

정보 구조 모델링을 이용한 분석 단계에서의 객체 추출방법에 관한 연구

정정인¹⁾ 박재년

숙명여자대학교 컴퓨터학과

jiin@cs.sookmyung.ac.kr inpark@sookmyung.ac.kr

A Study on the Object Extraction Methodology using the Information Structure Modeling in the Analysis Phase

Jeong-in, Jeong¹⁾ Jae-nyun, Park

Dept. of Computer Science, Sook-Myung Women's University

요 약

기존 객체 지향 방법론에 의한 소프트웨어 개발에서의 공통적인 문제점은 표기법이 복잡하고, 산출물도 많아 사용자 이해하기가 어렵다는 것이다. 또한 분석단계에 있어서도 객체추출의 자세한 가이드라인을 제시하지 않고 있어서 개발자들의 감각과 경험에 크게 의존하고 있어 일관성이 없으며, 추출된 객체들 사이의 연결을 찾기가 쉽지 않아 확장이 어렵다. 따라서, 본 논문에서는 간단한 표기법과 산출물로 개발 초기부터 사용자의 참여를 유도할 수 있는 정보 구조 모델링을 이용하여 분석 단계에서의 객체 추출에 대한 가이드 라인을 제시하고자 한다.

1. 서론

소프트웨어의 위기와 함께 등장한 소프트웨어 공학은 구조적 방법론을 거쳐 객체 지향 방법론(object-oriented methodology)이라는 새로운 패러다임에 이르렀다. 객체 지향 방법론은 더욱더 치열한 비즈니스 환경에 따라 더욱 복잡해져만 가는 소프트웨어의 개발과 관리의 문제를 해결해 줄 대안으로 각광받으며, 기존의 절차 지향(procedure-oriented) 개발 방법론의 단점인 개발 과정의 시간낭비 또는 개발 이후 유지보수 단계의 한계에 대해 객체 지향 개발 방법론은 새로운 시도로 받아들여져 활발히 연구가 진행되어왔다[1]. Booch, Rumbaugh, Jacobson, Shlaer-Mellor 및 Yourdon 등에 의해 꾸준히 연구가 이루어졌으며, 결국 Booch, Jacobson, Rumbaugh에 의해 통일된 개발 방법론인 UML(Unified Modeling Language)[2]이 등장하였다. 그러나, 아직도 표기법(Notation)과 산출물이 복잡하여 완전한 습득과 이해를 위해선 많은 노력이 요구된다. 따라서, 효율적인 분석을 위해선 표기법이 이해하기 쉬워야 하며 산출물도 많지 않아야 한다. 뿐만 아니라, 분석의 가장 중요한 관건인 객체 식별을 위한 구체적 인 가이드라인의 제시도 필수적이다. 상세히 제시되어 있는 지침에 따라 객체를 식별한다면 좀 더 효율적인 분석이 될 것이다. 본 논문에서는 사용자가 이해하기 쉬운 개발 초기부터 참여를 유도할 수 있는 사용자 중심의 분석을 위한 방법인 정보 구조 모델링 기법[3]을 이용하여, 분석 단계에서의 객체 추출에 대한 지침을 제시하고자 한다.

본 논문의 2장에서는 관련연구로서 정보 구조 모델링 기법의 기본 개념과 Jacobson이 분류한 객체에 대해서 설명하고, 3장에서는 정보 구조 모델링에서의 객체분류 및 비즈니스 업무를 위한 정보 구조 모델링 기법을 이용한 분석 단계에서의 객체 추출 방법을 제시한다. 마지막으로, 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 정보 구조 모델링

현재까지 개발된 객체 지향 방법론의 공통적인 문제점은 표기법이 복잡하고, 분석, 설계 그리고 구현간의 연계성이 부족하다는 점이다. 따라서 사용자 요구사항의 만족도가 떨어지고 개발자의 부담이 커지게 된다. 정보 구조 모델링은 이러한 점들을 고려하여 표기법을 간단히 하고, 사용자 모델과 개발자 모델의 일치성을 보임으로써 사용자와 개발자 사이에 호환성을 제거할 수 있는 분석 모델링 기법이다.

