

# 유즈케이스 기반의 컴포넌트 설계 워크플로우 확장 매커니즘

신정은<sup>0</sup> 장용혁 조현훈 류성열  
숭실대학교 컴퓨터학과  
(kikiwa<sup>0</sup>, jazz, hhcho)<sup>0</sup>@selab.ssu.ac.kr, syrhew@computing.ssu.ac.kr

## Expansionary mechanism of Component Design Workflow based on Use-case

Jung-Eun Shin<sup>0</sup> Yong-Hyeok Jang Hyun-Hoon Cho Sung-Yul Rhew  
School of Computing, Soongsil University

### 요 약

컴포넌트 기반의 소프트웨어의 중요성은 널리 인식되어, 국내에서도 컴포넌트화 된 시스템 개발은 이미 큰 조류를 형성하고 있다. 개발된 컴포넌트들을 인터페이스만을 통해 대체가 용이하도록 하기 위해서는 컴포넌트 식별 과정에서 전체 시스템을 관리하는 아키텍처를 구성해야 한다. 본 논문에서 제시하는 컴포넌트 식별은 요구 사항 분석을 통한 유즈케이스 모델을 중심으로 이루어진다. 또한 시스템 아키텍처를 유지하는 컴포넌트 생성을 위해서 비즈니스 계층과 시스템 계층의 산출물들을 파악하여, 컴포넌트를 위한 구성요소를 추출한다. 추출된 구성요소를 기반으로 UML 표기법과 확장 매커니즘을 적용하여 컴포넌트 설계 워크플로우를 모델링 하고자 한다.

### 1. 서 론

소프트웨어의 재사용성을 높이기 위해 부각된 컴포넌트는 생산성을 높이고 유지 보수 비용을 줄이기 위해 개발되고 있다. 현재 개발된 컴포넌트들의 요구 사항들은 계속 변경되고 있으며, B2B 전자상거래와 같은 시스템은 버전 업그레이드를 필요로 하지만, 중앙통제가 어려운 상황이다. 따라서 유지보수 비용의 절감은 컴포넌트의 재사용성만으로는 부족하며, 컴포넌트는 변화에 따라 쉽게 진화하거나 대체할 수 있도록 설계 되어야 한다. [1]

본 논문에서는 객체 지향의 특성인 데이터와 기능의 통합, 캡슐화, 식별성을 기반으로 대체 가능성이 쉽도록 컴포넌트를 식별하고 명세화하기 위한 확장된 워크플로우 정의한다. 또한 이를 바탕으로 의료처방 전달 시스템을 액티비티와 산출물 중심으로 적용하여 모델링 할 것이다.

### 2. 관련 연구

#### 2.1 Workflows Design

계속적이고 동적으로 변하는 환경에서 워크플로우를 디자인 하기 위해서는 액티비티 플로우의 융통성, 정확성, 신속성을 반영한 모델이 필요하다.

워크플로우 디자인의 목적은 액티비티들의 다른 흐름들을 나타내고, 수행해야 할 액티비티들을 제약하지 않는 범위 안에서 가능한 모든 상황들을 잡아내는 것이다.

워크플로우 디자인에서 고려되는 상황들은 전형적인 액티비티 흐름들, 액티비티들과 연관된 문서나 정보 아이탬, 혹은 업체에서 액티비티나 그 역할을 수행하는 관리자 와 관련 된 것들이다.[2]

워크플로우 디자인 시, 워크플로우 프로세스에서 시작점의 반복성, 개발성, 예외상황, 구조 변화에 대한 상황으로 고려되어야 한다.

#### 2.1 UML Components

명세 워크플로우는 세부분으로 나누어 설명할 수 있는데 컴포넌트 식별, 상호작용 분석, 컴포넌트 명세 작성에 대해 설명한다.

