

# 역할 모델을 이용한 컴포넌트 명세 방법

김정중 송호영<sup>0</sup> 박운재 송의철

경남대학교 컴퓨터공학과

{jjkim, humanism<sup>0</sup>}@zeus.kyungnam.ac.kr

## How to Specify Components Using Role Models

Jungjong Kim Hoyoung Song<sup>0</sup> Woonjai Park Euicheol Song  
Dept. of Computer Engineering, Kyungnam University

### 요약

양질의 컴포넌트 소프트웨어를 개발하기 위해서는 사용자의 요구사항을 정확하게 파악하고 체계적인 설계 방법이 필요하다. 그러나 컴포넌트 소프트웨어는 일반적으로 객체 모델링을 기반으로 설계되고 개발되기 때문에 복잡한 문제를 가진 대규모의 시스템에는 적합하지 않으며 상호작용과 협력의 표현, 상속에 의한 재사용 문제 등을 해결하는 데는 한계가 있다.

따라서 본 논문은 객체 모델링을 보완하기 위하여 객체 중심이 아닌 역할 중심으로 하는 새로운 추상화 기법인 역할 모델을 이용한 컴포넌트 명세 방법을 연구하였다.

### 1. 서론

컴포넌트 기반 소프트웨어 개발에서 양질의 컴포넌트 소프트웨어를 개발하기 위해서는 사용자의 요구사항을 정확하게 파악하고 체계적인 설계 방법이 필요하다. 그러나 컴포넌트 소프트웨어는 일반적으로 객체 모델링을 기반으로 설계되고 개발되기 때문에 복잡한 문제를 가진 대규모의 시스템에는 적합하지 않으며 상호작용과 협력의 표현, 상속에 의한 재사용 문제 등을 해결하는 데는 한계가 있다[1][2].

따라서, 이런 복잡한 대규모 시스템을 객체의 패턴에 따라 간단한 모델로 분할할 필요성이 있으며 이를 위하여 역할 모델링 방법이 제안되었다.

본 논문은 역할 모델링을 통하여 생성된 역할 모델을 이용하여 컴포넌트를 명세하는 방법을 연구한다. 이는 역할 모델링의 장점을 획득할 수 있을 뿐만 아니라 컴포넌트가 가지는 이점도 얻을 수 있다.

### 2. 관련연구

#### 2.1. 컴포넌트 기반 소프트웨어 개발

현재 새롭게 대두되고 있는 컴포넌트 기반의 소프트웨어 개발은 기존의 컴포넌트를 조합함으로써 시스템을 완성하는 것을 의미한다[3][4]. 컴포넌트를

사용한 소프트웨어 시스템 개발 기술은 과거 구조적인 방법이나 객체지향 방법이 해결하지 못하였던 소프트웨어 재사용과 생산성, 시스템 품질 관리 등에 대한 새로운 해결책으로 제시되고 있다.

컴포넌트는 인터페이스라 불리는 명세상의 의존관계를 통해 객체 명세를 정확히 기술함으로써 객체 기술 원칙을 확장하고 있다. 인터페이스는 객체를 호출하는 사용자와 객체의 기능을 분리할 수 있는 중요한 간접 계층이다. 또한 객체의 전체 기능을 명세화 하는 것은 여러 개의 인터페이스에 걸쳐 나누어 기술될 수도 있다.

컴포넌트 기술은 객체 기술을 기반으로 하고 있기 때문에 객체 모델링으로 컴포넌트를 개발하는 데는 한계가 있다.

#### 2.2. 객체 모델링의 문제점

객체나 클래스 개념으로 한 객체 모델링 방법으로는 재사용성과 생산성을 극대화하기에는 부족한 면이 있으며, 몇 가지 문제점을 나타내고 있다[1][2].

##### (1) 객체

객체의 상태 영역과 행위는 객체가 생성될 때 결정된다. 즉, 객체가 그것의 타입을 동적으로 변경하는 것이 불가능하다.

##### (2) 상호작용과 협력의 표현

객체들은 작업을 수행하거나 목표를 달성하기 위하여 서로 협력할 필요가 있다. 그러나, 객체 모델링 방법은 객체의 전체 집합 보다는 각 객체에 너무 초점을 둔다. 각 객체에 초점을 맞추는 것은 혼란을 가져올 수 있고 양질의 소프트웨어를 양성할 수 없다. 객체는 하나의 추상화 단위인 어플리케이션에서, 적절한 하위 집합 혹은 하위시스템을 추출하고 패키징하기가 매우 어렵다.

### (3) 상속에 의한 재사용

상속에 의한 재사용은 개발자가 상속하고 오버라이드 할 대상을 결정하는 것을 어렵도록 한다. 또한 상속은 하위 클래스가 모든 상위 클래스의 특성과 행위를 얻는다는 것을 의미하기 때문에 시스템의 분해를 어렵게 한다.

