

데이터를 활용한 화면 클래스 설계에 관한 연구

김태우*, 박선이*, 여정모**

*부경대학교 컴퓨터공학과

**부경대학교 컴퓨터공학과

e-mail: mtkim7895@pukyong.ac.kr

A Study on the Design of Screen Classes Using Data

Tae-Woo Kim, Jeong-Mo Yeo**

* Dept of Computer Engineering, Pukyong National University

**Dept of Computer Engineering, Pukyong National University

요 약

정보시스템 설계는 도출된 요구사항을 개발단계에서 수행하고자 하는 내용을 설계한다. 설계과정 중 코딩단계를 고려하여 논리적인 요소와 물리적인 요소를 연결시키는 클래스 설계와 같은 절차들에 대해서 많은 연구가 이루어졌다. 그럼에도 불구하고 경험기반의 설계요소가 존재하여 의사소통의 어려움이 발생하거나 비경험자들이 설계를 수행하기에 문제가 생길 수 있다. 이를 해결하고자 본연구에서는 업무기반의 애플리케이션 설계 절차를 바탕으로 클래스 설계 절차 중 화면의 내용을 객체지향기반으로 추상화하여 모델로 표현하며, 화면의 정보들 중 필수 요소들을 데이터로 관리하여 표현한다. 이를 이용하여 개발자 간의 의사소통의 향상과 화면의 변화에 대해 유지보수에 소요되는 시간 및 비용이 절감될 것이라 기대한다.

1. 서론

정보시스템은 기업의 복잡한 업무를 효과적으로 관리하기 위한 필수요소가 되었다. 정보시스템의 구축은 빠대가 되는 설계단계에 따라 추후 진행하는 개발 및 유지보수에 소요되는 시간과 비용을 결정하기 때문에 설계단계가 정보시스템의 성공 여부를 결정한다고 할 수 있다. 설계단계의 중요도가 증가함에 따라 개발 환경과 중요 요소들에 맞는 명확한 설계 절차에 대한 많은 연구가 이루어졌다[1].

설계 절차는 현실의 추상적인 업무를 정보시스템에 표현하기 위해 구체적인 모형을 단계별로 변환해가는 과정을 가지며 논리적인 내용뿐만 아니라 개발단계를 고려하여 설계된 내용이 개발단계에서 어떻게 변환되는지에 대한 내용도 고려되어야 한다. 다양한 설계 방법은 설계단계의 정보와 개발단계의 정보를 이어주는 맵핑설계 절차가 각각의 접근 방법에 따라 표현되고 있으며[2,3], 재사용을 통하여 개발에 소요되는 시간과 비용 절감 및 품질 향상을 목표로 하는 객체지향방법에는 클래스 설계절차를 사용하고 있다.

하지만 설계 방법들의 공통적인 문제로는 프로그램을 개발하기 위해 업무와 프로그램 업무를 통합하여 진행하게 되어 경험에 따라 다양하게 해석하여 설계되는 요소가 존재한다. 이로 인해 서로 다른 경험을 가진 설계자 및 개발자 간에 의사소통의 문제가 발생할 수 있으며, 경험이 부족한 비경험자가 설계를 수행하기에 어려움이 존재한다.

본 연구에서는 이러한 문제를 해결하기 위해 앞서 연구된 화면설계를 바탕으로 코딩단계를 소스 코

드로 변환하는 클래스 설계 방법 중 화면의 표현 방법에 대해 제안하고자 한다.

2. 관련연구

2.1. 클래스 설계

클래스 다이어그램은 시스템의 정적인 정보 구조를 나타내며 시스템에 필요한 클래스들과 이들 사이의 관계를 나타낸다. 클래스는 객체의 특성을 나타내는 여러 가지 속성들과 메소드들로 구성이 된다. 클래스 다이어그램은 객체지향 분석 방법 중 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 클래스들 간의 관계를 찾아 연결해 준다. 이후 유스케이스 시나리오를 기반으로 모든 속성과 메소드들을 도출한다[4].

클래스를 도출하기 위한 방법으로 잘 알려진 기법뿐만 아닌 다양한 방법이 존재한다. 전체적인 설계 절차 중 클래스를 도출하는 방법은 다각도로 분석하고 수행이 많이 걸리는 어려움이 존재한다. 이를 해결하기 위해 도출된 업무에서 클래스만을 도출하기 위한 방법들도 연구되어 왔다. 요구사항 정의서나 업무 기술서를 바탕으로 명사, 대명사, 명사구 등을 우선적으로 도출하고 클래스 후보 목록을 정의한다. 그

다음 설계자의 경험을 바탕으로 의미를 분석하여 중복되거나 클래스가 아닌 후보들을 제거해 클래스를 확정해 나간다. 확정된 클래스는 클래스 간의 관계를 파악하여 해당 업무의 클래스 다이어그램을 확정해 나간다[5].

