

사용자의 참여를 확장한 시스템 개발 방법의 설계

이 서 정[†] · 박 재 년^{††}

요 약

본 논문에서는 개발된 시스템에 대한 사용자의 만족도를 높이기 위한 방법으로 사용자의 참여를 확대하는 방안을 도입한 시스템 개발 방법을 제안한다. 배경도, 이벤트 다이어그램, 정보구조도, 행위구조도, 관리구조도 및 인터페이스 상태전이도 등이 주요 산출물이며, 이 방법의 특징은 인터페이스의 이동으로 시스템의 흐름을 표시하는 인터페이스 상태전이도와 시스템의 관리사항을 다이어그램으로 표현한 관리구조도 및 시스템의 행정적인 담당자를 배경도의 필수 관련자로 정한 것 등이다. 교수의 연구업무와 현금자동지급기에 이 방법을 적용해 보았고, 그 결과 비즈니스 업무의 개발에 적용될 수 있는 것으로 보인다.

Design for System Development Method with Extended user Participation

Seo-Jeong Lee[†] · Jai-Nyun Park^{††}

ABSTRACT

This paper suggests system development method with extended user participation. Context diagram, event diagram, information diagram, behavior diagram, management diagram and interface STD are staple products. This method introduces interface STD, management diagram and context diagram with administrator as relator. We adapted to research business of professor and ATM(Automatic Teller Machine), then it can be adaptable to business-oriented system.

1. 개 요

최근 사용되고 있는 시스템 개발 방법론들의 대부분은 다이어그램이나 개체 표현의 복잡함으로 일반 사용자가 참여할 수 있는 부분은 상당히 제한적이다. 다이어그램의 작성방법 다이어그램 내에 포함된 개체 간의 복잡한 관계 및 개체의 행위 등을 주어진 방식대로 표현 할 수 있어야 하기 때문이다.

기존의 구조적(structured) 개발방법론이나 나선형(spiral) 개발방법 등의 절차 지향(procedure-oriented) 개발 방법론의 단점인 개발 과정의 시간 낭비 혹은 개발 이후 유지 단계의 한계에 대해 객체 지향 개발 방법론(object-oriented methodology)은 새로운 시도로 받아들여 활발히 연구가 진행되어[1], Booch, Rumbaugh, Jacobson, Shlaer-Mellor 및 Yourdan 등에 의해 꾸준히 이루어 졌으나, 현실적으로는 하나의 방법론을 택하기 보다는 개발자의 필요에 따라 혼합하여 나름대로 사용하고 있다. 이렇게 된 원인은 다음의 현실적인 사항을 고려할 수 있다.

† 정 회 원 : 동덕여대 경영교육학부 대우전임

†† 정 회 원 : 숙명여대 정보과학부 교수

논문접수: 1998년 6월 5일, 심사완료: 1998년 12월 7일

- "객체 지향"을 현실적으로 표현하는 데의 미숙함
- 산출물의 복잡성
- 복잡한 표기법(Notation)

이를 해결하기 위해서는 개발 과정에 사용자의 참여를 확대하는 방안을 고려할 수 있다. 개발 초기부터 사용자가 참여한다면, 사용자의 요구가 즉시 반영될 수 있다는 점에서 개발 결과의 만족도를 높일 수 있을 것이다.

개발 초기부터 사용자가 참여할 수 있는 시스템은 시스템 내부의 프로세스가 일정한 순서를 유지하는 제어 시스템의 경우보다는 사용자의 행동이나 요구에 따라 다른 반응 및 응답을 하는 경우의 시스템에 더욱 적합할 것이다. 즉, 사용자와의 상호작용(interaction)에 의해 사용자의 요구에 따라 다른 반응을 하는 시스템이다. 예를 들어, 자료 중심(data-oriented)의 시스템을 생각할 수 있다.

방법론의 절차를 사용자가 이해하기 쉬워야 한다. 개발 방법의 전 단계 중, 사용자의 참여가 가장 높은 단계는 분석이고, 그 다음은 설계 단계이며, 대부분의 경우 구현 단계에는 사용자의 참여가 거의 없다는 것을 고려할 때, 분석과 설계의 과정은 사용자의 이해를 높일 수 있도록 진행되어 있어야 한다.