### (4) 복잡한 관계 구조

객체 모델링은 시스템 전체가 객체를 중심으로 결합되어 있으므로 복잡한 문제를 가진 대규모 시스템을 모델링하는 것은 쉽지가 않을 뿐만 아니라 이해하기도 어렵다. 또한, 복잡한 관계 구조로 인하여 추가적인 요구 사항에 대처가 쉽지 않다.

이런 여러 문제점들로 인해 새로운 추상화 방법으로서 객체 중심이 아닌 역할을 중심으로 한 모델링 방법을 제안한다.

## 2.3. 역할

클래스의 개념처럼 역할은 객체 집합의 기술이다. 그러나, 클래스와 다른 점은 클래스는 일반 특성들을 나타내는 객체 집합을 기술하지만 역할은 객체 패턴에서 같은 위치에 있는 객체 집합을 기술한다. 역할은 특정 문맥에 참여하는 개체의 행동이다. 즉, 표현된 하나의 추상개념이다. 역할은 하나의 개체에 대하여 관점에 따라 서로 다른 역할을 가질 수 있다[5][6].

또한, 역할은 객체의 속성과 오퍼레이션에 의해 행위를 기술한다. 역할에 의하여 정의된 행위는 객체가 어떻게 행동할지를 명확하게 하기 위하여 이용된다. 역할의 사용은 객체들이 변화와 확장에 잘 적응할 수 있도록 한다.

## 2.4. 역할 모델

역할 모델은 객체 모델에서 객체의 패턴을 추상화한 모델이며 객체 구조가 적절한 역할들을 수행함으로써 주어진 관계 영역을 완성하는 방법을 기술한다 [5]. 역할 모델은 관계의 분리를 지원하고 하나의 결합 모델에서 실제로 현상의 정적·동적 속성들을 기술한다. 이 관계의 분리는 상호작용하는 객체에 대해 대규모의 복잡한 구조를 가진 시스템에서 대규모의 복잡한 현상을 많은 하위 현상으로 분할되는 것

을 말하며, 각 하위 현상은 자신의 역할 모델에 의해 기술된다.

현상은 많은 협력하는 객체들에 의해 기술된다. 하위 현상은 그들의 관계 영역에 의해 명시되고, 하위 현상을 기술하는 객체들은 객체의 패턴에서 구성되며, 패턴에서 같은 위치를 가지는 모든 객체들은 역할에 의해 표현된다.

## 3. 역할 모델링

역할 모델링은 역할을 중심으로 객체들의 관계에 따른 새로운 추상화 방법으로서 객체가 수행하는 역할을 기반으로 모델링하는 방법이다.

### 3.1 역할 모델링 단계

역할 모델링은 8단계로 구성된다. 각 단계는 실행 가능한 일로 끝나며 역할 모델 상에 관점이다. 이들 관점에 대한 상대적 중요성은 모델의 목적에 달려있다. 단계들은 역할 모델이 충분히 정의될 때까지 반복하여 수행된다.

다음은 역할 모델링의 단계이다. 단계 ④와 ⑤는 병행하여 수행된다.

- ① 관계 영역을 결정한다
- ② 문제를 이해하고 객체의 성질을 식별한다
- ③ 환경 역할들과 자극(stimulus)/반응을 결정한다
- ④ 역할들을 식별하고 이해한다
- ⑤ 메시지 순서를 결정한다
- ⑥ 협력 구조를 결정한다
- ⑦ 인터페이스를 결정한다
- ⑧ 역할 행위를 결정한다

### 3.2 역할 모델링

역할 모델링을 위하여 회사 사람이 회사 경비로 여행을 가기 원한다고 할 때 역할 모델을 생성하는 과정을 예로 보여준다. 회사 조직에 대한 객체 모델을 생성하고, 여행 경비 예에서 상호작용하는 객체들에 대하여 패턴을 정의하고 추상화하여 역할 모델을 생성한다.

#### (1) 객체 모델

다음 그림 1은 회사 조직을 나타내는 객체 모델이다. 사각형은 객체이고 선은 객체들 사이 관계를 나타낸다. 화살표는 아래 시나리오를 바탕으로 한 흐름이다.

#### (2) 시나리오

Peter라는 사람이 회사 경비로 여행을 하기 원한다고 할 때 다음과 같은 처리가 이루어 질 수 있다.

- ① Peter는 그의 매니저 Eve에게 여행허가를 요청한다.
- ② Eve는 예산과 계획을 검사하고 Peter에게 여행을 허

가한다.