모델링 방식은 사건중심(Event-Driven) 방식으로 사용자와 시스템의 관계는 사용자가 시스템에 요구를 하고 시스템은 그에 대한 응답을 시스템내의 정보를 이용하여 처리하는 것을 기본 개념으로 한다. 다른 개발 방법론에서는 어려운 표기법과 많은 산출물로 일반 사용자들이 쉽게 이해하기 어려운 것에 반해 정보 구조 모델링에서는 표기법이 쉽고 산출물이 간단하여 사용자들이 이해하기 쉽도록 시스템을 명세한다. 정보 구조 모델링에서 분석단계의 산출물로는 배경도, 이벤트 다이어그램, 정보구조도, 행위구조도 등이 있다.[7]. 기본객체로는 인터페이스 객체, 엔터티 객체, 컨트롤 객체가 있으며, 각 객체마다 속성, 메소드를 갖는데, 특히 메소드는 사용자 관점(User view), 관리자 관점(Manager view), 운영자 관점(Maintainer view)으로 나뉘어 기술된다.

정보 구조 모델링 방식은 자료의 종류와 관계를 사건에 따라 명시하기 때문에 사건과 사건에 대응하는 처리가 무엇인지를 추상화시키고 사건에 따라서 대응해야 하는 객체의 종류와 그들간의 관계를 제시한다. 따라서, 사용자가 이해하기 쉽고 스스로 작성 할 수 있는 모델이다[3]. 정보 구조 모델링에서의 표기는 인터페이스 객체, 엔터티 객체, 컨트롤 객체 모두 동일하며, 그림 1.과 같다.

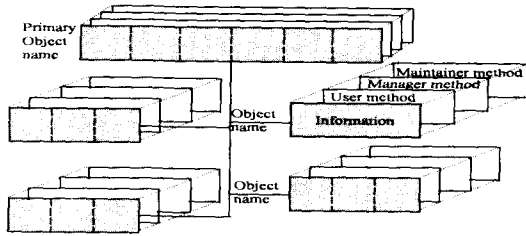


그림 1. 정보 구조 모델링 다이어그램

2.2 Jacobson의 객체분류

소프트웨어 시스템의 기본 구성 성분은 객체이다. 객체는 인간이 생각하고 표현할 수 있는 모든 것이 될 수 있으며, 객체는 서로 관련되는 정보와 행위를 묶어 놓은 일종의 패키지이다. Jacobson은 객체를 분류하기 위해 그림 2와 같이 행위, 정보, 표현의 3차원 분류 방법을 사용하여 엔터티, 인터페이스 그리고 컨트롤 객체로 분류를 한다[5].

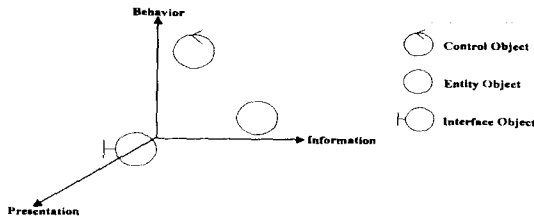


그림 2. Jacobson의 객체분류

3. 정보 구조 모델링을 이용한 객체 추출

3.1 정보 구조 모델링에서의 객체 분류

정보 구조 모델링에서도 객체는 기본 요소가 되며, 객체를 추출하는 것이 분석의 상당부분을 차지한다. Jacobson의 객체분류와 유사하게 정보 구조 모델링에서도 객체를 인터페이스 객체(Interface Object), 엔터티 객체(Entity Object), 컨트롤 객체(Control Object)로 분류한다.