본 논문에서는 이러한 점을 완화하여 일반 사용자가 시스템 개발에 더 적극적으로 참여하는 것을 유도하는 정보모델링(IM-UP: Information Modeling-User Participative)을 제안한다.

2. 사용자의 참여 범위를 확장한 정보모델링 (IM-UP)

2.1 기본 개념

하나의 시스템을 구성하기 위해서는 수많은 정보로서 연결되어 있어야 한다. 시스템의 크기에 따라 정보의 양은 차이가 있지만, 기본적으로 그 종류는 세 가지로 나눌 수 있다.

첫째, 사용자의 요구에 응답하기 위한 필수 자료이다. 데이터베이스로 본다면, 파일이나 테이블 내의 정보를 말한다.

둘째, 필수 자료를 움직이게 하는 사용자의 요구가 있어야 한다. 프로그램을 생각할 수 있다.

셋째, 시스템의 정보를 관리하는 자료가 있다. 즉, 시

스템의 보안을 위한 관리자, 비밀번호 등이나 파일, 데이터베이스의 물리적 위치를 설명하는 정보 등이 이에 속한다.

하나의 시스템 내에 존재하는 이런 정보들을 잘 정돈하여 관리를 할 수 있게 하는 것이 시스템 개발에서 해야 할 중요한 과제이다.

일반적으로 소프트웨어 개발 도메인(domain)은 사용자와 시스템으로 구성되며, 사용자는 시스템에 서비스를 요구하고, 시스템은 그에 해당하는 서비스나 제품을 제공한다. 이는 대부분의 경우에 적용될 수 있는 상황으로 시스템을 개발하는 관점을 제시한다. 즉, 사용자가 시스템의 고객이 되며, 시스템을 자극하는 주체가 된다. 시스템 내의 모든 자료나 처리는 사용자의 요구를 충족시키기 위해 준비되고 이루어지는 것이다. 이런 관점에서 생각할 때, 시스템의 개발을 위한 관점은 사용자 즉, 고객의 입장에서 생각되어야 하고, 개발 전 과정을 통하여 이러한 시각의 초점이 일관되게 이루어질 수 있다면, 개발자는 일정을 단축시키는 효과를 얻을 수 있게 된다[2][3].

산출물 중 표기법을 이용하는 산출물은 배경도(context diagram), 이벤트 다이어그램(event diagram), 정보 구조도(information diagram), 행위 구조도(behavior diagram), 관리 구조도(management diagram)와 인터페이스 상태 전이도(ISTD: Interface State Transition Diagram)등 5가지이며, 다이어그램의 목적에 따라 시스템의 도메인(domain), 관련자, 이벤트, 객체, 속성, 관리 사항 등이 표현되며, 객체 간의 연결(link, linked), 집산화(aggregation), 일반화(generalization) 등을 표현할 수 있다.

2.2 개발 절차

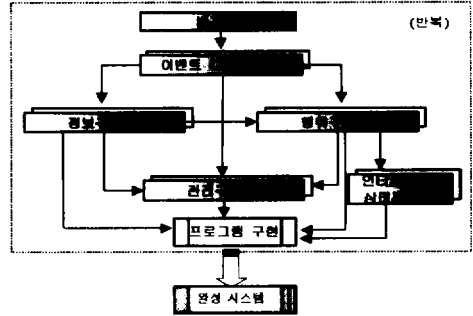
IM-UP의 개발 절차는 크게 분석, 설계, 구현 및 유지보수의 단계로 나누며, 본 논문에서는 분석, 설계 및 구현의 단계를 중점적으로 다루었다. <표 1>은 각 개발 절차에 따른 문서 및 도표 등의 산출물을 보여준다. (그림 1)은 <표 1>의 개발 절차에 따른 산출물에 대한 생성 및 참조사항을 도식화한 것이다.

<표 1>의 굵은 글씨(bold)로 표시된 산출물이 시스템 분석 및 설계의 주요 모델이다. 이들을 중심으로 전체 시스템의 개발이 이루어지며 (그림 2)는 시스템이 모델링 되는 과정을 이들 주요 산출물을 기준으로 표시된 그림이다.

