레거시 시스템에서 공통 클래스를 통한 컴포넌트 도출 방법

이종민°
고려대학교 소프트웨어공학과
mos9284@korea.ac.kr°,

Method of Identifying Component in Legacy System through Common Class

Jong-Min Lee° ,
Dept. of Software Engineering, Korea University

요 약

레거시 시스템을 컴포넌트화 하기 위해 시스템을 서브 시스템으로 계층화하고, 각각의 서브 시스템을 객체 기반으로 변형한 후, 래퍼(Wrapper)를 이용하여 컴포넌트화 한다. 이런 절차 중 Wrapper컴포넌트를 도출하는 방법 중 UML Component방법론을 사용, 컴포넌트 도출 중 여러 핵심타입(Core Type) 객체가 하나의 객체와 연관관계를 가지고 있는 경우 경험이나 직관을 최소화하며 의존성을 최소할 수 있는 개선된 컴포넌트 도출방법을 제안한다.

1. 서 론

컴포넌트 기반 개발은 시스템의 전체를 각각의 독립적인 컴포넌트로 나누어서 개발한 다음 이것을 통합하여 어플리케이션을 구축하는 방법이다. 컴포넌트 기반 개발은 개발 기간과 비용의 단축을 가져오기 때문에 산업계에서는 이러한 개발방법을 도입하려고 노력하고 있다[1]. 컴포넌트 기술은 객체지향의 목표와 유사하게, 재사용을 증진시켜 저비용으로 신뢰성 있는 소프트웨어 개발 및 사용을 지향한다. 작은 단위의 재사용 개체로는 효율적으로 작업하기 어려우나, 수많은 클래스와 객체들로 구성된 컴포넌트와 같은 보다 큰 단위의 도메인 기반 접근법을 이용하면, 이해나 유지보수가 어려운 대규모 시스템을 구조화, 이용, 개발하는 데 향상된 접근법을 제공해 준다[2]. 하지만, 산업체에서는 레거시 시스템을 그대로 사용하고 있다. 왜냐하면, 기업에서 사용하고 있는 현 시스템이 가장 안정적이기 때문이다. 그러나, 레거시 시스템은 급변하는 업무환경에 적합하게 대처할 수 없는 경우가 대부분이다[3]. 이러한 레거시 시스템에 대한 문제점을 해결하려는 많은 전략 중에는 레거시 시스템과는 관계없이 처음부터 현 상황에 알맞은 시스템을 개발하는 방법도 있다. 그러나 레거시 시스템은 차세대 시스템과 통합되어 새로운 기능을 생성해 낼 수 있는 자산으로서 레거시 시스템을 폐기하는 방법은 적절하지 않다. 따라서, 레거시 시스템의 의미 있는 정보(Semantics)를 추출해내는 것이 중요하다[4]. 이에 기존의 레거시 환경은 그대로 유지하면서 Wrapper컴포넌트를 이용하여 레거시 환경을 컴포넌트화 하는 방법들이 연구되고 있다. 본 논문에서는 이러한 절차 중에 UML Component 방법론에서 사용되는 컴포넌트 도출 방법 중 객체 간 연관관계(Association)에서 기존 방식보다 효율적인 도출방법을 제안하고자 한다.

이 논문의 구성은 다음과 같다. 2장에서는 기존에 레거시를 컴포넌트화 하는 방법과 UML Component 방법론의 문제점을 살펴보고, 3장에서는 기존 문제점을 해결하는 방법을 제안한다. 4장에서는 제안된 연구방법의 검증 및 기존 연구방법과 비교, 분석을 보이며, 5장에서는 결론 및 향후 과제에 관해 기술한다.