- ③ Peter는 티켓을 구매하고 여행을 한다. 그리고 돌아와서 경비 보고서를 작성하고 Eve에게 보고서를 전달한다.
- ④ Eve는 경비 보고서를 검사하고 허가한 다음 bookkeeper인 Bill에게 허가된 경비 보고서를 전달한다.
- ⑤ Bill은 금전 장부를 정리하고 Paymaster인 Joe에게 지불해줄 것을 요구한다.
- ⑥ Joe는 Peter에게 돈을 지불한다.

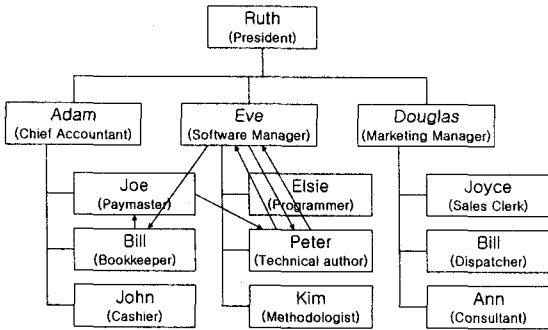


그림 1. 회사 조직의 객체 모델

(3) 역할 모델

위의 시나리오를 바탕으로 그림 1의 객체 모델에서 일정한 패턴을 정의하여 역할 모델을 생성할 수 있다. 상호작용하는 객체들은 Peter, Eve, Bill, Joe이다. 이들 객체들을 살펴보면 일정한 패턴을 유지한다고 볼 수 있다. 따라서, 이 객체 패턴을 추상화하여 Traveler, Authorizer, Bookkeeper, Payment 역할을 얻을 수 있다. 다음 그림 2는 여행 경비에 대한 역할 모델을 보여준다.

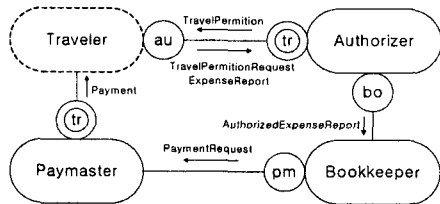


그림 2 여행 경비에 대한 역할 모델

4. 컴포넌트 명세

4.1. 컴포넌트 명세 단계

다음 그림 3은 컴포넌트 명세 단계를 나타낸다. 컴포넌트 명세 단계는 역할 모델링에서 생성된 역할 모델을 기반으로 하여 작업을 진행한다. 이들 명세된 컴포넌트를 조립하면 완성된 컴포넌트들을 생산

할 수 있다.

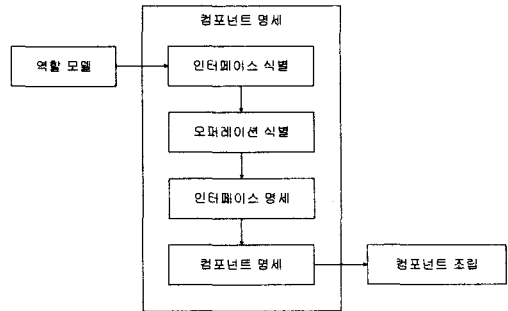


그림 3. 컴포넌트 명세 단계

4.2. 인터페이스와 오퍼레이션 식별

인터페이스는 역할들의 상호작용에 초점이 맞춰져 있으며, 그 상호작용으로부터 도출되어진다. 각 역할마다 1개의 인터페이스를 정의한다. 그 다음, 각 역할들의 행동을 하나씩 살펴보고 시스템이 이 행동들을 만족시키기 위해 어떤 기능을 제공해야 할 책임이 있는가를 판단한다. 만약 있다면 적당한 인터페이스가 제공하는 한 개 이상의 오퍼레이션으로서 표현한다. 다음 그림 4는 그림 2의 역할 모델을 기반으로 하여 식별한 인터페이스와 오퍼레이션들을 나타낸다.

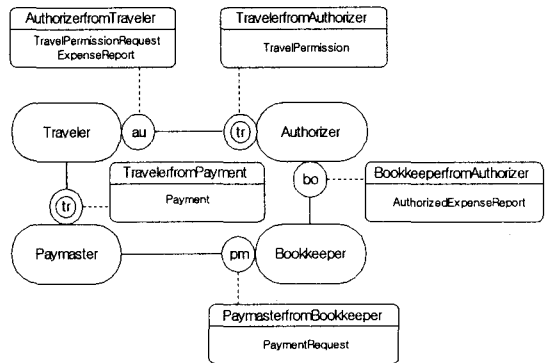


그림 4. 인터페이스와 오퍼레이션 식별

4.3. 인터페이스 명세

인터페이스 명세는 사용 계약을 정의한다. 인터페이스는 오퍼레이션들의 집합이며 각 오퍼레이션에 대한 기술을 정의하고 파라미터 타입을 포함한다. 다음 그림 5는 인터페이스 명세에 포함되어야 하는 항목이다. 그림 6은 그림 4에서 식별한 인터페이스들 중 BookkeeperfromAuthrizer에 대하여 명세를 작성한 것이다.