<표 1>의 산출물에 대한 정보와 실제 자료들은 정보저장소(repository)에 저장된다. 산출물 중 다이어그램의 경우, 사용자의 요구에 따라 유동적으로 변경되는 사항을 IM-UP의정보저장소를 이용한 버전관리[4]를 할 수 있다.

<표 1> 개발 절차와 산출물

개발절차	산출물
분석 단계	배경도 이벤트 다이어그램 정보구조도 행위구조도 상세이벤트 명세 행위조건 명세 테이블 관계도 개념적 스키마
설계 단계	관리구조도 인터페이스 상태 전이도 속성 기반값 테이블 동어 테이블
구현 단계	데이터베이스 스키마 사용자 인터페이스 프로그램 모듈
유지보수	테스트 보고서



(그림 2) 주요 산출물과 모델링의 관계

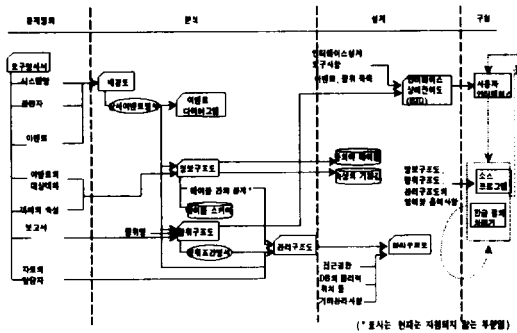
3. 산출물 작성 지침

3.1 분석 단계

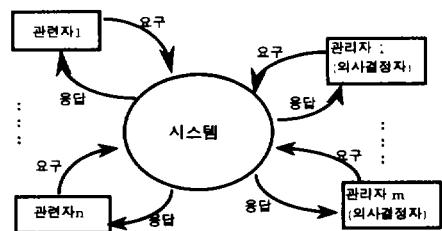
사용자의 요구사항을 파악하고, 개발할 시스템의 범위를 설정하며, 그에 따른 기본적인 자료를 취합하고 정리하는 단계이다[5].

(분석 1) 업무의 범위를 배경도로 작성한다.

배경도는 시스템의 범위를 표현한다. 시스템을 표현하는 타원과 관련자를 표현하는 사각형사이의 이벤트와 응답을 화살표로 표현한다. 관련자는 시스템에 직접 접근 가능하여 시스템에 자극(stimulus)을 줄 수 있는 사람이나 처리기(processor)로 정한다. (그림 3)은 배경도의 개념을 보여준다.



(그림 1) 개발 절차와 산출물의 관계



(그림 3) 배경도

또한 실제 자료를 저장하기 위한 데이터베이스 또는 테이블의 구성은 정보 구조도가 만들어지면, 데이터베이스 스키마가 자동으로 형성될 수 있다[13]. 결국 다음의 자료들이 정보 저장소에서 관리된다.

- 다이어그램의 정보(버전, 마지막 수정일, 물리적 위치, 디렉토리 및 관리자 등)
- 정보구조도의 데이터베이스 스키마
- 프로그램
- 실제 자료

여러 방법론에서 취하는 배경도는 시스템명, 관련자명, 요구 및 응답의 이벤트로 구성되어 되어 있는 것이 일반적이다. IM-UP에서는 이에 덧붙여, 시스템의 담당자를 관련자로 추가하여 시스템 내의 문서, 서식이나 각종 보고서에 관한 생성 및 관리 등의 시스템에 관련된 자료의 관리 측면을 강화했다.

즉, 배경도는 개발 범위를 보여주고, 시스템 내에서

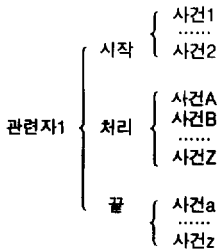
사용되는 자료의 종류를 암시적으로 표현하고 이벤트를 개괄적으로 표시해 주는 역할을 한다 [6].

(분석 2) 관련자와 그에 대한 이벤트를 이벤트 다이어그램으로 정의한다.

