컴포넌트 추출을 위한 UML 기반의 역할 모델 표현에 관한 연구

김정종 송호영O 박운재 송의철
경남대학교 컴퓨터공학과
{jjkim, humanismO}@zeus.kyungnam.ac.kr

A Study on UML-Based Role Model Representation for Extracting Components

Jungjong Kim Hoyoung SongO Woonjai Park Euicheol Song
Dept. of Computer Engineering, Kyungnam University

요약

컴포넌트는 일반적으로 객체 모델링을 기반으로 설계되고 개발되기 때문에 상호작용과 협력의 표현, 상속으로 인한 재사용 문제 등을 해결하는 데는 한계가 있으며 복잡한 구조를 가진 대규모 시스템에서 컴포넌트를 추출하기에는 용이하지 않다. 따라서 객체 모델링을 보완하기 위하여 객체 중심이 아닌 객체의 역할을 중심으로 하는 역할 모델링 기법이 제안되었다.

본 연구는 UML 기반에서 다양한 관점의 역할 모델을 표현하여 컴포넌트를 추출하는데 활용할 수 있도록 한다.

1. 서론

객체지향 소프트웨어 개발 방법은 객체를 중심으로 한 객체 모델링을 기반으로 시스템을 개발한다. 그러나 객체나 클래스를 개념으로 한 객체 모델링의 방법으로는 재사용과 생산성을 극대화하기에는 부족한 면이 있다. 또한 복잡한 문제를 가진 대규모의 시스템에는 적합하지 않으며 상호작용과 협력의 표현, 상속으로 인한 재사용 문제 등을 해결하는 데는 한계가 있다[4, 5].

따라서 객체 모델링 방법을 보완하고 개선할 수 있는 방법으로 객체가 아닌 객체의 역할을 중심으로 하는 새로운 추상화 기법인 역할 모델링 방법이 제안되었다. 역할 모델링은 객체들의 패턴을 추상화하고 복잡한 대규모 시스템을 관계의 분리를 통하여 간단한 모델로 생성할 수 있도록 한다[1, 7].

본 논문은 역할 모델링을 통하여 생성된 역할 모델을 UML에서 표현하는 방법을 연구한다. 물론 UML이 객체 모델링을 위한 표준으로서 다양한 관점으로 시스템을 모델링하며 역할 개념도 반영하고 있지만 이들 역할 개념들이 요구 사항을 충족시키지 못하고 있다. 따라서 본 연구는 다양한 관점으로 역할 모델을 표현할 수 있도록 UML을 기반으로 역할 모델을 표현하며 이를 바탕으로 컴포넌트를 추출한다.

2. 관련연구

2.1 객체 모델링

객체 모델링 방법이 많은 장점을 가지고 소프트웨어를 개발하는데 있어 각광을 받아오고 있지만, 객체나 클래스 개념으로는 재사용과 생산성을 극대화하기에는 부족한 면이 있으며, 몇 가지 문제점을 나타내고 있다[3, 4, 5].

(1) 객체

객체의 상태 영역과 행위는 객체가 생성될 때 결정된다. 애트리뷰트와 메소드들은 그것의 생명주기 동안 객체에 추가되거나 혹은 객체로부터 제거되도록 허용되지 않는다. 즉, 객체가 그것의 타입을 동적으로 변경하는 것이 불가능하다.

(2) 상호작용과 협력의 표현

컴포넌트는 작업을 수행하거나 목표를 달성하기 위하여 서로 협력할 필요가 있다. 그러나, 객체 모델링 방법은 객체의 전체 집합 보다는 각 객체에 너무 초점을 맞춘다. 각 객체에 초점을 맞추는 것은 혼란을 가져올 수 있고 양질의 컴포넌트를 생성할 수 없다.

(3) 상속

상속은 개발자가 상속하고 오버라이드할 대상을 결정하는 것을 어렵도록 한다. 또한 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 구조를 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 방법이 필요하다.

2.2 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[2, 7].

역할의 사용은 객체들이 변화와 확장에 잘 적응할 수 있도록 하고, 관련 속성들의 집합으로 그룹화하여 더 조직적으로 만들기 때문에 재사용성을 향상시킨다.

