

사용자 이벤트 기반의 정보구조 모델링을 이용한 비즈니스 업무 분석에서의 클래스 추출 방법

이 혜 선[†] · 박 재 년^{**}

요 약

비즈니스 시스템 분석을 위한 유스케이스 모델링은 유스케이스의 기능 별 분해 수준 및 시나리오의 구체적인 기술 단계를 어느 수준까지 작성해야 할 지 결정하기 어렵고 사용자가 분석 결과를 쉽게 이해하기가 어렵다. 본 논문에서는 비즈니스 시스템 분석 단계에서 사용자와 개발자 모두 쉽게 이해할 수 있는 표기법을 사용하여 사용자의 요구사항을 직관적으로 표현할 수 있는 정보구조 모델링을 정의하고, 객체 타입 별로 이벤트 객체, 재산 객체, 거래 객체로 분류 정의하여 추출하는 방법에 대해 제안함으로써 사용자의 요구사항 변경에 쉽게 대처하고, 개발자들이 분석 결과를 기반으로 설계를 위한 클래스 도출을 보다 용이하게 할 수 있도록 한다.

키워드 : 클래스 추출, 요구사항 분석, 비즈니스모델링, 유스케이스 모델링

User Event-based Information Structure Modeling for Class Abstraction of Business System

Hye-Seon Lee[†] · Jai-Nyun Park^{**}

ABSTRACT

Use case modeling is a widely used technique for functional requirements analysis of business system, but it is difficult to identify a use cases at the right level and use case specifications are too long and confusing. It is also hard to determine a functional decomposition phases of use cases. Therefore customer doesn't understand the use cases. This paper is defining concept of the Information Structure Modeling(ISM) and analyzing business system for the customer's perspective. ISM is an efficient mechanism for analyzing user requirements and for identifying objects in a business system using Attribute Structure Diagram, which is a major tool of the ISM that describes user event. This paper is also to show how the classes are classified and derived as event-asset-transaction type in ISM. It provides a user-friendly approach to visually representing business model.

Key Words : Class Derivation, Requirements Analysis, Business Modeling, Use Case Modeling

1. 서 론

비즈니스 시스템 분석은 관심의 대상인 현실세계, 즉 비즈니스의 객체와 관계를 추상화하여 비즈니스 모델을 만드는 작업이라 할 수 있다. 또한 새로운 시스템이 무엇을 할지를 미리 정의하는 것이며, 개발자의 고객인 사용자의 요구사항을 정의하는 것이라 할 수 있다.

현재 비즈니스 시스템 분석을 위해 UML의 유스케이스 모델링을 많이 사용하고 있다. 그러나 유스케이스 모델링은 유스케이스의 내용과 범위를 명확히 설정하기가 어렵다. 즉, 하나의 유스케이스를 기술하는 추상화 레벨이 분석가 별로

다르게 기술하고 있으며, 유스케이스 모델링을 작성하기 위한 명확한 지침이 부족하여 경험자에 크게 의존하고 있는 실정이다. 또한 고객은 유스케이스 모델링의 결과를 잘 이해하지 못하여 잦은 요구사항 변경의 원인이 되고 있기 때문에 개발 초기의 분석단계에서 요구사항을 정확히 추출하기는 어려운 실정이다[1-3].

따라서 본 논문에서는 고객의 이벤트를 기반으로 비즈니스 시스템을 분석하여 사용자와 개발자 모두 쉽게 이해할 수 있는 표기법을 사용하여 고객의 요구사항을 직관적으로 표현할 수 있는 정보구조 모델링에 대해 정의하고, 객체 타입 별로 객체를 정의하여 추출하는 방법에 대해 제안함으로써 고객의 요구사항 변경에 쉽게 대처할 수 있고 개발자들이 분석 결과를 기반으로 설계를 위한 클래스 도출을 보다 용이하게 할 수 있도록 한다.

※ 본 연구는 숙명여자대학교 2003년도 교내 연구비 지원에 의해 수행되었음.

† 준 회 원 : 숙명여자대학교 대학원 컴퓨터과학과 박사과정

** 정 회 원 : 숙명여자대학교 정보과학부 교수

논문접수 : 2005년 7월 6일, 심사완료 : 2005년 11월 25일

2장에서는 관련 연구로 유스케이스 모델링에서의 클래스 도출 과정과 어려운 점을 제시하고, 3장에서는 이벤트 기반의 정보구조 모델링에 대해 정의한다. 4장에서는 정보구조 모델링에서의 객체 타입 별 추출 방법에 대해 제안하고, 5장에서는 정보구조 모델링으로 분석 클래스를 도출하는 사례를 보이고 유스케이스와 비교하여 본다. 끝으로 6장에서는 본 연구의 결론을 기술한다.

2. 관련 연구

2.1 유스케이스 모델링을 이용한 클래스 도출 과정

유스케이스 모델링은 시스템의 외부 관점을 나타내는 모델로 액터의 요구사항에 대한 결과를 생산해주는 비즈니스 활동의 일련 과정이다. 유스케이스 모델링을 이용하여 클래스를 도출하기 위한 과정은 (그림 1)과 같다.

