UML 동적모델에서 컴포넌트 인터페이스 설계 기법

김지혜, 김수동
숭실대학교 컴퓨터학과
jhkim78@selab.soongsil.ac.kr, sdkim@computing.soongsil.ac.kr

Deriving Component Interfaces from UML Dynamic Model

Jee-Hae Kim, Soo Dong Kim
Dept. of Computing, Soongsil University

요 약

컴포넌트는 소프트웨어의 재사용을 통하여 소프트웨어 제품의 향상된 생산성을 제공해 줄 수 있는 방안으로 인식되면서, 학계와 산업계의 관심이 집중되고 있다. 그에 대한 반영으로 컴포넌트 자체를 개발하거나, 상용 컴포넌트를 기반으로 하여 어플리케이션을 개발하기 위한 방법론들이 제안되고 있고, 이런 개발 방법론이 바로 컴포넌트 기반의 개발 방법론이다. 컴포넌트 개발 방법론이 확산됨에 따라 성공적인 컴포넌트 기반의 프로젝트의 중요한 요소로써 효과적인 컴포넌트 인터페이스 설계 기법이 부각되고, 제안되었다. 그러나, 기존의 객체지향 방식으로 구현된 모델을 컴포넌트 기반의 모델로 전환 시의 지침들이 부족하다. 따라서, 본 논문에서는 객체 지향 방식으로 모델링 한 UML 동적모델에서 근거한 컴포넌트 인터페이스 설계 기법을 제안한다. 즉, 분석 단계에서 컴포넌트를 식별하고, 설계 단계의 산출물인 Use Case 모델과 클래스 다이어그램을 이용하여 컴포넌트의 메소드들을 추출한다. 그리고, 컴포넌트 인터페이스를 정의한다.

1. 서 론

컴포넌트 기반의 소프트웨어 개발 기법은 고품질의 소프트웨어를 효율적으로 개발하는 새로운 기법으로 부각되고 있다. 따라서, 학계나 산업계에서는 컴포넌트의 효율적인 생산성에 주목하면서 컴포넌트 관련 방법론 등이 주요 연구 대상이 되고 있다. 이 중 기존의 객체지향 분석/설계에 근거한 컴포넌트 추출기법에 대한 상세한 지침들은 계속해서 제공되고 있기는 하지만 컴포넌트 인터페이스설계에 대한 기법들은 부족한 상태이다. 따라서 본 논문에서는 UML기반의 객체지향 분석/설계 중 동적 모델에서 컴포넌트 인터페이스를 설계하는 기법에 대해 제시하고자 한다.

본 논문의 2장에서는 관련연구로서 CBD와 컴포넌트 인터페이스, 그리고 순차도에 대해서 설명하고, 3장에서는 컴포넌트 인터페이스 설계 기법과 사례연구를 제시한다. 마지막으로 4장에서는 결론을 내린다

2. 관련연구

2.1 Component-Based Development(CBD)

CBD는 소프트웨어의 개발성을 향상시키기 위해서는 기존에 존재하는 소프트웨어 컴포넌트를 사용하여 어플리케이션을 개발하는 것이다. 따라서, 컴포넌트에 기초한 요구사항 분석, 구조, 설계, 구현, 시험, 배치, 기술적인 기반 지원 그리고 프로젝트 관리를 포함하는 개발 생성주기의 모든 단계에 대한 소프트웨어 개발 접근을 의미한다. 이러한 의미는 모든 소프트웨어 개발이 컴포넌트 중심적이라는 컴포넌트 기반의 사고로써 소프트웨어를 생성한다[3].

2.2 컴포넌트 인터페이스

컴포넌트 인터페이스는 오퍼레이션들의 집합이다. 각각의 오퍼레이션은 컴포넌트 객체가 사용가에게 제공하는 몇몇의 서비스나 기능들을 정의한다. 따라서, 오퍼레이션은 사용자와 컴포넌트 객체사이에서의 규약을 나타낸다.

