

# 업무중심의 애플리케이션 설계 방법 중 클래스 설계 절차에 관한 연구

김태우\*, 박선이\*, 여정모\*  
\*부경대학교 컴퓨터공학과  
e-mail : mtkim7895@pukyong.ac.kr,  
psunyi@pknu.ac.kr,  
yeo@pknu.ac.kr

## A Study on the Class Design Procedure of Application Design Method in the Business Based

Tae-woo Kim \*, Sun-yi Park \*, Jeong-mo Yeo \*  
\*Pukyong National Univ, Dept. Of Computer Eng

### 요 약

하나의 애플리케이션은 기업의 업무를 분석하여 어떻게 만들지 설계를 하고, 설계를 바탕으로 설계의 내용을 그대로 개발로 구현하게 된다. 설계의 내용을 그대로 개발로 반영하기 위해 소스 코드에 대한 설계도 이루어 져야 한다. 다양한 방법에 대해 연구되어 왔지만, 경험기반의 설계가 반영되어 설계자 및 개발자 간에 의사소통의 오류가 발생하고, 미경험자가 설계절차를 수행하기에는 어려움이 존재한다. 이를 해결하기 위해 본 연구에서는 분석된 업무를 바탕으로 정의된 화면설계를 바탕으로 클래스로 변환되는 절차를 제안하여 초심자가 효율적으로 설계절차를 수행할 수 있을 것이며, 개발자 간에 원활한 의사소통이 이루어질 것이라 기대한다.

### 1. 서론

정보사회의 발전과 함께 기업에서는 기업에서 운영 하는 하나 이상의 애플리케이션은 필수적으로 사용되고 있다. 애플리케이션에 기업의 모든 업무를 잘 담아내고 사용하는 것은 매우 중요해지며, 애플리케이션을 어떻게 만드는지에 따라 기업의 수익을 증진시키는데 뒷받침 역할을 할지 아니면 손해의 책임이 될지 결정된다[1]. 애플리케이션을 개발하기 위해서는 크게 설계와 개발 두 단계로 나뉘어 수행한다. 모든 과정이 중요하지만 대부분 설계에 많은 비용과 시간이 투자되며 설계에 따라 애플리케이션의 완성도가 결정된다는 것과 다름이 없기 때문에 설계에 많은 비중을 두고 있다.

애플리케이션은 설계된 내용을 바탕으로 개발하므로 설계된 내용을 그대로 적용하여 개발하는 것이 중요한 문제가 된다. 그러기 위해서는 설계한 내용을 개발에 필요한 소스 코드로 변환되는 설계도가 필요하다[2]. 또한 다양한 인원이 투입되는 프로젝트에서 공통적인 개발을 위한 기준이 제시되어야 한다. 이러한 문제를 충족시키기 위해 애플리케이션을 만들기 위한 설계 방법론과 효율적으로 진행하는 절차들이 많이 연구되어 왔다[3]. 대표적인 개발 방법론들이 연구되고 현실에 맞게 새로운 방법론이 등장하면서 많은 기업들이 여러 방법을 적용 및 수행을 하고 있다 [4,5]. 하지만 연구된 개발 방법론은 경험과 노하우가

설계의 많은 부분을 차지하고 있어 처음 설계를 접하는 미경험자들이 설계하기에는 어려움이 존재한다.

본 연구에서는 이러한 문제를 해결하기 위해 업무 분석기반으로 연구된 선행연구인 애플리케이션 설계 방법들을 바탕으로 화면설계를 소스 코드로 변환하는 클래스 설계방법을 제안한다.

### 2. 관련 연구

#### 2.1. 객체지향 분석 기법

객체지향 분석 기법에서 클래스 다이어그램은 시스템의 정적인 정보 구조를 나타내며 시스템에 필요한 클래스들과 이들 사이의 관계를 나타낸다[6,7]. 클래스는 객체의 특성을 나타내는 여러 가지 속성들과 메소드들로 구성이 된다. 클래스 다이어그램은 유스케이스 시나리오나 문제 기술서를 바탕으로 클래스를 도출한다. 클래스 다이어그램을 도출하는 절차는 다음과 같다. 도출된 유스케이스 시나리오를 바탕으로 사용자와 상호작용을 담당하는 사용자 인터페이스 클래스를 도출한다. 데이터 클래스는 시스템의 데이터에 접근하여 수정, 조회, 등록, 삭제 등의 기능을 수행하는 클래스로 하나의 사용자 인터페이스 클래스는 하나 이상의 데이터 클래스를 가진다. 클래스들을 도출한 후 클래스 간의 관계를 찾아 연결해 준다. 이후 유스케이스 시나리오를 기반으로 모든 속성과 메소드들을 도출한다.