먼저 비즈니스 요구사항으로부터 시스템의 경계를 정의하고 액터와 유스케이스를 찾아 시스템의 기능별로 유스케이스 다이어그램을 작성하고, 각 유스케이스 별로 시나리오를 기술한다. 시나리오는 액터의 요구마다 시스템이 응답하기 위해 필요한 자료와 행위를 기술하는 것으로 처음 반복에서는 성공을 전제로 하는 성공흐름(Success Flow)으로 대화 위주의 단순 문장으로 작성하고, 반복하면서 확장흐름(Extension/Alternate Flow)과 예외흐름(Exception Flow)을 추가하여 작성한다. 유스케이스 별로 시나리오가 기술되었으면 각 시나리오를 분석하여 유스케이스를 객체로 매핑하는 유스케이스 실체화 과정을 거쳐 객체들의 책임과 역할을 구체화하여 객체를 추출한다[4-7].

유스케이스 실체화(Use Case Realization)는 유스케이스에서 시스템이 갖추어야 할 객체와 관계를 찾아내는 분석 작업으로 Jacobson이 분류한 바운드리 객체, 컨트롤 객체, 엔티티 객체의 3가지 타입으로 객체를 식별하여 응답에 필요한 작업 순서대로 객체를 나열하여 객체간의 상호작용을 표현하는 콜레보레이션 다이어그램을 작성한다. 또는 추출

된 객체간의 상호작용과 메시지를 시간적인 순서로 표현하는 시퀀스 다이어그램의 작성을 통하여 객체를 정제하면서 필요한 새로운 객체를 추출한다. 실체화의 결과로 추출된 객체들을 분석하여 클래스로 정의하고, 클래스들간의 관계를 분석하여 클래스 다이어그램을 작성한다[6-8].

2.2 유스케이스 모델링의 문제점 분석

UML 기반의 유스케이스 모델링에서 유스케이스는 기능적으로 분해된 명세서로부터 클래스를 설계하기 위한 객체 명세서로의 변환이 어렵고, 클래스에 대해 언급하지 않기 때문에 개발자들이 각자의 방법으로 객체를 분석하므로 재사용하기가 어렵다[9-11]. 또한 유스케이스 정의와 클래스 도출 및 책임 분담의 방법에 대한 지침이 따로 제시되지 않고 있기 때문에 분석가의 경험과 지식에 의해서 추출되고 있다. 그래서 분석모델은 분석가의 실무 경험과 관점에 따라 내용은 같아도 모델의 표현이 다양하게 만들어지는 것이 문제점으로 지적되었다[2].

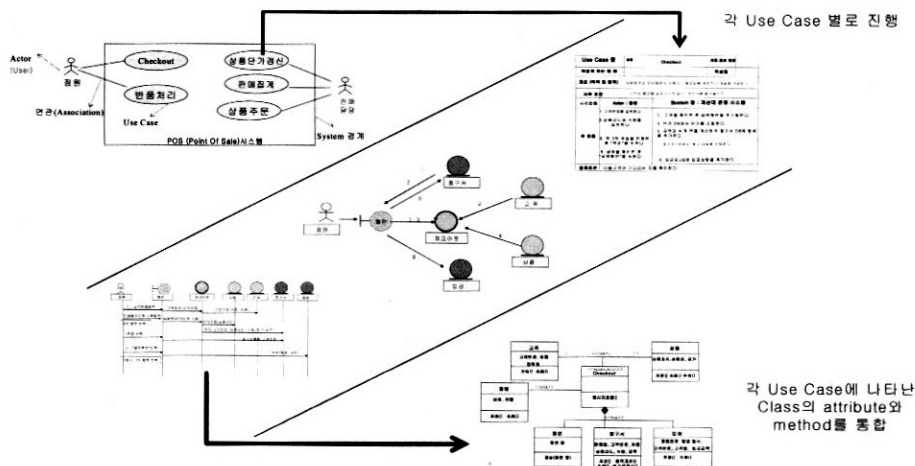
또한 유스케이스 모델링은 경험자가 많지 않기 때문에 익숙해지기 위해서는 많은 공부와 경험이 필요하다. 전문 분석가에게도 유스케이스 모델링에 익숙해지는 것이 어려운 일인데 더군다나 고객이 UML 모델링으로 표현된 모델을 이해하기란 매우 어려운 것이다[1].

따라서 본 논문에서는 개발 전문가인 시스템 분석가와 UML 표기에 익숙하지 않은 비즈니스 업무담당자가 공통의 목적과 관점으로 시스템을 분석하여 업무담당자가 특별한 경험이 없어도 이해할 수 있는 비즈니스 시스템 분석 모델인 정보구조 모델링을 정의하고, 정보구조 모델링에 의한 클래스 도출 방법에 대해 제안한다.

3. 고객 이벤트 기반의 정보구조 모델링

3.1 정보구조 모델링의 정의

정보구조 모델링(Information Structure Modeling)은 이벤트/응답 중심(Event/Response-Driven)의 방식으로 비즈니스



(그림 1) 유스케이스 모델링에서의 클래스 도출 과정

고객이 기업에 서비스를 요구하는 행위인 이벤트를 일으키고, 기업의 직원은 업무규칙과 업무지침서에 의해서 이벤트에 응답해주는 내부활동과 이 활동의 대상이 되는 기업의 재산을 기반으로 분석하는 시스템 분석 모델이다[12].

