존의 정보 트리라는 모델에 동적인 행위 정보를 포함하도록 확장하고, 검증 절차를 제시함으로써 시스템의 동적 행위를 검증할 수 있게 하였다. 또한, 객체 지향 요구 명세의 동적인 행위에 대한 검증은 위에서 구축된 확장 정보 트리를 사용하여 use case와 시나리오를 추적하면서 검증한다.

향후 연구 방향으로는 동적 검증을 분석단계에서만 극한하지 않고 분석 이후의 단계들에 대해서도 검증할 수 있는 방법들이 연구되어야 할 것이다

5. 참고문헌

- [1] Robert H. Bourdeau and Betty H Cheng, "A Formal Semantic for Object Model Diagrams" IEE Trans. on SE, Vol 21, No. 10, pp. 799-821, 1995
- [2] 이경환, 객체 모델링 방법론, 중앙대학교 출판국, 1996
- [3] Stephen S Yau, D H. Bae and K. Yeom, "An Approach to Object-Oriented Requirements Verification in Software Development for Distributed Computing Systems", Proc. 18th International Computer Software & Applications Conference, November, pp. 96-102.
- [4] 임근혁, "분산 시스템을 위한 객체 지향 소프트웨어 개발에서 요구 명세에 대한 검증 방법", 정보 과학회지, 23(1), 1996, pp 643-646
- [5] 백진욱, "객체지향 요구 명세의 동적 행위 검증에 관한 방법", 한국과학 기술원 석사학위 논문, 1997