명세 워크플로우는 요구사항 정의 워크플로우에서 생성하는 유즈케이스 모델과 비즈니스 개념 모델을 전단계 산출물로서 입력받는다. 명세 워크플로우는 또한 기존 소프트웨어 자산에 관한 정보를 사용하기도 한다. 이단계에서는 컴포넌트 명세와 컴포넌트 아키텍처를 생성한다. 컴포넌트 명세는 컴포넌트가 지원해야 하거나 의존하는 인터페이스 명세와 어떻게 컴포넌트간에 상호작용을 하는지 보여주는 컴포넌트 아키텍처를 포함하고 있다.[1],[3]

### 3. 컴포넌트 설계 워크플로우 확장

소프트웨어 개발 단계 중에서 컴포넌트 개발을 위한 핵심 단계는 크게 세단계로 분류하여 생각할 수 있다. 그림 1에서 문제영역에 대한 요구사항을 분석하는 Business Modeling and Requirements Analysis 단계와 컴포넌트를 식별하고 명세화 하는 Design and Analysis 단계, 명세화 된 컴포넌트에 따라 구현하고 테스트하는 Implementation and Test부분으로 나누어 생각 할 수 있다. 이를 통해 얻어진 결함들은 feedback을 통해 점진적으로 iteration을 수행하여 보완한다.

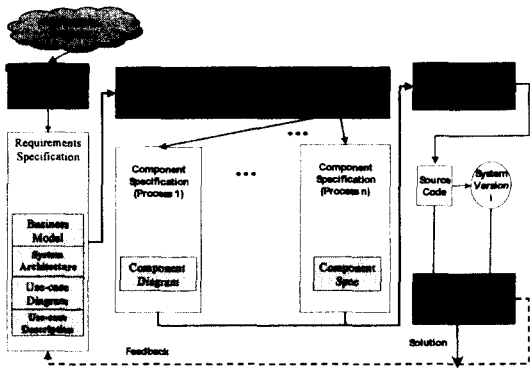


그림 1 컴포넌트 기반의 애플리케이션 개발 워크플로우

본 논문에서는 Business Modeling and Requirements Analysis 단계와 Design and Analysis 단계에서 필요한 산출물을 기준으로 수행하게 되는 액티비티들을 살펴 볼 것이다. 각 액티비티에 따른 산출물은 표 1과 같다.

표 1 액티비티에 따른 산출물 분류

요구사항 분석 모델링	요구사항	비즈니스 개념도 유즈케이스 다이어그램 소프트웨어 아키텍처
컴포넌트 인터페이스 식별	비즈니스 개념도 유즈케이스 다이어그램 소프트웨어 아키텍처	컴포넌트 다이어그램 (시스템 계층) 컴포넌트 다이어그램 (비즈니스계층)
컴포넌트 의존성 분석	컴포넌트 다이어그램 (시스템 계층) 컴포넌트 다이어그램 (비즈니스계층)	컴포넌트 명세서 시퀀스 다이어그램
컴포넌트 명세화	컴포넌트 명세서	컴포넌트 명세서 (Refined)

3.1 요구사항 분석 모델링

Business Modeling과 Requirements Analysis 단계에서 정확한 문제영역의 분석을 위해 우선 시스템이 제공 해야 하는 비즈니스 기능을 나타내는 비즈니스 모델을 그린다. 비즈니스 모델을 바탕으로 컴포넌트가 제공하는 비즈니스 레벨의 서비스를 파악할 수 있으며 이것은 차후에 비즈니스 레벨의 컴포넌트 식별 기준이 된다.

다음으로 사용자가 직접적으로 시스템과 상호 작용하는 기능을 추출하기 위한 유즈케이스 다이어그램과 유즈케이스 디스크립션을 작성한다. 유즈케이스 다이어그램은 시스템 계층의 컴포넌트를 도출하는 바탕이 된다.

또한 시스템 전체적 구조를 파악할 수 있는 시스템 아키텍처를 작성한다. 컴포넌트가 제공하는 기능을 시스템 레벨과 아키텍처 레벨로 나누어 각 계층별로 분류한다. 소프트웨어 아키텍처는 컴포넌트 식별단계에 적용하여, 시스템 아키텍처의 레벨을 준수할 수 있도록 한다.

3.2 컴포넌트 인터페이스 식별

컴포넌트의 식별은 Business Modeling 과 Requirements 에서 나온 산출물을 바탕으로 이루어진다.

컴포넌트의 인터페이스는 시스템 계층의 인터페이스와 비즈니스 계층의 인터페이스로 나누어 진다. 시스템 계층의 인터페이스는 유즈케이스를 통해 추출하고, 비즈니스 계층의 인터페이스는 비즈니스 다이어그램을 통해 추론한다.