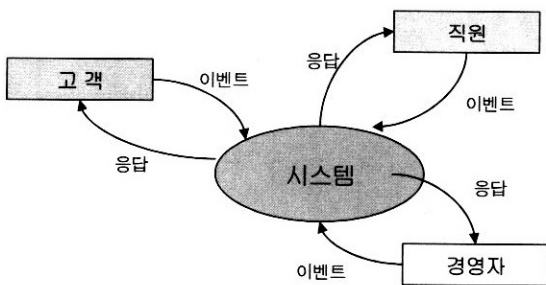
정보구조 모델링에서의 비즈니스 시스템 분석 시, 비즈니스는 직원들의 업무에서 사용되는 일상적인 용어 속에 객체의 속성과 관계가 포함된 경우가 많기 때문에 비즈니스 자체에서 이미 클래스나 관계를 설정해 놓고 업무를 본다 할 수 있다. 따라서 비즈니스 시스템에서 사용하는 용어나 관계를 그대로 표현할 수 있고, 업무에 해당하는 행위가 일어났을 때 그 행위에 필요한 자료 묶음들의 구조를 정리하여 속성구조 다이어그램과 행위구조 다이어그램으로 표현한 것이 정보구조 모델링이다.

다음은 비즈니스 시스템 분석을 위한 정보구조 모델링의 기본 모델링 도구의 개념에 대해 살펴본다.

3.2 정보구조 모델링 도구

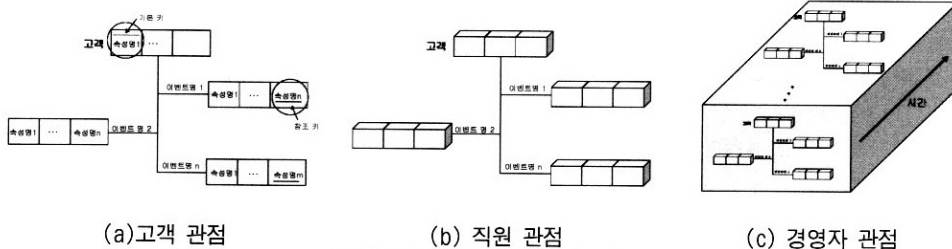
(1) 컨텍스트 다이어그램(Context Diagram)

컨텍스트 다이어그램은 비즈니스가 서비스할 고객을 정의하고 개발할 시스템의 범위를 결정하기 위한 것으로 (그림 2)와 같은 형태로 작성한다.



(그림 2) 컨텍스트 다이어그램

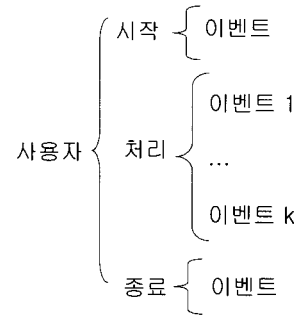
(그림 2)에서 사용자는 시스템으로부터 응답을 받는 주 사용자(Primary Actor)인 고객과 고객의 정보들을 관리하거나 고객들의 시스템 이용 현황에 대해 분석하고 통계 내는 역할을 하는 직원, 그리고 직원의 보고를 받아 시스템에 대한 전반적인 추세나 상황 등을 파악하여 의사결정을 할 수 있는 경영자로 구분할 수 있다. 컨텍스트 다이어그램은 각 사용자의 주요 업무에 대한 대표적인 이벤트와 응답만을 표현하여 작성한다.



(그림 4) 속성구조 다이어그램

(2) 이벤트 다이어그램(Event Diagram)

이벤트 다이어그램은 사용자의 요구사항인 이벤트를 (그림 3)과 같은 형태로 표현한 것으로, 사용자가 시스템을 이용하기 위해 등록하는 시작(Start), 시스템을 사용하기 위해 요구하는 이벤트 목록을 나열한 처리(Process), 그리고 더 이상 시스템을 사용하지 않고 시스템을 떠나기 위한 조건인 종료(End)로 구성된다.



(그림 3) 이벤트 다이어그램

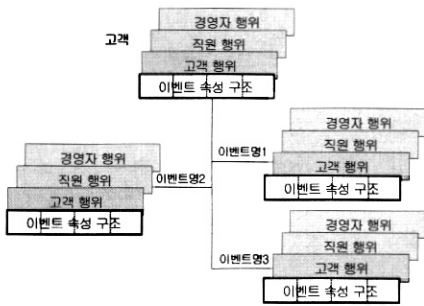
이벤트의 형식은 <주어+간접목적어+직접목적어+요구타동사>의 형태로 표현하며, 간접목적어는 기본적(default)으로 시스템을 의미하므로 일반적으로 생략하여 표현한다.

각 사용자 관점 별로 시스템에 서비스를 요구하는 이벤트는 시스템의 상태를 바꿀 수 있는 이벤트만을 분석 대상으로 하며, 사용자의 요구사항이 증가할 때마다 이벤트 목록은 증가된다.

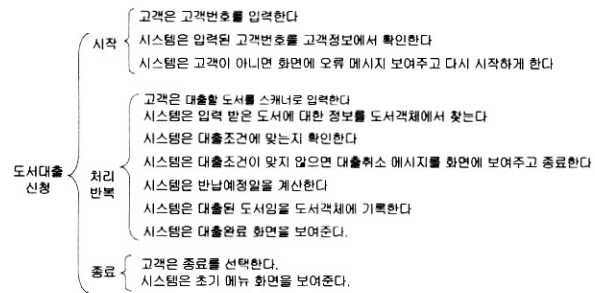
(3) 속성구조 다이어그램(Attribute Structure Diagram)

속성구조 다이어그램은 각 사용자 관점 별로 사용자의 기본 속성 정보와 이벤트 관련 속성정보를 표현한 것으로, 이 중에서 가장 핵심이 되는 것은 고객 관점의 속성구조 다이어그램이며 이는 고객의 이벤트를 중심으로 분석을 시작하기 때문이다.