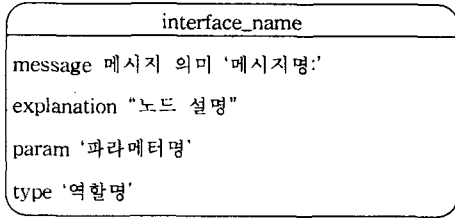


그림 5. 인터페이스 명세 항목

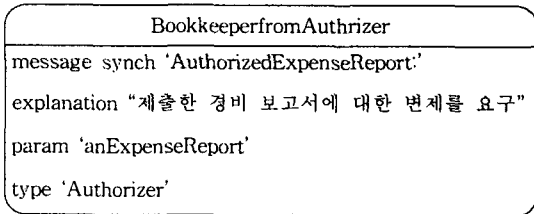


그림 6. 인터페이스 BookkeeperfromAuthrizer 명세

4.4. 컴포넌트 명세

컴포넌트는 단독으로 수행되는 작은 응용이라기 보다는 조립되고 커스터마이져 되어짐으로서 비즈니스 로직을 수행하는 부품이다. 따라서 오직 정규 포맷을 따르는 인터페이스로 명세화된 컴포넌트만이 활용 가능하다.

그림 7은 본 논문에서 활용한 여행 경비 예에 대한 컴포넌트 명세 다이어그램을 보여준다. 여행 경비(TravelerExpense) 컴포넌트는 지원되거나 사용되어야 하는 인터페이스들(AuthorizerfromTraveler, TravelerfromAuthorizer, BookkeeperfromAuthrizer, PaymasterfromBookkeeper, TravelrfromPayment)을 정의하고 있다.

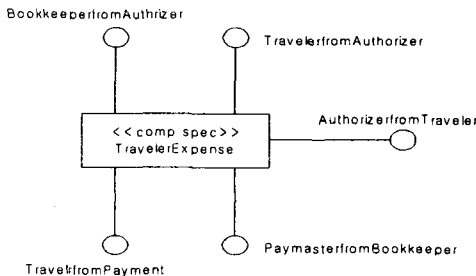


그림 7. TravlerExpense 컴포넌트 명세 다이어그램

5. 결론

본 연구에서는 객체 모델링의 문제점을 해결하기 위하여 새로운 추상화 기법인 객체의 역할을 중심으

로 한 역할 모델링을 기반으로 컴포넌트를 명세하였다. 이는 역할 모델링의 장점을 획득할 수 있을 뿐만 아니라 컴포넌트가 가지는 이점도 얻을 수 있다. 즉, 객체들의 패턴을 추상화하고 복잡한 대규모 시스템 모델을 관계의 분리를 통하여 작은 모델들로 분할할 수 있기 때문에 시스템의 이해력을 높일 수 있으며, 객체의 상호작용과 협력을 보다 잘 표현할 수 있고 재사용성을 향상시킬 수 있다.

향후 연구 과제로는 컴포넌트에 대한 보다 상세한 명세가 필요하며, 완성된 컴포넌트를 생산하는 것이다.

참고 문헌

- [1] E. P. Andersen, "Conceptual modeling of Object: A Role Modeling Approach," PH.D Thesis, University of Oslo, 1997.
- [2] Bent B. Kristensen, "Object Oriented Modeling with Role," In Proceedings of the 2nd International Conference on Object-Oriented Information Systems(OOIS'95), Dublin, Ireland, London, pp. 57-71, 1995.
- [3] Alan W. Brown, Kurt C.Wallnau, "The Current State of CBSE," IEEE Software, Vol. 15, No. 5, 1998.
- [4] Jun-Jang Jeng and Betty H.C. Cheng, "A Formal Approach to Reuse More General Components," IEEE Proceedings of 9th Knowledge-Based Software Engineering Conference, Monterey, California, September, 1994.
- [5] D. Bäumer, D. Riehle, W. Siberski, M. Wulf, "The Role Object Pattern," In Proceedings of 4th Conference on Pattern Languages of Programs, 1997.
- [6] Ralph Depke, Gregor Engels, Jochen Malte Küster, "On the Integration of Roles in the UML," Technical Report No. 214, University of Paderborn, August, 2000.
- [7] Liping Zhao, Elizabeth A. Kendall, "Role Modeling for Component Design," Proceedings of 33rd International Conference on Technology of Object-Oriented Languages, pp. 312-323, 2000.