이벤트 다이어그램은 관련자가 시스템에 요구하는 이벤트를 WOD(Wanier-Orr Diagram, 워니어-오어 다이어그램)의 형태로 표현한다[7]. WOD는 행위의 처리를 서술식으로 일목요연하게 볼 수 있는 장점이 있고, 그 형태는 (그림 4)와 같이 시작(start), 처리(process), 끝(end) 부분으로 구성되어 있다. 배경도에 나타난 관련자들과 시스템 간의 관계를 상세히 서술식으로 표현하여, 사용자에 대한 역할을 보여주는 도구이다.

이벤트 다이어그램은 배경도에 나타난 관련자의 개수만큼 이벤트 다이어그램은 만들어진다.

'{ '는 집합기호 { } 중 오른쪽 괄호()를 제외한 형태의 { 만을 취한 것이다. 이벤트 다이어그램의 제일 왼쪽, 즉 첫단계 { 앞에는 관련자명을 쓴다. 관련자는 배경도의 관련자와 동일하다. 시작{ 부분에는 관련자가 시스템에 접근하면서 발생하는 이벤트를, 끝 { 에는 시스템을 종료하면서 발생하는 이벤트를 기술한다.



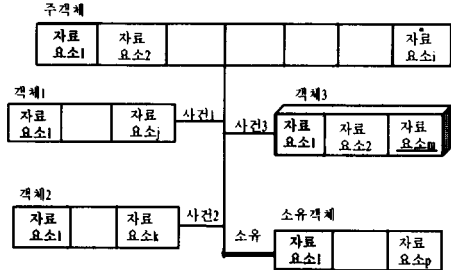
(그림 4) 이벤트 다이어그램

(분석 3) 배경도에 정의한 관련자에 대한 정보 구조도를 작성한다.

정보 구조도는 관련자와 관련자의 요구, 즉 이벤트를 '으로 (그림 5)와 같다. 이 그림을 통해, 사용자는 시스템에 어떤 요구를 할 수 있으며, 그 요구를 만족시킬 수 있도록 어떤 자료들이 구비되어 있어야 하는가 또는 준비되어 있는가를 알 수 있어야 한다.

구현 시 이 정보는 데이터베이스 스키마로 변환되며, 그림의 사각형 한 개의 내부는 속성들로 구분 표현한다. 사용자들은 정보구조도를 통해 이벤트에 요구되는 자료

를 알 수 있고, 개발자는 개념적 데이터베이스 스키마를 알 수 있다.



(그림 5) 정보구조도

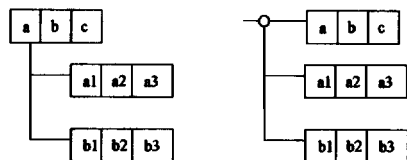
이벤트 다이어그램의 이벤트를 수행하기 위한 자료의 관계를 표현하기 위해서는, 관련자의 기본 자료, 이벤트 그리고 이벤트에 대한 이벤트 객체 및 그 속성에 관한 사항이 표시되어야 한다.

기본자료는 관련자가 시스템에 처음 접근할 때 갖고 있는 또한 가져야 하는 자료를 의미한다. 기본 자료가 정보 구조도의 주객체가 되며, 그 외 구조도의 아래 부분에 표시되는 객체는 소유 객체 혹은 이벤트 객체가 된다.

소유 객체는 관련자의 기본자료 중 중복되는 자료를 모아 별개의 객체를 만든 것이다.

이벤트 객체는 이벤트를 처리하기 위해 참조하는 관련자의 기본자료 이외의 자료를 의미한다.

또한, 객체 내부의 속성 중 사각형에 음영이 있는 속성은 그 객체의 주 키임을 의미한다. 속성명 밑에 밑줄(_)이 그어져 있는 속성은 다른 객체를 참조하는 속성이다. 참조하는 객체는 동일한 정보구조도내에 존재하지 않을 수도 있다. 이에 대한 명세는 관리구조도에 한다. 그 외 집단화와 일반화에 대한 표현은 (그림 6)으로 표현될 수 있으며, 개념적 데이터베이스 스키마는 (그림 7)과 같다.