2.3 역할 모델

역할 모델은 객체 모델에서 객체의 패턴을 추상화한 모델이며 객체 구조가 적절한 역할들을 수행함으로서 주어진 관계 영역을 완성하는 방법을 기술한다[2, 7]. 역할 모델은 관계의 분리를 지원하고 하나의 결합(coherent) 모델에서 실세계 현상의 정적·동적 속성들을 기술한다.

역할 모델은 다른 종류의 현상을 식별하고, 객체의 이점을 간직한 채 그대로 분리하는 것을 허용하기 때문에 객체지향 패러다임의 강력한 확장이다.

2.4 UML

UML(Unified Modeling Language)은 Rumbaugh의 OMT방법론, Booch의 Booch방법론, Jacobson의 OOSE 방법론과 기타 다른 전문가들의 주장을 통합하여 만든 모델링 개념의 공통집합으로 객체지향 분석 및 설계 방법론의 표준 지정을 목표로 제안되었으며 OMG에 의해 객체지향 모델링 언어의 산업 표준으로 승인되어 현재 널리 사용되고 있다[6].

3. 역할 모델링

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

3.1 역할 모델링 단계

UML에서 역할 모델링을 통하여 다양한 관점의 역할 모델을 생성하고 컴포넌트를 추출하기 위하여 9단계를 제안한다. 이들 관점에 대한 상대적 중요성은 모델의 목적에 달려있다. 단계들은 역할 모델이 충분히 정의될 때까지 반복하여 수행된다.

다음은 역할 모델링의 단계이다.

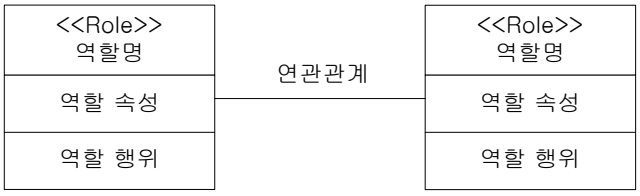
① 관계 영역을 결정
② 문제를 이해하고 객체를 식별
③ 객체 패턴을 식별
④ 역할들을 식별
⑤ 역할 행위를 결정
⑥ 메시지 순서를 결정
⑦ 협력 구조를 결정
⑧ 인터페이스 명세
⑨ 컴포넌트 명세

3.2 역할 모델의 표현

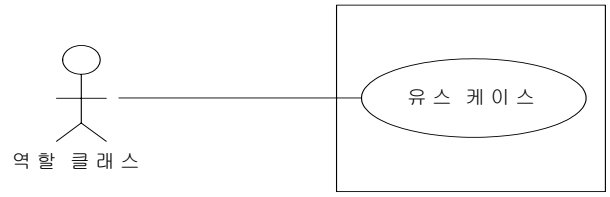
모델링을 통하여 생성된 모델은 UML 상에서 다양한 관점의 다이어그램들로 나타낼 수 있다.

(1) 역할 클래스 다이어그램

그림 1은 역할 클래스 다이어그램을 보여준다. 역할 클래스라는 것을 명시하기 위하여 스테레오 타입으로서 <<Role>>으로 표기하며 역할명을 기입한다. 각 역할 클래스는 역할이 가지는 속성과 행위들을 가진다. 또한, 각 역할 클래스들 사이 역할 관계는 연관관계로서 표현한다.



(그림 1) 역할 다이어그램



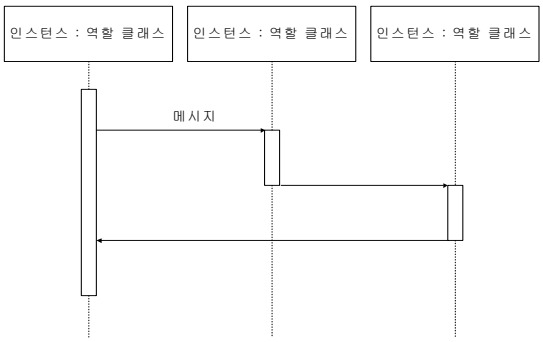
(그림 2) 역할 유스 케이스 다이어그램

(2) 역할 유스 케이스 다이어그램

그림 2의 역할 유스 케이스 다이어그램은 액터인 역할 클래스와 유스 케이스 사이 상호작용을 보여준다.

(3) 역할 순차 다이어그램

그림 3의 역할 순차 다이어그램은 객체들 사이 상호작용의 시간 순서에 대한 기술이다. 상호작용은 송・수신자 객체에 메시지를 전송하는 이벤트를 나타낸다. 송・수신하는 객체들은 역할 클래스의 인스턴스들로 표현된다.