2.2. 패턴지향 설계

패턴은 다양한 도메인의 정보시스템을 구축하면서 반복적으로 발생하는 문제들을 많은 사람들의 경험을 통하여 제시된 하나의 해결책을 패턴으로 정의하여 재사용성과 유연성을 향상시키는 결과를 가져왔다. 이러한 효과로 인해 기존의 설계 방법에서 발생하는 문제를 보완하는 형태로 많은 패턴들이 생겨나고 이를 체계적으로 관리 및 분석 도구에 대한 연구가 이루어졌다. 패턴 지향설계 방법은 기존의 설계 방법인 객체지향설계에 디자인 패턴 요소들을 추가시킨 방법이다. 패턴지향설계 방법의 절차는 크게 5 단계로 나뉘어 지며 요구사항 분석, 소프트웨어 디자인, 디자인 검증, 구현, 테스트 단계를 거친다. 요구사항 분석단계에서 애플리케이션의 요구 사항 중 가능한 패턴을 식별한다. 소프트웨어 디자인 단계에서는 객체와 디자인 패턴을 각각 식별하고, 클래스를 설계한 후 클래스에 디자인 패턴을 적용한다. 디자인 검증 단계에서는 요구사항에서 도출된 제약사항을 검증하고 패턴의 일관성 및 제약사항을 검증하는 두 단계를 수행한 후 구현 단계와 테스트단계를 진행한다[6,7].

2.3. 컴포넌트 기반 설계

컴포넌트 기반 설계 방법은 객체지향설계와 같이 요구사항과 기술의 변화에 품질을 유지하면서 개발 속도와 생산성을 높이기 위해 부품과 같은 작은 단위의 컴포넌트 단위를 사용하여 개발하는 방법이다. 패턴 설계와 유사하게 하나 이상의 기능을 수행하는 컴포넌트를 필요에 따라 적용하는 방식이지만 컴포넌트는 실행 가능한 하나의 모듈이란 점에서 차이가 존재한다. 컴포넌트 개발 절차는 도메인을 분석, 도메인 설계, 컴포넌트 추출, 컴포넌트 설계, 컴포넌트 구현, 컴포넌트 검증과 같은 절차를 수행한다. 컴포넌트 기반 설계는 완전한 소프트웨어 시스템을 만드는 것이 아니라 해당 도메인에 대해 재사용 가능한 부품을 만드는 것으로 재사용성을 고려하여 기능 중심의 설계이다[8].

클래스 설계, 패턴 지향 설계, CBD 설계와 같은 설계 방법들은 코딩단계에서 수행에 필요한 요소들을 고려하여 설계하는 공통점들이 존재하지만 소스 코드에 초점을 맞추어 있기 때문에 기능적인 요소들의 설계는 대부분 커버할 수 있지만 화면의 내용을 표현하는 요소는 부족함이 존재한다.

3. 화면클래스 설계

정보시스템 개발은 설계단계에 도출된 내용을 그대로 개발단계에 적용하여 만드는 것이라고 할 수 있다. 하지만 설계단계에서 도출된 내용이 개발단계에서 사용하는 틀이나 환경 및 개발자에 따라 다양하게 표현

이 가능하다. 표현의 다양성은 현실의 내용을 사실적으로 표현하는 데 도움이 되지만 통일화된 시스템 개발에는 어려움이 발생할 수 있다. 이를 해결하기 위해 많은 연구들이 이루어졌지만 경험기반의 설계요소가 존재하여 설계 및 개발에 어려움이 발생할 수 있다.

본 연구에서는 이를 해결하고자 "상품 주문시스템" 요구사항[9]을 바탕으로 분석된 요구사항에 따라 애플리케이션의 업무를 도출하는 업무 프로세스 절차와 수행되는 화면의 내용을 표현하는 화면설계 절차의 이후 절차인 클래스 설계 절차 중 일부분에 대해 다루고자 한다[10,11]. 클래스 설계 절차에서는 사용자가 직접 사용하는 화면 부분과 화면을 통해 데이터베이스와 데이터를 주고 받고 기능을 수행하는 데이터 및 기능 부분으로 나눈다. 본 연구에서는 화면설계에서 작도한 화면의 내용을 개발단계에서 수행하기 위해 데이터 관점을 활용한 화면 데이터 표현에 대해 다루고자 한다. 또한 화면 데이터의 표현은 자바 응용프로그램 기반으로 설명하고자 한다.