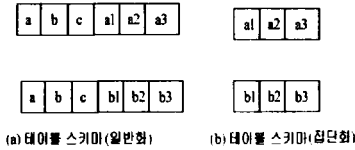


(a) 정보 구조도(일반화) (b) 정보 구조도(집단화)

(그림 6) 객체간의 관계

일반화는 상위 객체의 속성이 상속되고, 집단화는 그

형지 않은 모양이다[8]. 그러나, 실제 구현에 있어서는 두 경우 모두 다이어그램의 주객체의 주 키가 상속된다.

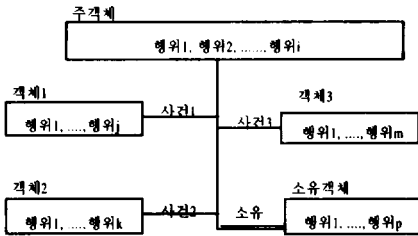


(그림 7) 개념적 스키마

(분석 4) 행위 구조도를 작성한다.

행위 구조도는 시스템에 요구되는 이벤트 처리에 포함된 시스템 행위의 명칭으로 표현한다. 기본적으로는 삽입, 삭제, 변경 등의 행위가 있고, 그 외에 특징적인 행위를 표현한다.

행위 구조도는 정보 구조도에 대응하여 각 객체별로 (그림 8)과 같이 표현되며, 전체 골격은 정보 구조도와 동일하다. 그러나, 각 객체의 내부에는 이벤트를 해결하기 위한 행위명을 적는다.



(그림 8) 행위구조도

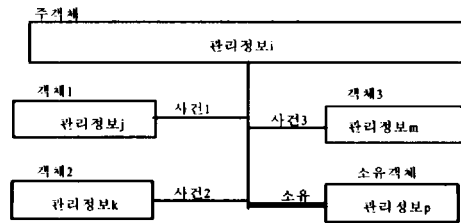
주객체는 수정, 삭제, 삽입, 조회의 4가지 기본 행위를 가지며, 적용 대상은 정보구조도에 표현된 기본 자료 및 소유 객체에 대한 자료를 포함해 사용 권한이 있는 자료이다.

주객체명이 담당자인 경우에는 기본 자료이외의 다른 관련자들의 확인 및 수정 요청 자료에 대해서도 접근 권한이 있지만, 담당자 이외의 다른 관련자는 대부분 개인 자료 이외의 자료에는 접근이 불가능하다.

행위구조도 내의 각 행위는 사용자 인터페이스를 이용해 처리할 수 있도록 개발되어야 한다. 행위 구조도의 작성시, 행위에 대한 조건이나 제약 사항(constraints)은 이벤트 조건 명세(event condition specification)에 등록한다.

(분석 5) 관리 구조도를 작성한다.

앞에서 설명한 구조도들은 데이터베이스의 구성이나 프로그램의 내용 등 시스템에 대한 개념적인 부분들을 보여준다. 이에 대한 물리적인 부분, 사용자 권한, 자료의 물리적 위치, 접근 경로, 데이터베이스 명이나 테이블 명에 대한 사항 등이 (그림 9)와 같이 표현된다. 이때의 각 객체는 정보 구조도의 객체와 일대일 대응 관계를 이루며, 다음의 사항을 명세한다. 관리 구조도의 사항은 분석 단계에서는 모두 명세하지 않아도 된다. 이는 주로 구현 단계에 참조될 사항이므로, 설계 단계까지 작성하면 된다.



(그림 9) 관리구조도

- 관리자 명(manager name)
해당 테이블이나 데이터베이스의 관리자를 등록한다. 업무 담당자나 전산실의 데이터베이스 관리자가 될 수 있다.
- 접근 권한(authority level)
숫자 혹은 문자열로 표시되며, 직급이나 담당업무에 따라 시스템 내에서 정해진 레벨을 의미한다. 이는 사용자가 시스템에 로그 인하는 시점에 권한에 따라 특정 업무에 대해 사용을 제한하는데 참조된다.
- 데이터베이스 명(database name) 또는 테이블 명(table name)
해당 객체의 실제 데이터베이스나 테이블을 의미한다. 시스템에 구현된 물리적인 데이터베이스나 테이블 명을 정의하여 한글 질의 처리를 이용한 정보 검색시 참조한다.
- 데이터베이스디렉토리(database directory)
데이터베이스의 실제 위치를 기록한다. 이는 데이터베이스 명이나 테이블 명과 함께 실제 데이터베이스에 접근하는 경우에 참조된다.
- 연결 테이블(linking table)