(그림 4)의 (a)는 고객 관점의 속성구조 다이어그램으로 고객이 시스템에 어떤 요구(이벤트)를 하면 그 요구를 만족시키기 위해서 어떤 정보들이 필요한지를 알 수 있다. 그러기 위해서는 먼저 고객이 가지는 고유한 정보를 기본자료로 식별하여 주 객체로 설정한 다음, 시스템에서 응답을 받기 위해 필요한 이벤트 관련 속성 정보들을 식별하여 종속 객체로 설



(a) 행위구조 다이어그램



(b) 행위 명세서

(그림 5) 행위구조 다이어그램 및 행위 명세서

정한다. 주 객체의 속성 명 위에 그려진 윗줄은 기본 키를 의미하며 각 이벤트에 대한 종속 객체의 속성 명 하단에 그려진 밑줄은 다른 객체를 참조하기 위한 참조 키를 의미한다.

(그림 4)의 (b)는 직원 관점의 속성구조로 직원은 모든 고객들의 이벤트 관련 정보를 관리할 수 있기 때문에 고객 개인의 정보가 축적된 자료형태인 3차원 형식의 입체적인 구조로 표현하며, (그림 4)의 (c)는 경영자 관점의 속성구조로 시간(Time)의 개념이 추가된 4차원 형태로 표현한다.

(4) 행위구조 다이어그램 및 행위 명세서

행위구조 다이어그램(Behavior Structure Diagram)은 (그림 5)의 (a)와 같이 고객의 이벤트인 속성구조 다이어그램과 관련된 각 사용자 관점에서의 다양한 행위를 메소드 명으로 표현한 것이다. 고객 관점에서의 행위는 고객이 받을 수 있는 서비스를 의미하며, 직원과 경영자 관점에서의 행위는 업무를 수행하기 위하여 관련 데이터를 가공하기 위한 메소드들의 집합을 표현한 것이다.

행위 명세서(Behavior Description)는 (그림 5)의 (b)와 같이 각 행위에 대한 구체적인 프로세스로 고객의 이벤트에 대한 시스템의 처리과정을 단순문장의 집합으로 기술한 것이며, 구현 시 개발자를 위한 자료가 된다.

4. 정보구조 모델링에서의 객체 타입 별 분류 및 추출 방법

정보구조 모델링에서의 클래스 도출 과정은 먼저 컨텍스트 다이어그램에서 주요 사용자인 액터를 추출하고, 각 액터 별 이벤트에 대한 속성구조 다이어그램과 행위구조 다이어그램을 작성하고 정련하면서 객체 타입 별로 객체를 추출하여 클래스로 정의한다.

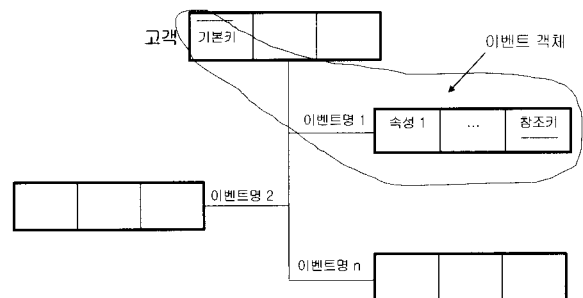
정보구조 모델링에서의 객체 타입은 이벤트 객체, 재산 객체, 거래 객체로 분류 정의한다. Jacobson이 분류한 객체와 비교해보면 이벤트 객체는 바운드리 객체와 유사하며, 재산 객체는 엔티티 객체에, 거래 객체는 컨트롤 객체에 각각 대응된다.

4.1 이벤트 객체(Event Object)

이벤트 객체는 시스템 사용자의 사용 목적에 따라 고객

관점의 이벤트와 직원 또는 경영자 관점의 이벤트인 업무로 구분하여 작성할 수 있으며, 다음과 같이 정의한다.

- ① 시스템 입장에서 어떤 서비스를 제공해야 할 지를 알 수 있도록 외부에서 받아들이는 인터페이스 정보
- ② 고객 입장에서 시스템에 서비스를 요구하는 이벤트와 관련된 정보로, 누가(고객이) 무엇을(이벤트를) 어떤 조건으로(속성정보로) 시스템에 요구하는 지에 대한 정보



(그림 6) 이벤트 객체

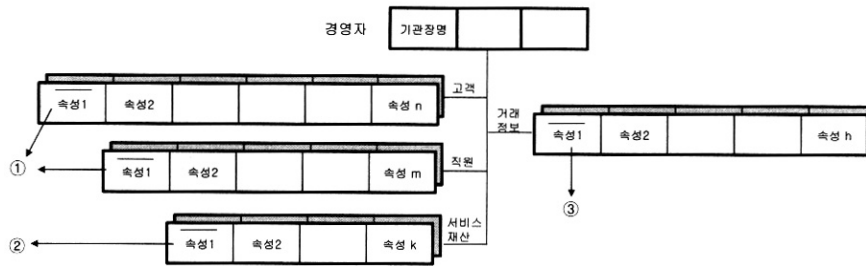
이벤트 객체는 (그림 6)과 같이 고객의 속성구조 다이어그램에서 주 객체인 고객의 기본 키와 이벤트 명 그리고 이벤트 관련 속성정보와 기본적인 CRUD(Create, Read, Update, Delete) 메소드를 통합하여 추출한다.