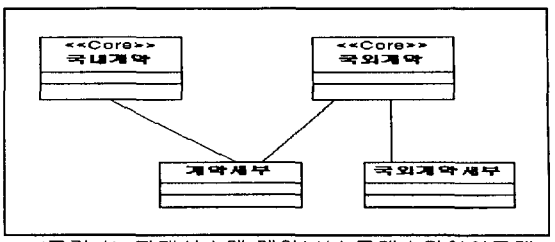
2. 기존 연구

레거시 시스템을 변환하여 외부시스템과 연동을 하는 방법에는 여러 가지 방법이 있다. 첫째, 기존 레거시 시스템을 일부 변경하여 외부시스템과 연동할 수 있도록 변경하는 방법. 둘째, 레거시 시스템의 변화를 최소화 하기 위해 Wrapper컴포넌트나 Adapter컴포넌트를 이용하는 방법 셋째, 자바 기반의 기술인 JNI(Java Native Interface)를 이용하여 자바 외에 언어로 구성된 시스템과 연동하는 방법 등이 있다. 이중에서 두 번째 방법인 레거시 시스템을 Wrapper컴포넌트나 Adapter컴포넌트를 이용하여 레거시 시스템을 컴포넌트화 하고 이를 외부 시스템과 연동하는 방법이 산업계에서는 가장 널리 사용 된다. 그 이유는 기존 레거시 시스템의 변화를 최소화 하여 연동이 가능하며, 추후 외부시스템의 변경이 발생하는 경우에도 컴포넌

트만이 변경이 될 뿐, 레거시 시스템의 변경은 없기 때문이다. 기존의 레거시 시스템을 컴포넌트화 하는 절차는 다음과 같다. ISA(Identification of Subsystems based on Association) 방법을 이용하여 레거시 시스템을 서브 시스템으로 계층화하고, 각각의 서브 시스템을 객체 기반으로 변형한 후, 래퍼를 이용하여 컴포넌트화 한다[2]. ISA의 절차 중 추출된 객체를 기반으로 컴포넌트를 도출하는 방법에는 여러 가지 방법이 있으나, 본 논문에서는 UML Component를 중심으로 진행한다. UML Component 방법론은 CBD96 방법론과 컴포넌트 식별 방법은 유사하지만, CBD96에서의 컴포넌트 식별 방법과 다르게 시스템 컴포넌트와 비즈니스 컴포넌트의 개념을 명확히 분리하여 적용하고 있다. 컴포넌트 식별은 시스템의 전체 도메인을 중심으로 타입 다이어그램과 비즈니스 타입 다이어그램을 추출하고 추출된 핵심타입(Core Type)을 중심으로 관련된 타입을 그룹화하여 비즈니스 컴포넌트를 도출한다. 동시에 요구사항을 분석을 통하여 시스템 컴포넌트를 추출한다. 그 이후에 인터렉션 다이어그램을 의해 추출된 비즈니스 컴포넌트를 정제하고 컴포넌트 아키텍쳐를 정의하기 위하여 추출된 시스템 컴포넌트에 포함될 비즈니스 컴포넌트를 찾아 조합한다[5]. 하지만, CBD96과 UML Component 방법론의 공통적인 문제점은 현업 담당자가 자기의 업무에 대해서 의미 있는 개념을 추출하고 그들 사이의 연관관계를 나타냄으로 타입 다이어그램을 그리는데 소프트웨어에 대한 아무 개념이 없는 현업 담당자가 어떠한 단어가 의미가 있는지 없는지 구별하기가 쉽지 않고 그들 사이에 관계성을 도출하기가 쉽지 않다[5].

3. 공통클래스의 컴포넌트 도출 기준

<그림 1>은 자재시스템 구축에서 비즈니스 타입 다이어그램으로부터 나온 객체모델이다. 자재시스템의 기본적인 구성은 구매와 계약으로 볼 수 있는데, 계약의 경우 국내계약과 국외계약으로 나눌 수 있다. 계약의 내용에는 공통적인 사항과 세부적인 사항이 있으며 대부분의 경우 이것을 분리하여 나타낸다. 국내계약과 국외계약의 핵심적인 내용을 각 핵심 클래스로 도출 하였고 거기에 따른 세부사항을 클래스로 도출하였다. <그림 1>의 계약세부 클래스가 하나의 클래스로 도출된 배경은 국내와 국외계약의 세부사항이 비슷하기 때문에 하나의 클래스로 도출하였고, 국외의 경우 환율 정보 등 추가적인 세부사항을 하나의 클래스로 도출하게 되었다. 이 경우 핵심클래스를 중심으로 컴포넌트를 도출 시에 세부계약 클래스를 어느 핵심 클래스의 그룹으로 두느냐 하는 문제가 발생한다. 이는 업무 관점에 따라 공통 클래스가 포함 되