객체의 속성 중 다른 테이블을 참조해야하는 경우, 연결 객체로 정의되고 정보구조도에서 속성명에 밑줄로 표현한다. 이런 속성은 실제 스키마로 구현될 시에는 필드 값은 다른 테이블로 연결되는 링크(link)를 갖게 되는데, 이 때 참조하는 테이블을 연결 테이블이라 한다. 연결 테이블은 같은 데이터 베이스 내로 한정했으므로, 테이블 명만 기록한다.

■ 연결되는 테이블(linked table)

연결 테이블의 반대 되는 개념으로 해당 테이블이 참조되는 테이블을 기록한다. 연결 테이블의 링크에 변동이 발생하는 경우의 검증을 위해서 필요하다. 공백이 되면 시스템 내의 다른 어떤 객체에서도 해당 객체를 참조하지 않는다는 것을 의미한다.

■ 프로그램 모듈명(program module name)

해당 이벤트의 행위는 행위 조건 명세를 참조하여 사용자 인터페이스의 일부에 구현이 된다. 하나의 이벤트에 대해 최소 한 개 또는 그 이상으로 구현이 되는데, 이에 대한 제어 프로그램 명 또는 모듈명을 기록한다.

■ 행위 조건 명세 파일명(behavior condition specification file name)

행위의 제약 사항에 대한 명세를 행위 조건 명세라 하고, 이는 텍스트 형태로 파일로 저장한다. 하나의 행위에 대해 하나의 파일이 될 수도 있고, 하나의 이벤트에 대해 하나의 파일이 될 수도 있다. 그 기준은 개발 환경에 따라 다를 수 있으나, 행위 조건 명세 파일명을 등록을 하면, 파일을 분리하는 기준은 임의로 할 수 있다.

■ 상세 이벤트 명세 파일명(detail event specification file name)

상세 이벤트 명세는 이벤트가 다른 이벤트의 묶음, 즉 대표 이벤트라면, 그에 대한 자세한 이벤트를 정의한 명세이다. 공백이 될 수 있다.

(분석 6) 구조도 검사기로 크로스 체크(cross checking)을 한다[4].

입력된 속성명과 업무에서 사용되는 보고서나 자료의 항목명 이 간의 불일치를 검사한다.

(분석 7) 오류 또는 변동 사항이 있을 경우 수정한다.

분석 과정 중, 필요한 경우 동의어 테이블, 기본 속성값 테이블, 상세 이벤트 명세 및 행위 조건 명세를 작성한다. 동의어 테이블, 기본 속성값 테이블은 설계 단계에 완성된다.

3.2 설계 단계

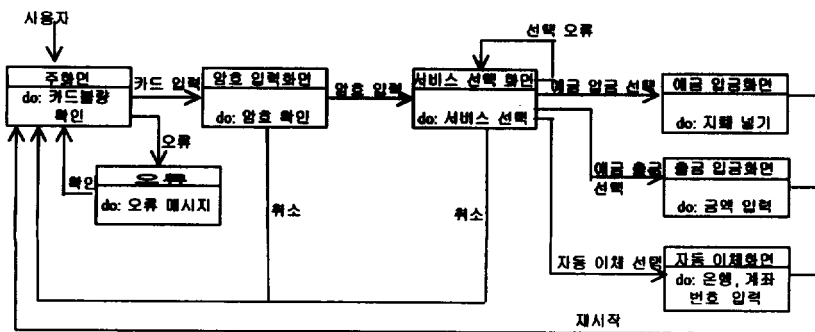
(설계 1) 관리 구조도를 완성한다.

분석 단계에서 완성하지 못한 관리 구조도의 요구 사항을 추가하여 완성한다.

(설계 2) 행위명세를 작성한다. - 제어 객체의 설계 행위구조도의 각 행위에 대한 상세 명세나 조건 등에 대한 명세를 한다. 이는 구현 단계에서 프로그램 모듈의 구현으로 연결된다.