이벤트 객체는 고객이 비즈니스에 거래를 요구하는 행위를 알려주기 위한 객체로서 기본적으로 고객이 입력해야 하는 정보를 나타낸 컴퓨터 화면으로 이해하면 된다. 이벤트 명은 시스템이 갖고 있는 서비스를 고객이 요구하는 유스케이스 명으로 볼 수 있다.

4.2 재산 객체(Asset Object)

재산 객체는 고객의 이벤트에 응답하기 위해 기본적으로 소유할 필요가 있는 경영자의 재산이 되는 객체로, 다음의 경우에 해당하는 객체를 재산 객체로 정의하여 (그림 7)과 같은 속성구조 다이어그램으로 작성한다.

- ① 고객과 고객에게 제공하는 제품이나 상품, 기술 그리



(그림 7) 자산 객체

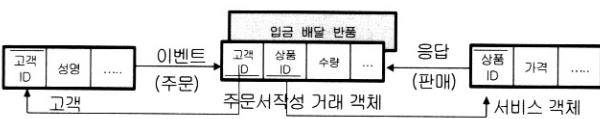
- 고 운영인력 등 경영자의 비즈니스 운영 자산
- ② 고객이 요구하는 서비스와 관련된 자산
- ③ 거래가 성립되었을 때, 거래와 관련된 객체는 회사의 재산인 비즈니스 활동의 수입이므로 이를 자산 객체로 정의

추출된 자산 객체의 메소드는 각 사용자의 관점에 따라 사용 용도가 다를 수 있으므로 처음에는 기본적인 CRUD만을 포함하며, 설계단계에서 구체적으로 추가된다.

4.3 거래 객체(Transaction Object)

거래 객체는 비즈니스에서 제공하는 여러 종류의 재산을 이용하여 실제적으로 거래를 처리하는 비즈니스 논리로서 다음과 같이 정의한다.

- ① 고객의 요구사항인 이벤트를 처리하기 위하여 2개 이상의 자산 객체를 필요로 하는 경우
- ② 고객의 이벤트 객체에 응답하기 위해 필요한 시스템의 자산 객체를 이용하기 위하여 각종 서비스 규칙을 적용하여 자산 객체를 연결해 주는 비즈니스 프로세스를 정의한 객체



(그림 8) 거래 객체

(그림 8)은 거래 객체의 추출 예를 보여준 것으로, 고객이 주문 이벤트를 요청하면 회사의 재산이 되는 모든 서비스 자산 객체들을 이용하여 주문 이벤트에 응답하기 위해 필요한 '주문서작성' 정보를 만들기 위한 비즈니스 프로세스를 정의한 것이 거래 객체라는 것을 표현한 것이다. '주문서작성' 거래 객체의 결과는 자산 객체가 될 수 있으며, 이 자산 객체를 이용하기 위한 '입금', '배달', '반품' 등의 비즈니스 프로세스들이 있으며, 이들은 모두 '주문서작성' 거래 객체의 메소드로 정의될 수 있다. 메소드는 비즈니스 규칙에 따라 계속해서 확장 개발된다.

이상과 같이 정보구조 모델링에서는 고객의 이벤트를 시작으로 속성구조 다이어그램을 작성하여 이벤트 객체를 추

출하고, 각 이벤트 객체에 응답하기 위해 필요한 시스템의 자산 객체를 정의하고, 이벤트 객체와 자산 객체간의 상호 작용인 비즈니스 프로세스를 정의하는 거래 객체를 추출하고, 추출된 거래 객체에 대한 사용자 관점 별 행위를 표현한 행위구조 다이어그램의 작성을 통하여 객체를 구체화한다. 추출된 객체는 정적인 형태의 클래스로 정의됨으로써 정보구조 모델링을 이용한 비즈니스 시스템의 분석 클래스가 도출된다.

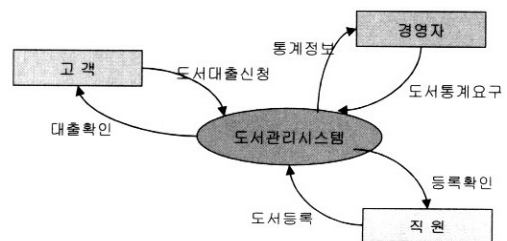
다음은 정보구조 모델링을 적용한 도서관리시스템 분석과정에 대해 살펴본다.

5. 정보구조 모델링을 이용한 도서관리시스템 분석 사례

5.1 정보구조 모델링에서의 클래스 도출 과정

(1) 컨텍스트 다이어그램 작성

도서관리시스템의 도메인을 분석하여 사용자의 주 업무에 대한 대표적인 이벤트와 응답을 표현한 컨텍스트 다이어그램은 (그림 9)와 같다.