컴포넌트는 컴포넌트의 내부 구현으로부터 인터페이스의 명세를 완전히 분리해야한다. 즉, 컴포넌트의 내부 구현이 변경되더라도, 기존의 컴포넌트 인터페이스 자체를 변경하지 않고도 컴포넌트의 서비스들을 제공한다. 따라서, 컴포넌트 인터페이스는 새로운 컴포넌트를 사용하거나 기존의 컴포넌트를 변경하는 경우, 컴포넌트의 적응력 평가 시간을 감소시키고 테스트를 단순화시킨다[5].

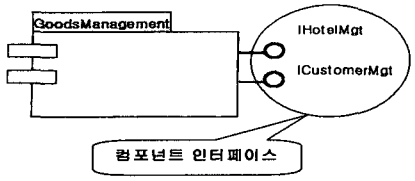

그림 1. 컴포넌트 인터페이스

2.3 순차도와 시스템 순차도

순서도는 클래스 다이어그램에서 나타나는 여러 객체들간에 수행되는 기능을 나타낸다. 이러한 기능들을 수행하기 위해서는 객체들간에 주고 받는 메시지 즉, 객체들간의 상호적으로 호출하게 되는 오페레이션이 시간적 순서에 의해서 나타난다. 따라서, 순차도는 Use Case 다이어그램에서 나타냈던 각각의 Use Case에 대응되어 객체들이 Use Case의 행동을 어떻게 수행하는지를 표현한다[4]. 시스템 순차도(System Sequence Diagram)는 컴포넌트와 액터(Actor)간의 상호작용을 표현하여 상호작용의 방향과 순서가 명시적으로 파악한다.

3. 컴포넌트 인터페이스 설계 기법과 사례연구

본 논문에서는 컴포넌트 인터페이스 설계 기법만 다루기 때문에 다음과 같은 분석 결과를 가정하고, 각 단계별로 간단한 사례를 제시한다. 또한, 컴포넌트는 Required 인터페이스와 Provided인터페이스를 포함하는데, 하나의 컴포넌트가 다른 컴포넌트와 상호작용할 수 있는 방법은 Provied 인터페이스를 통해서이므로, Provided 인터페이스만 다룬다.

인터넷 쇼핑몰 도메인 중 설계단계에서 추출된 물품관리 업무 컴포넌트에 대한 컴포넌트 인터페이스 설계 기법을 적용한 예이다.

물품관리는 쇼핑몰에서 다루는 물품을 추가하고, 검색하고, 삭제하고, 물품의 정보를 갱신하는 업무이다.

컴포넌트를 식별하기 위해서는 다음과 같이 3 단계가 수행된다.

첫째, Use Case 들간에 존재하는 관계를 고려하여 컴포넌트를 후보 컴포넌트를 식별한다. 이 단계에서는 컴포넌트의 핵심적인 비즈니스 기능과 use case들의 범주를 고려하여 후보컴포넌트를 식별한다.

둘째, Use Case와 클래스 사이에 존재하는 관계를 고려하여 각 컴포넌트에 관련된 클래스들을 식별한다.

셋째, 클래스들간에 존재하는 관계를 고려하여 컴포넌트를 정제한다. 객체 모델(association, inheritanace, composition, aggregation, dependency)에 존재하는 클래스들간의 관계를 고려해야하며, 클래스들은 같은 컴포넌트에서 발생되는 composition 관계와 같이 긴밀하게 연결된다.

Use Case 모델과 클래스 다이어그램을 이용하여 그림 2과 같이 GoodsManagement 컴포넌트 한 개가 식별된다.

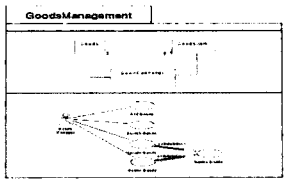

그림 2. 식별된 컴포넌트

GoodsManagement 컴포넌트는 분석결과로 산출되어 Add Goods Use Case, Search Goods Use Case, Update Goods Use Case, Delete Goods Use Case 들과 액터와의 상호작용을 나타내는 Use Case Diagram과 Goods, GoodsItem, GoodsController와 그들간의 관계를 보여주는 클래스 다이어 그램으로 구성된다.