(설계 3) 인터페이스 상태 전이도를 작성한다. - 사용자 인터페이스 객체의 설계

일관된 시각으로 시스템의 흐름을 이해하기 쉽게하도록 인터페이스에 대한 상태 전이도를 작성한다. (그림 10)은 ATM(Automatic Teller Machine)의 상태전이도이다. 다른 방법론의 상태전이도와 비슷하지만, 작성 기준이 사용자 인터페이스라는 점에서 뚜렷이 구별될 수 있다.



(그림 10) ATM의 인터페이스 상태전이도

3.3 구현 단계

구현 단계는 분석 및 설계 단계의 산출물을 근거로 개발 환경에 맞게 프로그래밍이나 데이터베이스를 구축하는 단계이다.

(구현 1) 관리 구조도와 정보 구조도에 의해 데이터베이스 스키마를 형성한다.

데이터베이스 또는 테이블 스키마는 정보구조도에 의해 생성될 수 있으며, 이를 구현 단계에서 개발자의 의도에 따라 수정이 가능하다.

(구현 2) 질의어 명세를 생성한다.

시스템 운영상 자주 사용할 질의어 또는 한번 이상 사용된 질의어를 질의어 명세에 등록하여, 반복된 질의 처리가 요구될 경우 질의어 명세에서 선택할 수 있도록 한다.

(구현 3) 제어객체의 구현-프로그램 모듈을 생성한다.

이벤트 다이어그램이나 행위 구조도의 이벤트 또는 행위를 구현하는 과정이다. 개발 환경에 맞는 프로그램 언어로 구현한다. 프로그램 모듈명은 관리구조도에 기록된다.

(구현 4) 사용자 인터페이스 객체의 구현-사용자 인터페이스를 개발한다.

설계단계에서 나온 사용자 인터페이스를 구현하고, 제어 객체와 연결한다. 이 과정은 개발 도구에 따라 (구현 3)단계와 동시에 진행될 수도 있다.

4. 특 징

4.1 인터페이스 상태전이도(ISTD)

객체 지향 개발 방법은 자료를 정의하고, 시스템의 흐름을 정의하고, 시스템 흐름이 자료를 참조하기 위한 기능에 대한 정의로 구분될 수 있다. 동적 모델링은 시스템의 흐름을 정리하여 표현하는 과정이다[9].

상태 전이도는 시스템의 흐름을 동적으로 표현하는 도구으로써 사용자 및 개발자에게 시스템에 대한 이해를 높이는데 도움을 주므로, 대부분의 방법에서 상태 전이도를 채택하고 있다.

그러나, 기존 방법의 상태전이도(STD)는 프로세스(process)를 기준으로 프로세스간의 전이(transition)를 표현하거나, 이벤트(event)를 기준으로 이벤트의 수행 전후의 상태(state) 변화를 표현하기도 했다.[10][11]

상태전이도는 주로 설계 단계에서 작성되는데, 원칙적으로는 하나의 객체나 시스템을 대상으로 하나의 상

태전이도가 만들어진다. 그러나, 실제로는 하나로 그리기에 너무 복잡해져 이벤트 단위로 나누어 그린다. 그 결과 시스템이 큰 경우 수십, 수백의 상태 전이도가 작성되어 이를 참조하고 수정하는 작업이 어려워진다.

IM-UP에서는 이 점을 극복하고, 시스템의 흐름을 표현하기 위해, 구현된 사용자 인터페이스의 한 화면을 하나의 상태로 하여 그들간의 관계를 표현하는 인터페이스 상태전이도를 제안했다. 인터페이스 상태 전이도를 통해, 사용자는 소프트웨어의 흐름을 이해하고, 그에 따른 입출력 사항을 도면을 통해 알게 되어 시스템 전체에 대한 이해도를 높일 수 있다.

4.2 새로운 배경도-업무 담당자를 필수 관련자로

기존의 방법에서 사용하는 배경도는 시스템명, 관련자명, 자료 흐름 또는 이벤트명 등으로 구성되어 있으며, 관련자의 자격에 대해서는 '시스템에 관련된 사람이나 기관(부서)'[12] 이외의 다른 언급은 없었다. 관련자의 정의는 시스템의 개발 범위를 암시하는 중요한 요소가 되는데 반해, 관련자의 범위를 결정하는데 있어서는 애매했다.