- (1) 인터페이스 객체 : 시스템이 고객의 요구사항을 받아들이고 응답해주는 데 필요한 정보를 가진 객체이다. 설계단계에서 더욱 구체화되어 이는 구현단계의 화면으로 표현되며 고객이 제공하는 사건을 처리하는 단위가 된다. 인터페이스는 사용자와 시스템 사이의 매개체 역할을 하여, 입력을 받아 사용자에게 출력을 보여준다.
- (2) 엔터티 객체 : 시스템이 오랫동안 보존해야 할 정보로서 데이터 베이스에 저장되며, 시스템이 고객에게 제공해야 하는 제품이나 기술에 관한 정보이다. 엔터티 객체는 다른 객체의 기반이 되며, 산출물을 유도하는 과정에서 도구의 역할을 담당하거나, 객체간의 집합으로 이루어진 산출물이 될 수 있다. 엔터티 객체의 기본 프로토타입은 쉽게 삽입되거나 삭제되지 않으며, 엔터티 객체의 관리자만이 속성을 삽입하거나 삭제할 수 있다.
- (3) 컨트롤 객체 : 인터페이스 객체와 엔터티 객체를 자연스럽게 조절할 수 있는 객체이다. 인터페이스 객체의 요구에 대해 여러 개의 엔터티 객체에 조합으로 응답할 수 있도록 한다. 따라서 사용자의 요구사항에 적절히 대응할 수 있는 중간결과라 할 수 있다. 본 논문에서는 정보구조 모델링 기법을 이용하여 각

각의 객체들을 표현하며 이들은 설계단계에서 더욱 구체화된다.

3.2 정보 구조 모델링 기법을 이용한 객체 추출 단계

3.2.1. 배경도(context diagram)를 작성한다.

배경도는 분석 대상의 범위를 설정해주는 역할을 하며 시스템이 무엇(what)을 하는 것인가를 나타내주는 다이어그램이다. 분석하고자 하는 시스템과 외부 관련자들간의 관계, 즉 순수한 입력과 출력이 무엇인지를 나타내준다. 따라서, 배경도를 통하여 분석대상의 범위를 보여주고 시스템과 관계되는 대표적인 이벤트를 정의한다.

3.2.2 각 관련자(terminator)별로 이벤트 다이어그램(event diagram)을 작성한다.

이벤트 다이어그램은 관련자가 시스템에 행할 수 있는 이벤트를 WOD(Wanier Orr Diagram)의 형태로 표현한다[6]. WOD는 행위와 처리를 서술식으로 일목요연하게 볼 수 있는 장점이 있고, 형식은 시작(start), 처리(process), 끝(end)으로 구성된다. 배경도에서 나타난 관련자가 시스템에 행할 수 있는 이벤트를 더욱 상세하게 표현하며 사용자에게 대한 역할을 보여준다.

3.2.3. 초기 인터페이스 객체(initial Interface Object)를 추출한다.

인터페이스 객체는 시스템이 고객의 요구사항을 받아들이고 응답해주는 데 필요한 정보를 가진 객체이다. 인터페이스는 사용자와 시스템사이의 중간 매개체 역할을 하여, 입력을 받아 출력을 사용자에게 보여준다. 따라서 인터페이스 객체에서는 사용자가 인터페이스에서 발생시킬 수 있는 각각의 이벤트마다 필요한 정보들을 표현한다. 하나의 이벤트 다이어그램마다 하나의 정보구조도로서 인터페이스 객체들을 표현한다. 초기 인터페이스 객체를 추출하는 단계는 다음과 같다.

- (1) 관련자를 인터페이스 객체의 주객체로 한다. 관련자는 시스템에 정보를 제공하거나 반환 받을 수 있는 사람, 조직, 시스템이 될 수 있다. 따라서, 관련자는 인터페이스 객체를 표현하는 정보 구조도에서 주객체가 되며 속성(attribute)필드에는 신상이나 필수 정보들을 나타낸다.
- (2) 인터페이스 객체명을 정의한다. 이벤트 다이어그램에 나타난 이벤트들은 사용자(관련자)가 인터페이스를 통하여 발생시킬 수 있는 것이므로 이를 참조하여 인터페이스 객체명을 정의한다. 그리고 추출된 인터페이스 객체는 주객체에서 뺀 수직의 선에 연결하여 표현하며 이 선은 관련자와 인터페이스 객체사이의 연관관계를 나타낸다.
- (3) 각 인터페이스 객체의 속성필드에 그 이벤트를 수행하기 위한 필수 정보들을 나타낸다. 인터페이스 객체의 속성은 그에 해당하는 이벤트를 수행할 때 필요한 필수 정보가 된다. 이 단계에서는 아직 인터페이스 객체 속성의 중복이나 다른 객체와의 연결요소 등은 표시하지 않고 순수하게 필요한 정보들을 나타낸다.
- (4) 각 인터페이스 객체의 메소드를 정의한다. 인터페이스 객체마다 적절한 메소드를 정의하는데, 정보 구조 모델링에서의 메소드는 항상 3가지 관점으로 분류하여 정의된다. 사용자관점, 관리자관점, 운영자관점이 그것이다. 이때의 사용자는 시스템 사용자, 관리자는 시스템 혹은 시스템을 관리하는 직원, 운영자는 시스템을 운영, 유지보수하는 사람을 일컫는다. 정보 구조 모델링에서 메소드들을 정의하여 놓은 것을 행위 구조도라고 하며, 운영자 관점의 메소드는 설계 단계에서 하

드웨어나 그의 조건들을 고려하여 추출, 정의한다.