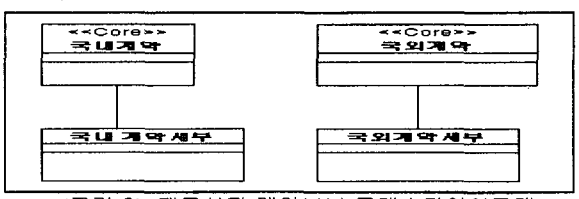


<그림 1> 자재시스템 계약부분 클래스다이어그램

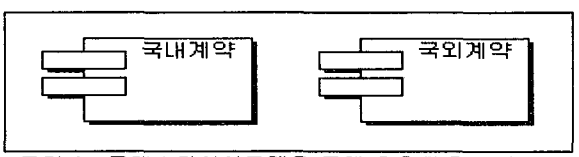
어야 할 핵심클래스가 달라지기 때문이다. 결국 개발자의 주관적인 경험에 따라 많은 시간의 토의와 의견을 거쳐 컴포넌트가 도출되게 된다. 또한, 이런 과정을 걸쳐 도출된 컴포넌트는 의존성이 많아지기 때문에 재사용성에 많은 제약을 가지게 된다. 이런 문제를 해결하기 위해 여러 핵심클래스와 연관관계를 가진 클래스를 각 핵심클래스에 복사하는 방법을 제안한다. 이후로는 여러 핵심클래스로부터 연관관계를 갖는 클래스객체를 공통클래스(Common Class)라 명한다. 아래와 같은 2가지의 기준을 공통클래스 복사 경우로 제안한다.

① 공통객체가 상속, 집합 관계를 가지는 경우는 제안방법에서 제외한다.
② 핵심클래스와 연관관계를 갖는 클래스가 공통클래스 외에 공통클래스와 유사한 성격을 가진 클래스가 있다면 공통 클래스와 합친다.

위의 2가지 제안방법에 따라 계약세부 클래스를 복사하여 국내계약, 국외계약 클래스에 국내계약세부 클래스, 국외계약세부 클래스로 나타낸다. 다음에 국외계약세부 클래스는 기존에 있던 국외계약세부 클래스와 병합, 새로운 국외계약세부 클래스로 생성한다. 제안방식에 따라 <그림 1>의 클래스다이어그램은 <그림 2>와 같이 재구성이 되며, <그림 2>의 클래스다이어그램을 통해 컴포넌트를 추출 하면 <그림 3>과 같이 나타남을 알 수 있다. UML Component 방법론을 이용한 컴포넌트 도출에서 공통 클래스에 관한 그룹화를 공통 클래스를 각 각의 핵심클래스에 복사함으로써 개발자나 업무 전문가의 주관적인 판단에 의해 컴포넌트가 도출되는 것을 최소화 하고, 컴포넌트간의 의존성을 최소화 함으로서 컴포넌트의 재사용성을 최대화 한다.