정보시스템의 개발은 화면에 표현되는 컴포넌트를 배치하고 컴포넌트에 표현되는 정보를 나타내기 위한 기능들을 정의한다. 화면 컴포넌트는 개발 툴에 따라 다양하게 구분을 짓고 있지만, 설계단계에서 표현한 화면의 모양은 동일하게 표현한다. 자바 또한 AWT와 SWING을 통하여 화면의 모양을 나타낸다. 동일한 툴을 사용하더라도 표현되는 화면의 모양은 다양한 방법으로 표현할 수 있다. 화면의 모양은 여러 데이터의 표현을 테이블로 표현할 수 있으며 리스트로도 표현할 수 있고, 동일한 위치라도 레이아웃에 따라 다양한 방법으로 표현이 가능하다. 같은 내용이라도 다양한 표현방법이 존재할 경우 개발이나 유지보수 중 변경이 필요하면 개발자에 따라 통일되지 못한 표현으로 소요되는 시간과 비용이 증가한다. 그러므로 화면의 내용은 변경에 유연하게 대처하도록 관리되어야 한다. 본 연구에서는 이를 위한 방법으로 표현되는 화면의 내용을 데이터로서 관리하고자 한다.

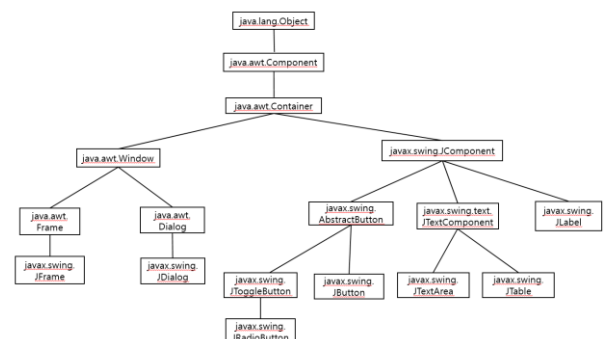


그림 1. 자바에서 제공하는 클래스들의 트리 구조 중 일부

자바에서는 표현되는 내용이 그림 1과 같이 트리 형태의 객체로서 관리되고 사용된다. 자바에서 제공하는 말단의 컴포넌트에서는 컴포넌트만의 특징을 변수로 관리하며 개발자들이 기본적으로 사용하는 배치되는 위치, 이름, 내용, 색상 등의

컴포넌트 요소는 상위의 객체에서 메소드로써 제공하므로 사용하지 않을 경우 기본값을 제공하고 있다. 그러므로 기본값이 아닌 다른 값들의 데이터를 관리하게 되면 고정된 메소드의 사용으로 개발 시 화면을 구성할 수 있다.

화면을 구성하는 요소는 화면의 틀을 구성하는 컨테이너와 실제적으로 화면의 모양을 나타내는 컴포넌트들로 구분된다. 화면을 모양을 표현하는 방법으로는 화면을 구성하는 기본 틀인 컨테이너 위에 컴포넌트들을 레이아웃을 지정하여 배치하거나 패널을 이용하여 컴포넌트들의 구역을 나누어 각각 독립적으로 배치를 달리하거나 변경을 용이하기 위해 패널 위에 컴포넌트를 배치한다. 이와 같이 컨테이너에 배치하는 컴포넌트요소와 패널에 배치하는 컴포넌트를 구분하여 그림 2 와 같이 모델로 나타낸다.

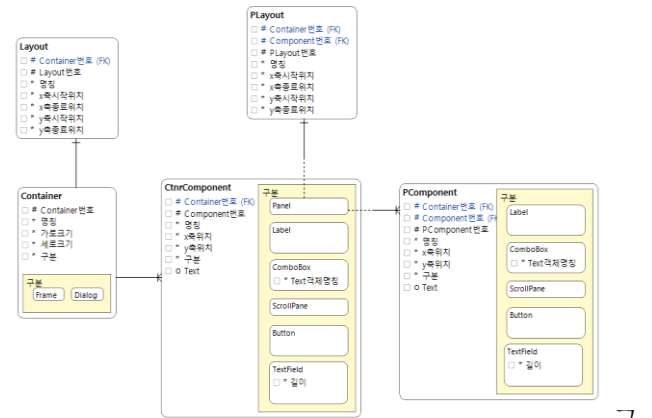


그림 2. 화면을 표현하는 객체들의 모델 구조 중 일부
표 1, 표 2, 표 3 은 "주문관리시스템"의 화면표현 데이터를 그림 2의 구조에 맞추어 관리되는 데이터의 일부를 보여준다. 표 1 은 화면을 구성하는 틀인 컨테이너에 대한 데이터 일부를 나타내며, 표 2 는 컨테이너에 배치되는 컴포넌트들의 데이터들 중 일부를 나타내며, 표 3 은 패널에 포함된 컴포넌트 정보를 보여준다. 이와 같은 데이터를 관리함으로써 개발단계에서는 메소드들을 활용하여 개발할 수 있을 것이다.