2.2. 문장기반 클래스 도출 기법

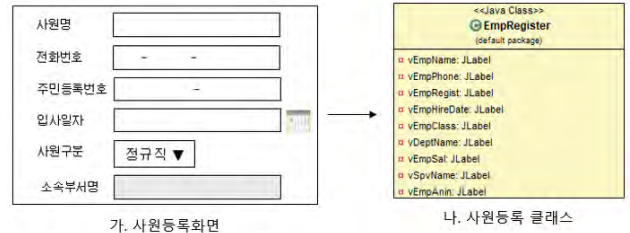
클래스를 도출하기 위한 방법으로 잘 알려진 기법뿐만 아니라 다양한 방법이 존재한다[8,9]. 전체적인 설계 절차 중 클래스를 도출하는 방법은 각각도로 분석하고 수행이 많이 걸리는 어려움이 존재한다. 이를 해결하기 위해 도출된 업무에서 클래스만을 도출하기 위한 방법들도 연구되어져 왔다. 요구사항 정의서나 업무 기술서를 바탕으로 명사, 대명사, 명사구 등을 우선적으로 도출하고 클래스 후보 목록을 정의한다. 그다음 설계자의 경험을 바탕으로 의미를 분석하여 중복되거나 클래스가 아닌 후보들을 제거해 클래스를 확정해 나간다. 확정된 클래스는 클래스간의 관계를 파악하여 해당 업무의 클래스 다이어그램을 확정해 나간다.

클래스를 정의하는 데 있어 논리적인 표현을 물리적인 표현으로 변환하는데 프로그래머의 성향에 따라 의사소통의 어려움과 미경험자가 상세한 클래스를 도출하는데 어려움이 존재한다. 본 연구에서는 하나의 화면을 하나의 클래스로써 정의를 가지며, 화면의 구성요소 및 설명들이 클래스 다이어그램으로 변환되는 대응 정보를 제시하여 이러한 문제를 해결하고자 한다.

3. 클래스 설계 절차

정보시스템이 완성되기까지 많은 절차를 수행하게 된다. 설계를 거쳐 프로그래머가 실질적으로 시스템을 구현하기 위해 소스코드를 작성하여 프로그램을 완성하게 된다[10]. 설계단계에서 기업의 업무를 분석하여 정보시스템을 어떻게 구성할 것인지 구조를 정의하고, 정의한 내용을 그대로 가져와서 애플리케이션에 녹여내야 한다. 그러기 위해서는 애플리케이션의 구조뿐만 아니라 코드를 위한 설계도 이루어져야 한다. 하나의 정보시스템을 만들기 위해서는 다양한 사람들이 참여하기 때문에 각각의 경험과 습관에 따라 같은 의미의 내용을 다른 방식으로 개발을 할 수 있다. 그렇기 때문에 하나의 기준을 내세워 동일하고 유사하도록 개발이 이루어지게 하기 위해 코드 작성에 대한 설계가 이루어져야 한다. 본 연구에서는 객체지향방식기반으로 이루어지는 프로그램을 개발하기 위해 클래스를 설계하는 클래스 설계에 대해 다루고자 한다.

본 연구에서는 "상품주문시스템" 요구사항[11]을 바탕으로 분석된 요구사항에 따라 애플리케이션의 절차인 업무 프로세스가 도출되고, 이를 활용한 화면설계를 바탕으로 클래스로 전환하고자 한다. 도출된 화면설계를 화면에 표현되는 디자인, 화면에서 실질적으로 수행되는 절차인 이벤트, 화면에서 사용되는 데이터 세 부분으로 나누어서 변환한다.