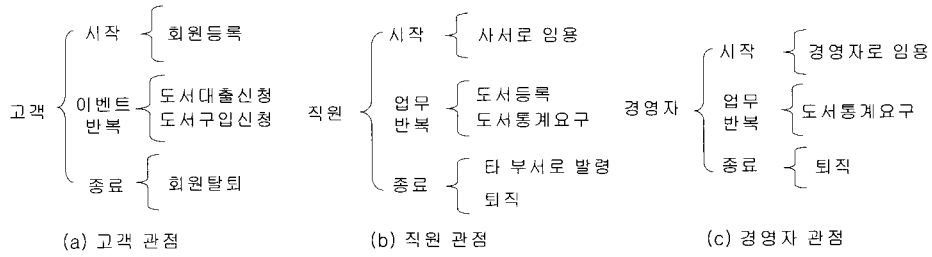


(그림 9) 컨텍스트 다이어그램

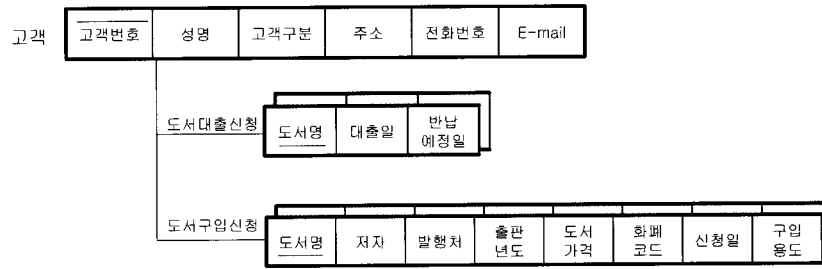
(그림 9)에서 고객의 주 이벤트는 도서대출신청이며, 직원의 이벤트인 도서등록은 고객이 도서대출신청을 요구하기 전에 시스템에서 미리 제공해야 하는 업무이며, 경영자의 도서통계요구는 시스템의 상태를 바꾸는 이벤트가 아닌 단순히 통계현황 조회를 요구하는 업무로 볼 수 있기 때문에 표현은 문맥상 이벤트로 그렸지만 실제적으로는 직원과 경영자의 업무로 보아야 한다.

(2) 이벤트 다이어그램 작성

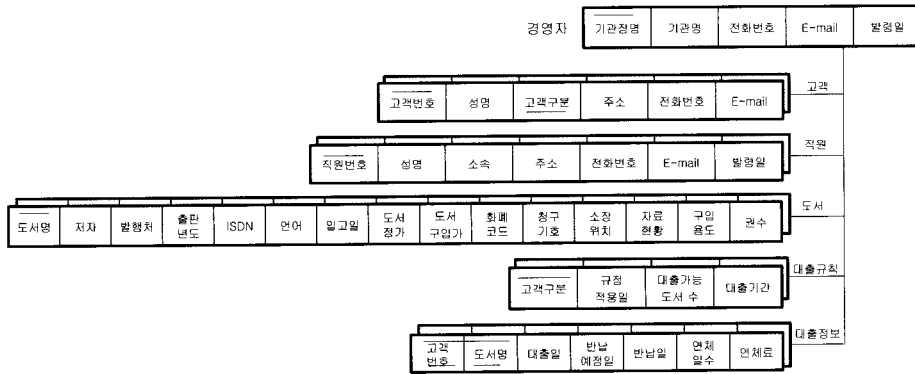
이벤트 다이어그램은 도서관 기구표와 업무분장 그리고



(그림 10) 이벤트 다이어그램



(그림 11) 고객의 속성구조 다이어그램



(그림 12) 재산 객체 속성구조 다이어그램

대출관련 문서 및 기존 시스템의 컴퓨터 화면을 조사하여 담당 직원들과 함께 각 사용자 별로 도서관리시스템에 요구하는 주요 이벤트 목록을 정의하여 (그림 10)과 같이 작성한다.

(그림 10)에서 고객은 외부 이벤트이지만 직원과 경영자는 비즈니스 내부에서 업무를 처리하기 위해 정보화된 시스템에 일으키는 내부 이벤트로 직원의 업무로 볼 수 있기 때문에 “업무반복”으로 선정하여 기록하였다.

(3) 이벤트 객체의 속성구조 다이어그램 작성

주 객체인 고객의 이벤트와 관련된 속성구조 다이어그램을 작성하면 (그림 11)과 같으며, 본 사례에서는 도서관리시스템의 주요 기능인 고객의 도서대출신청 이벤트 만을 대상으로 분석한다.

(그림 11)에서 주 객체인 고객의 기본 키는 고객번호이며, 고객이 일으키는 이벤트 객체의 참조 키는 도서명이다. 따라서 도서라는 재산 객체와 연결된다는 것을 알 수 있다. 또한 주 객체에서 상속 받은 고객번호와 도서대출신청 이벤트 관련 속성정보 그리고 기본적인 CRUD 메소드로 구성된

도서대출신청 이벤트 객체를 추출할 수 있다. 마찬가지로 도서구입신청 이벤트 객체도 추출된다.