먼저 시스템 계층의 컴포넌트를 식별하는 과정을 살펴 보자. 유즈케이스 디스크립션의 각 스텝을 통해 시스템이 사용자에게 직접적으로 제공하는 기능들을 하나씩 추론한다. 유즈케이스를 바탕으로 식별되는 컴포넌트는 하나의 유즈케이스 범위를 넘지 못 할 수 있다. 만약, 다른 유즈케이스 내에서도 공통적으로 사용되는 기능이 있다면 이를 하나의 컴포넌트로 만드는 것이 바람직하다. 이를 가능하게 하는 것이 비즈니스 모델이다. 유즈케이스는 사용자 관점에서 시스템이 사용자에게 제공하는 기능을 이미 독립적으로 분류하여 정의 한 것이다. 따라서 다른 유즈케이스들 사이에서 공통적으로 제공하는 기능은 시스템 내부적으로 비즈니스 레벨에서 제공하는 기능들이라는 것을 알 수 있다. 비즈니스 모델은 유즈케이스로 기능을 분류하기 이전의 시스템 전체적인 개념들을 바탕으로 도식화 한 것이다. 시스템 레벨에서 공통적으로 사용되는 기능들을 추론해 낼 수 있다.

3.3 컴포넌트 의존성 분석

컴포넌트의 인터페이스를 통해 컴포넌트간의 의존성의 분석하여 해당 인터페이스 내에 오퍼레이션을 추출 한다.

먼저, 시스템 레벨의 컴포넌트 인터페이스의 오퍼레이션을 추출하기 위해 사용자가 시스템에 요구한 트랜잭션의 결과값이 사용자에게 전달되기까지의 이벤트 흐름을 파악한다. 이 때 유즈케이스를 바탕으로 시퀀스 다이어그램을 작성한다. 시퀀스 다이어그램에서 시스템 바운더리 클래스가 호출하는 컨트롤 클래스가 포함하는 메소드들을 바탕으로 시스템 컴포넌트 인터페이스의 오퍼레이션을 구성한다.

시스템 컴포넌트와 비즈니스 컴포넌트의 의존성을 파악하기 위해 시퀀스 다이어그램을 통해 호출되는 비즈니스 컴포넌트의 오퍼레이션들을 추출한다. 또한 이전의 비즈니스 모델을 바탕으로 오퍼레이션과 비즈니스 컴포넌트의 인터페이스를 정제한다.

3.4 컴포넌트 명세화

다음으로 인터페이스만을 통해 컴포넌트를 사용하고 대체하기 위해서는 명확한 명세서가 필요하다.

컴포넌트와 인터페이스의 명명은 CBD96에서 제시하는 표준에 따르며, 각각의 컴포넌트들을 명확하게 식별 할 수 있도록 한다.

4 사례연구

본 논문에서 제시된 워크플로우 대로 인터넷 처방전 시스템의 컴포넌트를 디자인 해 보았다. 인터넷 처방전의 요구사항을 분석한 유즈케이스 다이어그램은 그림3과 같다.

그림 3에서 “처방전을 전송한다”의 유즈케이스 디스크립션을 바탕으로 컴포넌트의 인터페이스를 추출하는 과정은 그림 4와 같다.

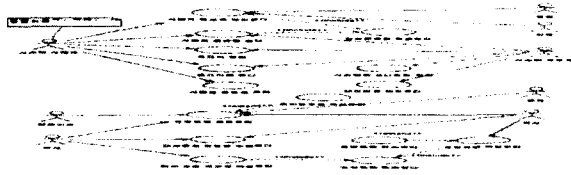


그림 3 인터넷 의료 처방 전달 시스템의 유즈케이스 다이어그램

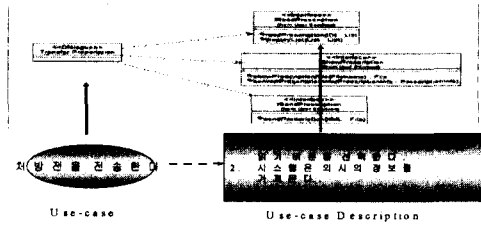


그림 4 시스템 계층의 컴포넌트 인터페이스 도출

그림 4에서 “처방전을 전송한다”라는 유즈케이스는 의사가 시스템 화면에 “처방전 전송”이라는 전송버튼을 선택하는 순간부터 시작되며, 처음으로 수행하는 기능은 처방전을 읽어오는 것이다. 이 때, 처방전 읽기 기능을 인터페이스로 도출해 낸다.