(그림 3) 역할 순차 다이어그램

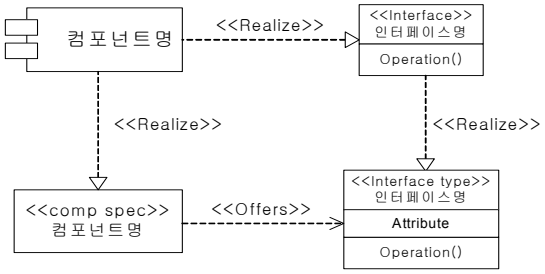
(4) 역할 협력 다이어그램

그림 4의 역할 협력 다이어그램은 역할 인스턴스들과 이들 사이 메시지 경로를 보여준다. 즉, 역할 인스턴스들 사이 상호작용하는 관계를 나타낸다. 역할 협력 다이어그램은 협력하는 역할 인스턴스들의 포트를 서로 연결하여 메시지가 어떤 경로를 통하는지에 초점을 둔다. 각 역할 클래스의 인스턴스들 사이 상호작용은 연관관계에 메시지를 첨부하여 나타내며 메시지 흐름은 화살표로서 표현한다.

(그림 4) 역할 협력 다이어그램

(5) 역할 컴포넌트 다이어그램

역할 컴포넌트 다이어그램은 컴포넌트, 인터페이스로 구성되어 있으며 각각의 관계를 설정한다. 그림 5는 역할 컴포넌트 다이어그램을 나타낸다.



(그림 5) 역할 컴포넌트 다이어그램

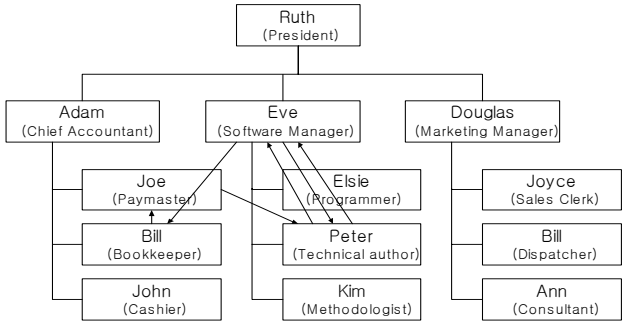
그림 5는 역할 컴포넌트 다이어그램을 나타낸다. 인터페이스는 컴포넌트에 작은 원을 연결하여 나타낼 수 있고 그림과 같이 실체화하여 오퍼레이션들의 집합으로 나타낼 수도 있다. 인터페이스와 컴포넌트에 각각의 명세(인터페이스 타입과 comp spec)를 실체화시킬 수 있다.

4. 사례연구

사례연구는 회사 사람이 회사 경비로 출장을 갈 때 모델링하는 과정을 보여준다. 먼저, 회사 조직에 대한 객체 모델을 생성하고, 상호작용하는 객체들에 대하여 패턴을 정의하고 추상화하여 UML 기반의 역할 모델에 대한 다양한 다이어그램들을 생성한다.

(1) 객체 모델

다음 그림 6은 회사조직을 나타내는 객체 모델이다. 사각형은 객체이고 선은 객체들 사이 관계를 나타낸다. 화살표는 아래 시나리오를 바탕으로 한 흐름으로서 객체의 패턴을 식별할 수 있다.



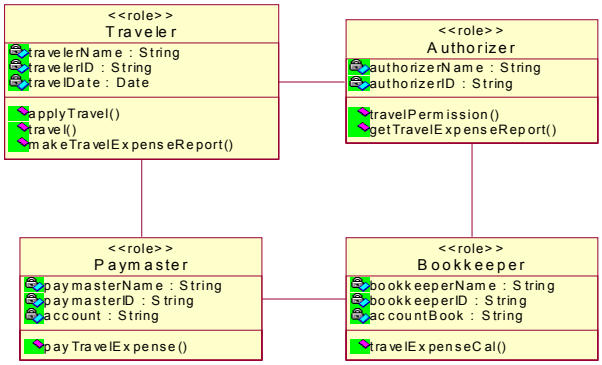
(그림 6) 회사 조직의 객체 모델

Peter라는 사람이 회사 경비로 출장을 갈 때 다음과 같은 시나리오로 처리가 이루어진다.