표 1. “주문관리시스템”의 화면들의 컨테이너 데이터 중 일부

Container 번호	명칭	가로크기	세로크기	구분
1	고객등록	500	500	Frame
2	사원등록	500	500	Frame

표 2. “주문관리시스템” 화면들의 컴포넌트 데이터 중 일부

CtnrComponet 번호	Component 번호	명칭	x 축위치	y 축위치	구분	Text	Text 객체명칭	길이
1	1	Panel1	0	0	Panel			
1	2	Panel2	0	300	Panel			
2	1	vName	0	0	Label	사원명		
2	2	vBasic	0	300	Label	기본주소		

표 3. “주문관리시스템” 화면들의 패널컴포넌트 데이터 중 일부

CtnrComponet 번호	Component 번호	Pcomponent 번호	명칭	x 축위치	y 축위치	구분	Text	Text 객체명칭	길이
1	1	1	vName	0	0	Label	고객명		
1	1	2	vBasicAddr	0	50	Label	기본주소		
1	2	1	vId	0	300	Label	아이디		
1	2	2	vPw	0	350	Label	비밀번호		

4. 결론

본 연구에서는 화면의 요소들이 가지는 필수적인 정보를 데이터 관점에서 모델로 관리하는 방법에 대해 정의하고 제안하였다. 이 방법을 이용하여 개발에 필요한 정보들을 명확하게 제시함으로써 프로젝트에 투입된 설계자 및 개발자 간에 원활한 의사소통이 가능할 것이라 기대하며, 개발 중이나 완성된 이후 변경이 발생할 때 정보의 변경으로 편리하게 화면의 변경이 이루어질 수 있어 유지보수에 소요되는 시간과 비용의 절감이 될 것이라 기대한다.

이후에는 시스템에 필요한 기능과 데이터베이스와 화면 사이에 주고받는 데이터들에 정의와 이들 간에 수행되는 순서들에 대한 연구가 이루어져야 할 것이다.

참고문헌

- [1]박경선, 양경식, 김현수 “성공적인 전사적 아키텍처 구축을 위한 위험요인에 관한 연구,” 한국 IT 서비스학회지, Vol.5, No.3, pp. 1-23, 2006.
- [2]김영복, 정기원 “새로운 소프트웨어 개발방법론의 통계적 평가방안,” 정보과학회지, Vol.21, No.7, pp. 1244-1251, 1994.
- [3]김치수, 임경미, 권민주 “소프트웨어공학 설계 패턴을 이용한 객체지향 방법론에 관한 연구,” 정보처리학회지, Vol.6, No.6, pp.1556-1562, 1999.
- [4]박가영, 이용훈, 이상범 “개념클래스 다이어그램 도출 시스템의 설계 및 구현,” 한국산학기술학회지, Vol.12, No.1, pp. 420-426, 2011.
- [5]최미숙, 이종석, 이서정 “효율적인 시스템 설계를 위한 클래스 간의 결합 척도,” 인터넷정보학회지, Vol.9, No.5, pp. 85-97, 2008.
- [6]김운용, 최영근 “디자인 패턴지향 소프트웨어 개발 지원 도구,” 정보과학회지, Vol.29, No.7, pp. 555-564, 2002.
- [7]김치수, 임경미, 권민주 “소프트웨어공학 : 설계 패턴을 이용한 객체지향 방법론에 관한 연구,” 정보처리학회지, Vol.6, No.6, 1999.
- [8] M.C.Jr. da Silva, P.A. de C Guerra, C.M.F. Rubira “A Java component model for evolving software systems,” *IEEE*, 2003.
- [9]여정모, 박상용, 명재호, 실무에서 유용한 데이터베이스론 오라클 중심, 엔코아컨설팅, 서울, 2016.
- [10]S. Y. Park, T. W. Kim, J. M. Yeo, “Study on the Business Process Design Method for Designing Applications,” CUTE 2017, 2017.
- [11]T. W. Kim, S. Y. Park, J. M. Yeo, “Study on the Screen Design Technique for Designing Applications,” CUTE 2017, 2017.