3.2.4. 초기 인터페이스 객체에서 엔티티 객체(Entity Object)를 추출한다.

엔티티 객체는 시스템이 오랫동안 보존해야할 정보로 시스템이 고객에게 제공해야 하는 제품이나 기술에 관한 정보이다. 따라서, 엔티티 객체의 추출을 위해서는 인터페이스 객체에서 나타나는 정보 중에서 쉽게 변하지 않고 보존되어야 할 것과 인터페이스 객체 사이에 공유되어야 할 정보들을 찾아내어야 한다. 엔티티 객체가 될 수 있는 후보는 다음과 같다.

- (1) 주객체(관린자)에 대한 정보 : 관린자는 시스템에 정보를 제공하거나 반환 받을 수 있는 사람, 조직 또는 시스템으로서 이들에 대한 정보는 시스템에 독립적이며 이들의 기본 프로토타입은 쉽게 변하지 않고, 오랫동안 보존되어야 한다.
- (2) 제품, 용역, 공용인 물건에 대한 정보 : 초기 인터페이스 객체의 정보 중에 단순히 관린자와 시스템 사이의 상호작용에서 필요한 제품이나 용역에 대한 정보, 그리고 여러 인터페이스 객체에서 공동으로 사용되는 물건은 엔티티 객체로 정의 되어 저장되고 공유되어야 한다.

이러한 지침에 따라 도서관 시스템 분석과정에서 추출된 엔티티 객체는 그림 3과 같다. 또한, 엔티티 객체 추출시에는 고려되어야 할 우선순위가 있다. 이미 기존 시스템에서 이미 저장되어 있는 자료, 엔티티 객체로의 구축이 쉬운 자료, 업무의 효율과 영업의 이익을 증대시키는 절박한 필요성에 의한 자료일 경우 우선적으로 엔티티 객체 후보가 되는 것이다. 이러한 엔티티 객체는 초기 인터페이스 객체의 이벤트가 더욱 구체화되어감에 따라 추가된다.우선순위에 따라 엔티티 객체를 추출함으로써 더욱 신뢰성있는 시스템으로의 선계가 가능해진다.

고객

고객 ID	Password	성명	주민등록번호	소속	신분	E-mail	전화번호	주소	생년월일	회원등급	회원비
-------	----------	----	--------	----	----	--------	------	----	------	------	-----

사서

사서 ID	Password	성명	주민등록번호	소속	직책	출생년도	E-mail	전화번호
-------	----------	----	--------	----	----	------	--------	------

도서

도서명	출판사명	저자명	출판사명	출판년도	ISBN	연차	종류	장수	가격	소장	자료위치	자료번호
-----	------	-----	------	------	------	----	----	----	----	----	------	------

그림 3. 도서관 시스템 분석 - 엔티티 객체

3.2.5. 초기 인터페이스 객체와 엔티티 객체를 참조하여 컨트롤 객체(Control Object)를 추출한다.

컨트롤 객체는 인터페이스 객체와 엔티티 객체를 자연스럽게 조절하는 객체로서 인터페이스 객체에게 엔티티 객체를 넘겨주는 역할을 한다. 초기 인터페이스 객체에서 나타나는 이벤트들은 가공되지 않고 단순히 입력된 정보 혹은 저장되어 있는 엔티티 객체만으로 수행될 수도 있지만, 많은 경우 엔티티 객체의 조합 또는 가공된 정보를 필요로 한다. 이때 엔티티 객체를 조합하거나 정보에 처리를 가하는 역할을 담당하는 것이 컨트롤 객체이다. 컨트롤 객체 추출의 관건은 인터페이스 객체의 메소드로 정의하느냐, 컨트롤 객체로 정의하느냐이다. 이때의 기준은 첫째, 인터페이스 객체에서 입력된 자료와 하나의 엔티티 객체만의 조합으로 필요한 정보를 얻을 수 있다면 메소드로 정의한다. 둘째, 똑같은 행위가 계속 반복이 된다면 이는 컨트롤 객체로 정의하여 효율성을 높인다.