(그림 1) 사원등록의 화면 변환 중 일부

화면에 표현되는 요소들은 각각 하나의 컴포넌트로서 변환이 이루어진다. 그림 1은 사원등록 화면의 디자인요소가 클래스 설계로 변환된 모습을 보여주며, 왼쪽 '가'의 사원등록의 화면에서 사원명은 '나'의 클래스에서 레이블로 변환되며 입력창은 텍스트 필드로 변환이 이루어진다. 단순 컴포넌트는 하나의 컴포넌트변수로 변환이 되며 테이블과 같은 컴포넌트는 컬럼 문자배열과 테이블 컴포넌트가 복합적으로 변환이 이루어진다. 이와 같이 설계된 컴포넌트는 해당 컴포넌트 객체로 변환이 이루어지며 컴포넌트에 필요한 컴포넌트변수들로 변환된다.

```

2. <소속부서 선택> = {[ 부서부정번호1]-L: <{사원부정번호1}.{소속부서번호, 소속부서명} = { 부서부정번호1}.{부서번호, 부서명}-sys:
3. <정규직?><직속상사 선택> = <{정규직}>{<직속상사부정번호1}>-L: <{사원부정번호1}.{직속상사사번, 직속상사명} = {직속상사부정번호1}.{사원번호, 성명}-sys:
4. [등록] > 사원등록처리({사원부정번호1}): > 오류객체명?오류객체명({사원부정번호1}) > 오류객체명?사원입력 오류": "사원등록 저장":
    
```

가. 사원등록화면

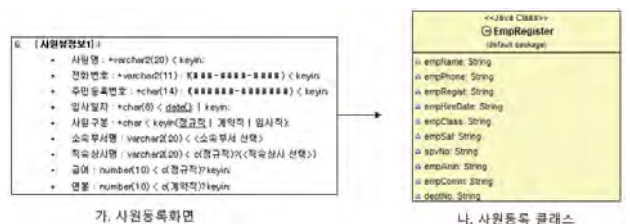
```

@ actionPerformed(ActionEvent):void
▲ getDeptData(List<Map<String,Serializable>->):void
▲ getSvpData(List<Map<String,Serializable>->):void
▲ setData():void
▲ objectCheck():int
▲ showWindow(int):int
    
```

나. 사원등록 클래스

(그림 2) 사원등록의 이벤트 변환 중 일부

애플리케이션에서 업무의 수행을 나타내는 이벤트는 버튼을 클릭했을 때 동작하거나 테이블의 행을 선택했을 때 동작 등으로 이루어져 있다. 이벤트가 발생했을 때 수행되는 동작으로서는 함수를 호출하여 수행하거나 특정 데이터를 데이터베이스 또는 다른 화면/창으로 전달하는 과정들이 있다. 자바에서는 버튼, 마우스, 키보드 등의 이벤트 리스너가 이벤트 발생을 알리며 이를 각각 필요한 메소드를 호출한다. 각 메소드는 함수를 수행하거나 정보전달과 같은 행위가 이루어진다. 그림 2는 사원등록화면에서의 이벤트 요소가 클래스설계로 변환된 모습을 보여주며, 위쪽 '가'의 화면설계 화면에서 설계된 각각의 이벤트 내용이 '나'의 클래스와 같이 리스너를 통해 필요한 메소드를 호출하도록 변환되어진 모습을 보여준다.



(그림 3) 사원등록의 데이터 변환 중 일부

화면설계에서 데이터는 업무수행에 있어 전달되는

내용을 나타낸다. 화면에 표현되는 데이터는 데이터베이스에 영구적으로 저장되어 전달되는 데이터뿐만 아니라 임시적으로 업무수행에 필요한 데이터도 포함된다. 이러한 데이터는 클래스에서 단순 변수들로 일대일 대응되어 변환된다. 지속해서 사용되는 데이터는 데이터베이스와 연관되어 읽기, 쓰기, 변경, 삭제 작업을 수행하므로 애플리케이션과 데이터베이스 연결을 위한 변수들도 생성하게 된다. 그림 3 은 사원등록의 화면에서 나타나는 데이터가 클래스에서 일반 변수로 변환되는 모습을 보여준다.

```

* [[부서정보]] : {{*부서번호, *부서명_asc1, 부서위치, *전화번호, 부서장구분, 정규직부서장시번, 계약직부서장시번}}
* [[부서부정보1]] : {{*부서번호, *부서명_#_asc1, 부서위치#, *전화번호, 부서장구분, 정규직부서장시번, 계약직부서장시번}} <= [[부서정보]]
* [[직속상사정보1]] : {{*사원번호, *근무부서명_#asc1, *성명_#asc2}}
    
```