그림 2에서 각각의 Use Case들은 GoodsManagemnent 컴포넌트의 내부 구조를 숨긴채, 제공하는 서비스를 정의한다. 따라서, GoodsManagement 컴포넌트는 Add Goods, Search Goods, Update Goods,Delete Goods의 기능을 제공한다.

3.1 각 컴포넌트별 순차도 작성

식별된 컴포넌트는 여러 개의 Use Case들로 구성된다.하나의 Use Case는 클래스와 같은 식별자에 의해 제공되는 함수의 기능을 나타내는 단위이므로, 내부적 행동과 구현은 숨긴다. 즉, 각 Use Case의 외부액터에게 제공되는 서비스를 명세한다. 따라서, 하나의 Use Case는 주고받는 메시지의 흐름과 상호작용을 나타내는 순차도로 상세화될 수 있다.

따라서, 그림 3은 각각의Use Case에 대한 순차도를 상세히 나타낸다.

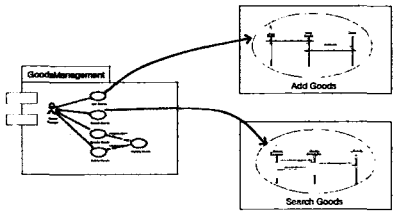

그림 3. GoodsManagement 순차도.

GoodsManagement 컴포넌트는 Add Goods, Search Goods, Update Goods, Delete Goods Use Case로 구성되는데, 이 중에서 Add Goods, Search Goods Use Case에 대해서만 순차도를 작성한다.

3.2 시스템 순차도 작성

순차도의 경우, 설계의 초점이 컨트롤러가 여러 객체에게 어떻게 메시지를 보내고, 받을지에 관한 것이므로 구현시의 API, 즉 컴포넌트 인터페이스와는 차이가 있을 수 있다. 그러

므로, 그림 3와 같이 수집된 순차도들을 가지고 액터와 컨트롤러 사이에서의 실제적인 상호작용을 고려한 시스템 순차도를 그림 4과 같이 작성한다

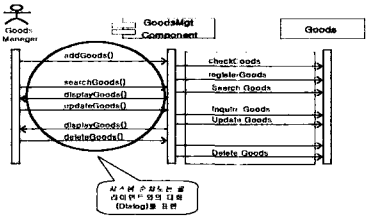

그림 4. 컴포넌트 시스템 순차도

이때에는, 어떠한 API를 가질지를 모델링 하는게 주 목적이므로 컨트롤러 이후에 나타나 시스템 순차도에 참여하는 객체들간의 메시지 흐름은 표현하지 않는다. 순차도에서는 액터와 시스템의 상호작용이 자연어 형식의 메시지 흐름으로 표현되었는데, 시스템 순차도에서는 액터와의 상호작용이 하나 혹은 그 이상의 메시지 흐름으로 구성되어 실제 주고받을 수 있는 함수형식으로 정의된다. 따라서 , GoodsManagement 컴포넌트 Add Goods, Search Goods, Update Goods, Delete Goods 메시지 흐름은 addGoods(), searchGoods(), updateGoods(), deleteGoods()로 정의되고, 액터의 요구인 serchGoods()와 updateGoods() 의 반응하는 displayGoods() 가 실제 외부의 액터와 주고받을 수 있는 함수되어, 정의된다