① Peter는 그의 매니저 Eve에게 출장허가를 요청한다.
② Eve는 예산과 계획을 검사하고 Peter에게 출장을 허가한다.
③ Peter는 티켓을 구매하고 출장을 간다. 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.

(2) 역할 클래스 다이어그램

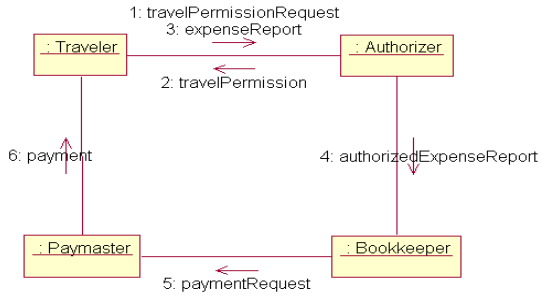
그림 6에서 상호작용하는 객체들은 Peter, Eve, Bill, Joe이다. 이들 객체들을 살펴보면 일정한 패턴을 유지한다고 볼 수 있다. 따라서, 이 객체 패턴을 추상화하여 Traveler, Authorizer, Bookkeeper, Paymaster 역할 클래스를 얻을 수 있다. 다음 그림 7은 역할 클래스 다이어그램을 보여준다.



(그림 7) 역할 클래스 다이어그램(출장 경비 예)

(3) 역할 협력 다이어그램

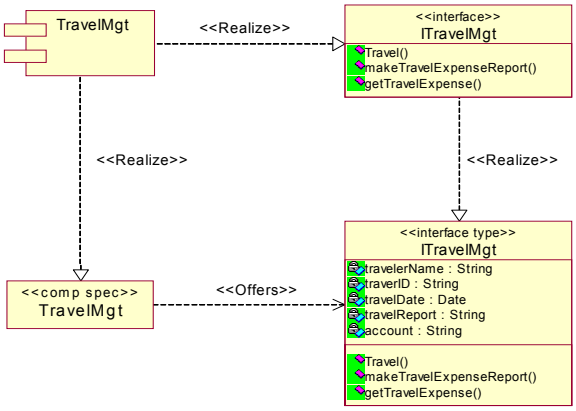
다음 그림 8은 각 역할 클래스들 사이 상호작용하는 관계를 나타내는 메시지 경로를 보여준다.



(그림 8) 역할 협력 다이어그램(출장 경비 예)

(4) 역할 컴포넌트 다이어그램

다음 그림 9는 역할 컴포넌트 다이어그램을 보여준다.



(그림 9) 역할 컴포넌트 다이어그램(출장 경비 예)

5. 결론

본 논문에서는 객체를 중심으로 한 객체 모델링 방법으로 컴포넌트를 추출하지 않고 객체의 역할을 중심으로 한 역할 모델링 방법을 이용하여 컴포넌트를 추출하는데 목적을 두고 있다.

따라서 본 논문에서는 역할 모델링을 통하여 생성된 역할 모델을 UML에서 요구되는 다양한 관점의 다이어그램들로 표현하였다. 역할 모델링은 객체들의 패턴을 추상화하고 복잡한 대규모 시스템을 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. 또한 하나의 역할 모델 단위로 하나의 컴포넌트를 추출할 수 있으므로 컴포넌트 추출이 용이하다.

향후 연구과제로는 UML에서 확장된 컴포넌트 명세 방법이 필요하며 완전한 컴포넌트를 개발하는 것이다.

참고문헌

[1] D. Bäumer, D. Riehle, W. Siberski, M. Wulf, "The Role Object Pattern", In Proceedings of 4th Conference on Pattern Languages of Programs, 1997.

[2] E. P. Andersen, "Conceptual modeling of Object: A Role Modeling Approach", PH.D Thesis, University of Oslo, 1997.

[3] Georg Gottlob, Michael Schrefl, Brigitte Rock, "Extending Object-Oriented Systems with Roles", ACM Transaction on Information Systems, 13(3), pp. 268-296, 1996.

[4] 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.

[5] Ralph Depke, Gregor Engels, Jochen Malte Küster, "On the Integration of Roles in the UML", Technical Report No. 214, University of Paderborn, August, 2000.

[6] The Object Management Group(OMG). "Unified Modeling Language", Version 1.3, OMG, June, 1999.

[7] T. Reenskaug, P. wold, O. A. Lehne, "Working with objects: The OOram Software Engineering Method", Manning/Prentice Hall, 1996.