가. 사원등록화면



나. 사원등록 클래스

(그림 4) 사원등록의 데이터집합 변환 중 일부

애플리케이션에서 데이터는 하나의 결과 집합으로 데이터베이스, 다른 화면 또는 창으로 전달되기도 하지만 현실에서 다양한 업무를 처리하기 위해서는 복수의 결과 집합을 주고받는 경우가 대부분이며, 이러한 다중 결과 집합은 자료구조를 활용하여 사용한다. 그림 4 는 화면설계에서 데이터의 집합형태로 전달되는 구조를 클래스에서 자료구조를 활용하여 전달하는 모습을 보여준다. 그림 4 의 화면설계에서 중괄호로 묶은 집합정보는 클래스에서 리스트와 같은 자료구조로 전달된다.

#### 4. 결론

본 연구에서는 설계된 화면설계를 개발 시 사용되는 클래스에 대한 정의하기 위해 클래스 설계기법을 제안하였다. 이 방법을 이용하여 설계하면 다수의 개발자가 공통적인 변수, 메소드 등의 사용으로 의사소통이 원활할 수 있으며 본인이 아닌 다른 개발자 및 사용자들이 어렵지 않게 이해할 수 있을 것이며, 설계 경험이 부족한 미경험자들도 효율적으로 클래스를 설계할 수 있을 것이라 기대한다.

이후에는 본 연구에서 제시하는 클래스 설계를 바탕으로 애플리케이션을 사용함에 있어 실제 사항을 고려하여 데이터베이스와 애플리케이션이 데이터를 주고받을 때 직접 주고받지 않고 뷰를 통해 데이터를 주고받아 적합성과 보안을 높일 수 있는 뷰 설계에 대한 연구가 이루어져야 할 것이다.

#### 참고문헌

- [1] Misook Choi, Jongsuk Lee, Seojeong Lee. "Combining measures of classes for efficient system design," Korean Society for Internet Information, pp85-97, 2008.
- [2] Yeongseon Kim, Useung Choe. "The Design of UML Class Diagram Create Tool using Java Code," The Korea Society of Computer and Information, pp.28-34, 2000.
- [3] Gyeongsu Joo, Dohyeong Jo. "Development of Integrated Design Methodology for Relational Database Application - Focusing on Object-Oriented Analysis and Design Methodology," The Korea Society of Computer and Information, pp25-34, 2011.
- [4] Kim Chi Su, Im Kyoung Mi, Kwon Min Ju. "Software Engineering: A Study on the Object-Oriented Methodology on the Basis of Design Patterns," Korea Information Processing Society, pp1556-1562, 1999.
- [5] Gyeongsu Joo, Jeongung Woo. "An Object-Oriented Analysis and Design Methodology for Secure Database Design -focused on Role Based Access Control," The Korea Society of Computer and Information, pp63-70, 2013.
- [6] Eunju Han, Haenggon Kim. "Meta Design Patten for Object / Class Composition," Korean Institute of Information Scientists and Engineers, pp539-542, 1997
- [7] Gayeong Park, Yonghun Lee, Sangbeom Lee. "Design and Implementation of A Conceptual Class Diagram Elicitation System," Korea Academia-Industrial cooperation Society, pp420-426, 2011.
- [8] Geun Lim. "Verification method and Simulation of Object model Converted to Formal Specification," The Korea Society of Computer and Information, pp123-130, 2007.
- [9] Geun Lim, Yeongman Kwon. "The Translation Method to formal specification of Object Model," The Korea Society of Computer and Information, pp21-27, 2003.
- [10] Jaehun Kim, Yunho Kim. "The Design of a Class Diagram Authorization Tool based on the MVC Design Pattern," The Korean Institute of Communications and Information Sciences, Vol.14 No. 12, pp2707-2715, 2010.
- [11] Jeongmo Yeo, Sangyong Park, Jaeho Myoung. "Useful Database Oracle Center in Practice," en-core, 2016.