3.2.6. 최종 인터페이스 객체를 추출한다.

초기 인터페이스 객체를 통해 엔티티 객체와 컨트롤 객체를 추출한

후 초기 인터페이스 객체를 정제하는 단계를 거친다. 이 단계에서는 초기 인터페이스 객체의 정보중에서 엔티티 객체와 컨트롤 객체를 참조하는 것은 밀출로써 연결요소임을 표현한다. 이러한 작업의 반복을 통해 서로 다른 인터페이스 객체의 연결을 나타내고 인터페이스 객체의 엔티티 객체와 컨트롤 객체와의 연결도 나타내게 된다. 이때 유의할 점은 인터페이스 객체가 엔티티 객체나 컨트롤 객체를 참조할 때 객체내에서 정보의 중복이 있어서는 안된다는 것이다. 이러한 작업은 시스템이 복잡해지고 자료가 방대해지면 인간의 능력으로는 완벽하게 수행하기가 불가능하며, 자동화를 통해 중복여부를 찾는 것이 바람직하다.

모든 분석이 끝난후 사용자의 요구사항이 추가될 경우에는 3.2.2단계부터 시작하여 추가된 요구사항만을 고려 하여 위의 단계들을 수행하면 된다. 연결요소와 메소드를 함께 표현한 최종 인터페이스 객체는 그림 4.과 같다.

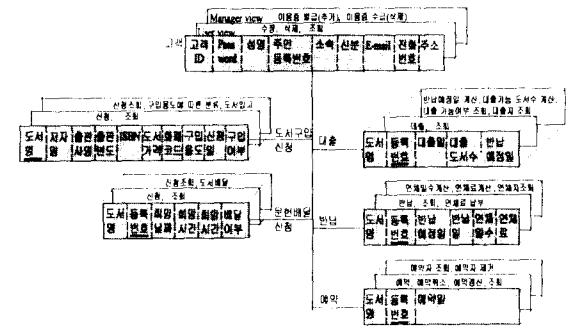


그림 4. 도서관 시스템 분석 - 고객 인터페이스 객체

4. 결론 및 향후 연구 과제

본 논문에서는 정보구조 모델링 기법을 이용한 분석 단계에서의 객체추출 방법을 제안하였다. 정보 구조 모델링은 간단한 표기법으로서 사용자의 요구사항을 파악하기 쉽고, 이벤트 별로 필수 정보를 한눈에 볼 수 있으며, 객체의 연결을 쉽게 추적 가능하여 확장이 용이하다. 따라서, 위와같은 지침에 따라 객체를 추출하면 좀 더 일관성있고 효율적인 분석이 가능하다. 현재는 인터페이스 객체의 입력으로 객체간의 중복발견 및 저장되어 있는 지침에 따라 자동으로 엔티티 객체를 추출하도록 하는 도구의 설계에 대한 연구가 진행중에 있다.

5. 참고문헌

- [1] Donald G.Firesmith, Object Oriented requirements analysis and design, John Wiley & Sons, 1993
- [2] Martin Fowler, "UML Distilled Applying the standard object modelin language", Addison-Wesley Longman, Inc. , 1997
- [3] 이서정, 박재년, "사용자의 참여를 확장한 시스템 개발 방법의 설계", 정보처리학회 논문지, 제6권, 제5호, pp.1272-1279, 1999년 5월.
- [4] Ivar Jacobson, The object Advantage : Business process reengineering with object technology, Addison-Wesley, 1994
- [5] Ivar Jacobson, "Object-Oriented Software Engineering", Addison-Wesley, 1994
- [6] 박재년, "기초전산학", 교학사, 서울, pp337-366, 1997년
- [7] 박재년, "정보 구조 모델링에 의한 시스템 분석", 숙명여자대학교 논문집, 제33집, 1992년 12월