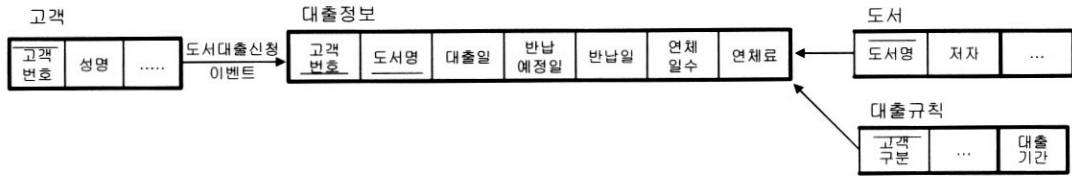
(4) 재산 객체의 속성구조 다이어그램 작성

고객의 도서대출신청 이벤트 객체의 참조 키인 도서명에 의해서 도서는 고객이 대출하겠다고 요구하는 시스템이 가진 상품이므로 시스템의 재산이라는 것을 알 수 있다. 따라서 도서관리시스템에서의 비즈니스 재산을 속성구조 다이어그램으로 표현하면 (그림 12)와 같다.

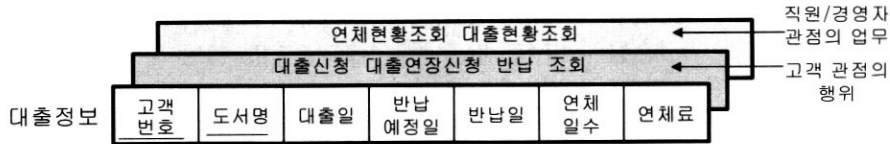
(그림 12)에서 고객, 직원, 도서는 비즈니스 운영 재산이며, 대출규칙은 고객이 요구하는 서비스와 관련된 재산 객체이다. 또한 대출정보는 도서대출신청 이벤트에 의한 비즈니스 프로세스의 수행 결과를 나타낸 대출정보 거래 객체를 비즈니스 재산의 재산 객체로 정의한 것이다.

(5) 거래 객체의 속성구조 다이어그램 작성

거래 객체는 고객의 이벤트에서 발생되어 도서관리시스템에서 승인한 이벤트 객체로서 (그림 13)과 같은 속성구조로



(그림 13) 거래 객체



(그림 14) 대출정보 거래 객체의 관점 별 행위구조 다이어그램

<표 1> 정보구조 모델링과 유스케이스 모델링 비교

| 정보구조 모델링 | 유스케이스 모델링 |
|--|---|
| 고객의 요구인 이벤트를 시작으로 분석 | 시스템의 기능별로 분해된 유스케이스 추출에서부터 분석 |
| 이벤트 관련 데이터 중심으로 분석 | 유스케이스 기능 중심으로 분석 |
| 비즈니스에서 일이 진행되는 순서로 분석하므로 사용자와 개발자 모두 이해하기 쉬움 | 전문가적인 시각에서 분석되므로 사용자가 이해하기 어려움 |
| 속성구조와 행위구조 작성을 통해 직관적으로 객체 추출 | 유스케이스 별 시나리오 작성에서부터 시퀀스 다이어그램 작성까지의 일련의 과정을 거쳐서 객체 추출 |

표현된다.

(그림 13)에서 도서대출신청 이벤트의 최종 결과는 고객과 도서, 대출규칙 간의 상호작용으로 대출정보라는 거래 객체의 속성으로 표현되며, 거래 객체의 속성은 이벤트 객체의 속성에 응답에 필요한 비즈니스 프로세스를 처리하기 위해 필요한 속성이 추가되어 정의된 것이다. 거래 객체의 메소드는 각 사용자 관점에 따른 행위구조 다이어그램의 작성을 통하여 다음과 같이 정의된다.

(6) 거래 객체의 행위구조 다이어그램 작성

대출정보 거래 객체에 대한 사용자 관점 별 행위를 정의한 행위구조 다이어그램은 (그림 14)와 같다.

고객 관점의 행위는 도서대출화면을 통하여 대출을 신청하고, 대출정보를 조회하고, 대출연장신청을 하고, 대출도서 반납을 수행하는 행위가 있다. 또한 직원과 경영자 관점에서의 행위는 모든 고객의 대출정보현황에 대한 조회와 연체현황을 조회하는 일 등의 각종 통계표를 출력하는 업무를 행위 명으로 해서 행위구조 다이어그램을 작성한다.

사용자 관점 별로 작성된 행위 명은 거래 객체의 메소드로 정의되며, 각 행위에 대한 구체적인 프로세스는 행위 명세서로 표현될 수 있다. 또한 거래 객체의 행위 명 하나하나 유스케이스로 볼 수 있다.

5.2 유스케이스 모델링과의 비교 분석

도서관리시스템을 유스케이스 모델링으로 학생들의 실습에 적용했던 결과는 처음부터 유스케이스가 액터의 관점 별

지침이 없이 시작되어서 유스케이스 도출 자체가 난해하고 또 시나리오 작성에서 클래스를 도출하는데 시나리오의 범위와 관점 그리고 표현의 추상화 수준이 작성자마다 달라서 클래스의 정의뿐만 아니라 그 속성을 추출하기도 어렵고 클래스의 수도 많아져서 관리도 복잡하였다.

반면에 정보구조 모델링은 고객이 요구하는 이벤트를 중심으로 분석하기 때문에 고객의 요구에 응답하기 위해 필요한 객체의 속성을 쉽게 정의할 수 있으며, 시스템에 필요한 기능을 속성구조 다이어그램 작성을 통하여 한 눈에 쉽게 파악할 수 있기 때문에 복잡하게 시나리오를 작성하여 분석하지 않더라도 필요한 객체를 쉽게 추출할 수 있었다.

<표 1>은 정보구조 모델링과 유스케이스 모델링을 비교한 것이다.