3.3 컴포넌트 인터페이스 추출

컴포넌트의 인터페이스는 추출된 메소드들의 집합이다. 즉, 분석과정에서 나온 Use Case이용하여 각각의 순차도를 추출할 수 있다. 추출된 순차도는 프로그램 디자이너 입장에서 본 컴포넌트 내부의 흐름도를 나타낸다. 따라서, 컴포넌트 인터페이스는 컴포넌트와 컴포넌트 외부의 사용자간의 상호작용을 나타내고, 추출된 순차도를 기반으로 하여 사용자와의 메시지 형식으로 표현된 상호작용이 시스템 순차도로 표현되면서 함수의 형식으로 바뀌게 된다. 따라서, 이렇게 바뀌어진 함수의 시그네쳐(Signature)가 컴포넌트의 인터페이스로 추출된다.
본 연구에서는 물품관리 기능에 관한 메소드들을 가지고 표 1과 같이 하나의 인터페이스한다.

```
GoodsManagement

interface GoodsMgt {
    status addGoods(String name, int pur-
                chasePrice, int salePrice,
                String company);
    status searchGoods(String goodsID);
    status updateGoods(String goodsID,
                       int count);
    status deleteGoods(String goodsID);
```

표 1. 구현된 GoodsMgt 인터페이스

정의된 GoodsMgt인터페이스는 addGoods, searchGoods, updateGoods, deleteGoodsd의 메소드들로 구성된다. 따라서, 컴포넌트 외부의 사용자는 이 메소드들을 사용하여 컴포넌트를 사용하게 되고, 컴포넌트는 기능에 변화에 따른 메소드 및 인터페이스의 수정과 독립적일 수 있다.

3.4 컴포넌트 인터페이스 검증

본 논문에서는 요구 컴포넌트 식별 활동이 Use Case 모델을 기반으로 이루어졌다는 가정하에 컴포넌트 인터페이스를 설계하였다.

식별된 컴포넌트의 물품관리 Use Case는 액터와의 상호작용을 보여주므로 전체 시스템에서 물품관리 컴포넌트에 대한 기능적 모델링의 결과가 산출된다. 따라서, 이렇게 모델링된 기능들은 액터에게 제공되는 기능들이 때문에, 컴포넌트의 인터페이스 역할을 하게된다. 본 논문에서 제시된 사례에서는, 식별된 GoodsManagement 컴포넌트는 Add Goods, Search Goods, Update Goods, Delete Goods Use Case를 제공하므로, 이 Use Case와 같은 역할을 하는 메소드들인 addGoods(), searchGoods(), updateGoods(), deleteGoods()로 구성된 GoodsMgt 컴포넌트 인터페이스가 추출된다. 또한, GoodsMgt 컴포넌 인터페이스의 메소드들이 GoodsManagement 컴포넌트내의 Goods 클래스의 메소드와 일치됨을 알 수 있다.

4. 맺음말

본 논문에서는 물품관리 시스템을 사례 연구 도메인으로 하여, 순차도의 컴포넌트 인터페이스 설계 기법에 대해서 제안하였다. 본 논문에서 제시된 방법이 모든 순서도에서 컴포넌트 인터페이스를 추출할 수 있는 것은 아니지만, 공용성이나 가변성등을 다룬 순차도를 제외한 순차도의 컴포넌트 인터페이스 추출은 이루어지도록 하였다. 따라서, 향후에는 컴포넌트의 공용성과 가변성 문제를 고려하고 컴포넌트 인터페이스 추출 기법을 제시하고자 한다.

5. 참고문헌

[1] 유영란, "UML기반의 컴포넌트 인터페이스 추출기법", 정보처리학회 논문지, 2000년 4월.

[2] 정승재, "객체지향 분석/설계 기법에 근거한 컴포넌트 추출기법", 숭실대학교 석사학위 논문, 2001년 12월.

[3] Peter Herzum, Oliver Sims, *Busiiness Compo-nent Factory*, Wiley 2001.

[4] Grady Booch, James Rumbaugh, Ivar Jacobson, *UML* Distilled Second Edition, Addision Wesley, 2000

[5] D'Souza D., Cameron Wills A. , *Objects, Compone-*nents and Frameworks with UML, Addison Wesley, 1998