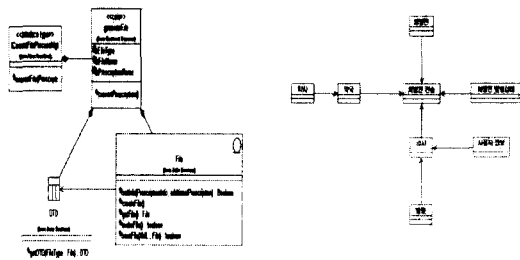


그림 5 처방전 전송과 관련한 비즈니스 다이어그램

그림 5는 처방전 전송과 관련한 비즈니스 개념도이다. 다음으로 시스템 컴포넌트 인터페이스의 오퍼레이션을 도출하기 위해 처방전 전송의 유즈케이스 디스크립션을 바탕으로 그림 5와 같이 시퀀스 다이어그램을 그린다. 처방전 읽기 이벤트의 트랜잭션을 하나의 인터페이스 `IreadPrescription`로 매핑하고 하나의 트랜잭션을 처리하는 흐름 중 시퀀스 다이어그램의 컨트롤 클래스에 포함되는 메소드가 인터페이스가 제공하는 오퍼레이션으로 추출한다. 다음 그림 6을 통해 `IreadPrescription` 인터페이스의 오퍼레이션은 `readPrescriptionList(ID)`, `displayList(List)`가 됨을 추론할 수 있다. 그림 5의 비즈니스 다이어그램에서 도출된 비즈니스 컴포넌트에 대해 그림 6의 시퀀스 다이어그램에서 호출하게 되는 경우 시퀀스 다이어그램에서 호출되는 비즈니스 컴포넌트 인터페이스의 오퍼레이션을 추출해 낼 수 있다.

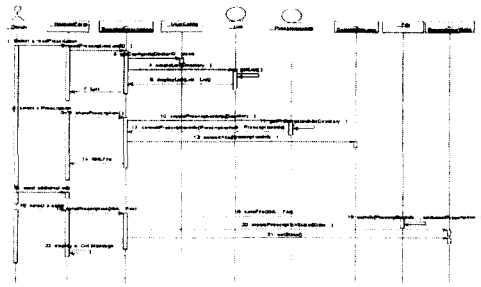


그림 6 시나리오에 따른 시퀀스 다이어그램

그림 7은 이를 바탕으로 식별된 시스템 컴포넌트와 비즈니스 컴포넌트이다.

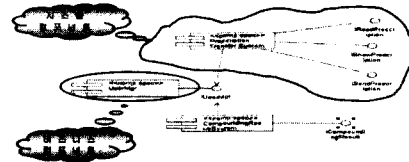


그림 7 컴포넌트 다이어그램

5. 결론 및 향후 연구 과제

시스템 계층과 비즈니스 계층을 유지하며, 컴포넌트 식별과 명세를 위한 컴포넌트 워크플로우를 진행하였다. 유즈케이스 기반으로 진행된 컴포넌트 식별은 유즈케이스의 크기에 종속적일 수 있다. 따라서 우선적으로 유즈케이스가 명확하게 도출되어야 하며, 유즈케이스 간 공통적으로 포함하는 기능을 컴포넌트화 하기 위해 비즈니스 다이어그램을 통해 컴포넌트를 식별하고 명세화하였다.

향 후, 컴포넌트 디자인 시 적용 될 수 있는 컴포넌트 사이즈에 대한 명확한 지침이나 기준을 보완할 것이며, 컴포넌트 모델링에 따라 EJB 기술을 적용하여 매핑되는 기법에 대한 연구를 진행할 것이다.

6. 참고 문헌

- [1] John Cheesman, UML Components, p28 Addison Wesley, 2000
- [2] Mariagrazia Fugini, Using Patterns to Design Rules in Workflows, IEEE Transaction on Software Engineering, Vol.26, No. 8, P760, 2000
- [3] Desmond Francis DSouza Component-Based Development Using Catalysis , 1997.