6. 결론 및 향후 연구과제

본 논문에서는 고객의 이벤트를 기반으로 비즈니스 시스템을 분석하는 정보구조 모델링에 대해 정의하고, 정보구조 모델링을 이용하여 주 객체인 고객의 이벤트 관련 속성정보와 비즈니스의 소유제산을 표현한 속성정보, 그리고 고객의 이벤트에 응답하기 위해 필요한 비즈니스 프로세스를 정의한 거래 객체의 속성구조와 사용자 관점 별로 거래 객체의 행위를 정의한 행위구조 다이어그램을 작성하면서 객체의 유형 별로 객체를 추출하여 클래스를 도출하는 과정에 대해 제안하고, 도서관리시스템을 사례연구로 적용해봄으로써 정보구조 모델링이 사용자도 쉽게 이해할 수 있는 분석 방법

임을 보였다.

사례연구의 분석과정을 통해 살펴본 정보구조 모델링의 장점을 기술하면, 먼저 정보구조 모델링은 속성구조와 행위 구조 다이어그램을 이용하여 시스템의 기능과 데이터를 통합하여 추상적으로 볼 수 있는 형태를 제공해 주므로 사용자와 개발자 모두 쉽게 이해할 수 있는 구조를 제공해 준다. 또한 재산 객체의 속성구조 다이어그램을 통하여 시스템 모델에 대한 전반적인 범위와 개념, 내용을 한눈에 쉽게 파악할 수 있으며, 속성구조를 보는 사용자의 관점에 따라 필요한 행위를 쉽게 추출할 수 있다. 따라서 시스템 분석가와 비즈니스 업무담당자가 공통의 목적과 관점으로 시스템을 분석하여 업무담당자가 특별한 경험이 없어도 이해할 수 있는 비즈니스 시스템 분석 모델이 될 수 있다.

향후 연구 과제로는 소프트웨어개발 전반에 대한 모델링 연구와 함께 모델링 지원 CASE Tool의 개발이 필요하다.

참 고 문 헌

[1] Susan Lilly, "Use Case Pitfalls: Top 10 Problems from Real Projects Using Use Cases", Proceedings of TOOLS USA '99, IEEE Computer Society, 1999.

[2] Hermann Kaindl, "Difficulties in the Transition from OO Analysis to Design", IEEE Software, 1999.

[3] 한/카네기멜론대학 기술교류협회, "최신 소프트웨어 공학기법", V.I.LAND Co.Ltd, 2002.

[4] Hassan Gomaa, "Designing Concurrent, Distributed, and Real-Time Applications with UML", Addison Wesley, 2000.

[5] Evelyn Stiller, Cathie Leblanc, "Project-Based Software Engineering: An Object-Oriented Approach", Addison Wesley, 2002.

[6] Frank Armour, Granville Miller, "Advanced Use Case Modeling Software Systems", Addison Wesley, 2000.

[7] Pan-Wei Ng, "Business Process Modeling and Simulation with UML", Part I: Defining a UML Transaction Model That Maps to RUP Business Models, the Rational Edge, http://www.therationaledge.com/content/apr_02/t_business_ProcessModelling_pn.jsp, 2002.

[8] Ivar Jacobson, Maria Ericsson, Agneta Jacobson, "The Object Advantage: Business Process Reengineering With Object Technology", Addison Wesley, 1995.

[9] "Process and Method: An Introduction to the Rational Unified Process", <http://people.cs.uchicago.edu/~matei/CSPP523/lect4.ppt>, 2002.

[10] Doug Rosenberg, "Top Ten Use Case Mistakes", Software Development, February 2001.

[11] Ellen Gottesdiener, "Top Ten Ways Project Teams Misuse Use Cases - and How to Correct Them", the Rational Edge, June 2002 (Part I), July 2002 (Part II).

[12] 박재년, "정보 구조 모델링에 의한 시스템 분석", 숙명여자대학교 논문집, 제33집, 1992.



이 혜 선

e-mail : hslee@sookmyung.ac.kr

1989년 숙명여자대학교 전산학과 (이학사)

1993년 숙명여자대학교 대학원 컴퓨터과학과 (이학석사)

1989년~1992년 교통개발연구원 연구원

1999년~2000년 숙명여자대학교 아태여성정보통신센터 R&D 교육실 선임연구원

2003년~2004년 서울보건대학 인터넷정보과 초빙교수

2005년~현재 숙명여자대학교 대학원 컴퓨터과학과 박사과정

관심분야: 시스템 분석 및 설계, 정보구조도, 설계패턴, 아키텍처



박 재 년

e-mail : jnpark@sookmyung.ac.kr

1966년 고려대학교 물리학과 (이학사)

1969년 고려대학교 고에너지 물리학과 (이학석사)

1972년 독일 함부르크대학 전산학과

1981년 고려대학교 고에너지 물리학과 (이학박사)

1970년~1972년 독일 국립 입자 가속기 연구소(DESY) 연구원

1972년~1979년 고려대학교 전자계산소 총간사

1978년~1983년 전남대학교 계산통계학과 교수

1987년~1998년 숙명여자대학교 전자계산소 원장

2002년~2004년 숙명여자대학교 이과대학 학장

1983년~현재 숙명여자대학교 정보과학부 교수

관심분야: 시스템 분석 및 설계, 정보구조도, 컴포넌트 기반 개발 방법론