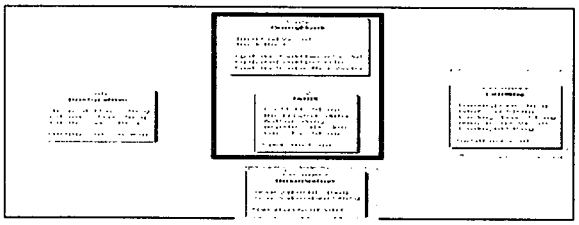


<그림 2> 재구성된 계약부분 클래스다이어그램

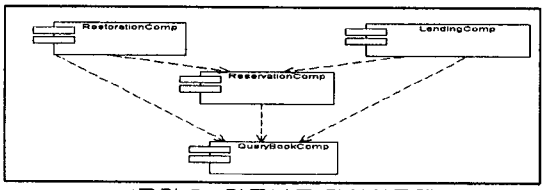


<그림 3> 클래스다이어그램을 통해 추출된 컴포넌트

4. 분석 및 검증

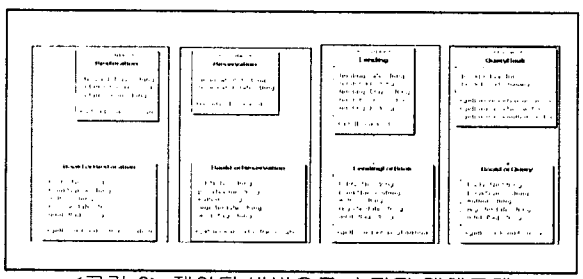


<그림 4> 도서대여 시스템 클래스다이어그램

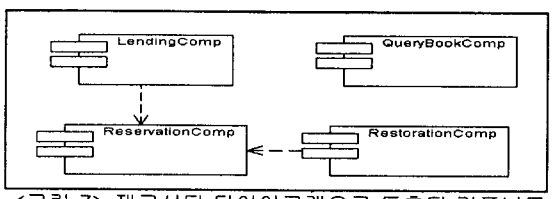


<그림 5> 컴포넌트 다이어그램

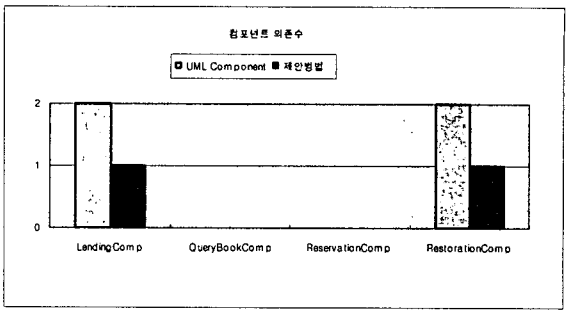
간단한 도서 대여시스템을 통해 제안방법을 검증하고자 한다. 도서 대여시스템은 도서 대여, 반환, 예약, 검색의 4가지 기능을 가지며, 시스템의 객체 모델을 단순화 시켜 하나의 기능에 관하여 하나의 객체로 표현한다. <그림 4> 에서와 같이 클래스 다이어그램에서 핵심클래스는 QueryBook, Reservation, Lending, Restoration 4개로 구성되어 있다. <그림 4>의 클래스다이어그램을 이용 Book 클래스를 QueryBook 클래스의 그룹에 포함시키는 것이 가장 합리적이라고 가정 후, 컴포넌트를 도출하면 <그림 5>와 같이 도출된다. 제안된 방법을 사용하는 경우 Book 클래스를 BookForLending, BookForRestoration, BookForQuery, BookForReservation으로 각 핵심 클래스에 복사한다. <그림 5>의 클래스 다이어그램을 제안 방법으로 재구성을 하면 <그림 6>과 같이 구성이 되며, 컴포넌트는 <그림 7>같이 나타난다. <그림 5>와 <그림 7>을 비교하면 컴포넌트의 개수는 동



<그림 6> 제안된 방법으로 수정된 객체모델



<그림 7> 재구성된 다이어그램으로 도출된 컴포넌트



<그림 8> 컴포넌트 의존개수

일하지만, 각 컴포넌트가 참조하는 컴포넌트 수를 비교하여 보면, <그림 8>과 같이 기존 방법 보다 제안방식의 컴포넌트 의존수가 적음을 알 수 있다.

5. 결론 및 향후 연구계획

본 논문에서는 레거시 시스템을 컴포넌트화 하는 과정에서 사용되는 Wrapper 컴포넌트의 도출방법을 UML 컴포넌트 방법론을 이용한다. 하지만 UML Component방법론에서 여러 개의 핵심클래스에 하나의 클래스와 연관관계를 갖는 경우 어느 핵심클래스 그룹에 포함시켜야 하는 문제가 발생된다. 따라서, 이러한 연관관계에서의 문제점을 최소화 하고자 많은 연관관계를 가지고 있는 클래스를 각 각의 핵심클래스에 복사하여, 핵심클래스를 기준으로 연관된 객체들의 그룹화를 명확히 하여 컴포넌트 도출 시 객관적인 컴포넌트 도출 방법 및 의존성이 높은 컴포넌트 도출방법을 제시하였다. 향후에는 공통클래스가 상속관계나 집합관계를 가지고 있을 때에도 핵심클래스를 기준으로 효율적인 컴포넌트 도출방식을 연구 할 것이다. 또한 본 제안 방식을 대규모 시스템에 적용하여 효율성을 입증 할 것이다

참고문헌

[1] 박인근, 김수동," 소프트웨어 컴포넌트 재사용성 측정 메트릭" , 정보과학회논문지 31권 6호 760p, 2004년
[2] 이은주, 신우창, 이병정, 우치수," 객체지향 모델로부터 정적 메트릭을 이용하여 컴포넌트 기반 시스템으로 변환하는 기법" , 정보과학회논문지 31권 6호 728p~729p, 2004년
[3]P. Aiken, A. Muntz, and R. Richards, " A Framework for Reverse Engineering DoD Legacy Information Systems," Proceedings Working Conference on Reverse Engineering, IEEE Computer Society Press, pp. 180~191, 1993.
[4] 이창목, 유철중, 장옥배, " 객체 재사용성 향상을 위한 레거시 시스템 인터페이스 기반 객체추출 기법" , 정보과학회논문지 31권 6호 1455p~1456p, 2004년
[5] 최미숙, 윤용익, 박재년," 컴포넌트 식별 방법에 관한 비교 연구" , 한국컴퓨터산업교육학회 3권 3호 385p~386p, 2002년