IM-UP에서는 관련자는 '시스템에 직접 접근할 수 있는 권한을 가진 사람 또는 기관(부서)'로 정하고, 시스템의 행정적인 담당자를 필수 관련자로 설정했다. 이렇게 함으로써, 담당자를 결정하는데 있어서의 애매함을 해소했다.

업무상의 담당자가 모든 시스템 외부의 행정적인 관련자 - 관공서, 은행, 타 기관 혹은 부서 등 - 의 창구 역할을 하며, 시스템 내부에서 사용되거나 생성되는 행정 문서에 관련된 업무를 담당한다.

담당자를 배경도에 도입한 효과는 배경도를 통해 업무의 범위를 파악하기가 이전 보다 명료해졌고, 구현에 있어서도 기존의 주체가 분명히 드러나지 않던 이벤트나 출력물에 대한 정의를 분명히 할 수 있게 되었다.

4.3 관리구조도의 도입

시스템에 대한 행정적이거나 데이터베이스의 보안 등에 관한 사항을 자료의 구조를 표현하는 정보구조도와 같은 형태의 다이어그램으로 표현함으로써 사용자나 개발자의 이해를 돕는다.

이는 다른 방법론에서는 채택하고 있지 않지만 시스템의 관리체계를 한 눈에 알아볼 수 있도록 도와주는 도구이다.

5. 사례 분석

본 논문에서 제안한 방법을 교수의 연구업무와 현금 자동지급기(ATM)에 적용해 보았다. 그 결과, 프로세스 보다는 자료 중심의 비즈니스 시스템으로 사용자의 요구가 다양하게 발생하고, 그에 대한 응답이 단순히 물건이기 보다는 화면 표시나 문서 등의 자료가 되는 경우에 적합하다는 것을 알 수 있다. ATM와 연구 업무의 예는 사용자가 시스템에 다양한 요구를 하며, 이를 이용한 처리는 비교적 간단하여, 본 논문에서 제안하는 방법의 표현으로 시스템의 사용자나 서비스의 범위를 쉽게 파악할 수 있지만, 자동 제어 시스템의 경우에는 본 논문의 방법이 적합하지 못했다.

6. 결 론

본 논문에서는 개발된 시스템에 대한 사용자의 만족도를 높이기 위한 방법으로 사용자의 참여를 확대하는 방안을 도입한 시스템 개발 방법을 제안하였다.

다이아그램의 작성방법 다이아그램 내에 포함된 개체 간의 복잡한 관계 및 개체의 행위 등의 표현을 일반 사용자도 쉽게 이해할 수 있고 개발자는 이를 바탕으로 시스템을 개발할 수 있도록 절충하기 위해 배경도, 이벤트 다이아그램, 정보구조도, 행위구조도, 관리구조도 등의 산출물을 이용한다. 특히, 인터페이스 상태 전이도와 관리구조도의 도입은 개발되는 시스템의 흐름과 데이터 및 행정적인 관리사항을 표현 할 수 있다.

적용범위는 비즈니스 업무 개발에 더욱 적합할 것으로 보이며, 주요 산출물 외에 부가적으로 요구되는 산출물의 관리 및 데이터의 관리에 대한 구체적인 방안이 연구 중이다.

참 고 문 헌

[1] Donald G.Firesmith, Object-Oriented requirements analysis and design, John Wiley & Sons, 1993.
 [2] 박재년, "정보구조모델링에 의한 시스템 분석", 숙명여자대학교 논문집, 33집, pp.677-692, 1992년 12월.
 [3] Martin Fowler, "Analysis patterns and business object," proc. of OOPSLA'96, 1996.
 [4] 정미선, "정보구조도의 자료요소 검사기 설계와 구현에 관한 연구", 숙명여자대학교 석사논문, 1995년

12월.

[5] 이주현, "실용 소프트웨어 공학론", 법영사, 1993년.
 [6] 정기현 등, "실행 가능한 표현 방법으로 확장가능한 자료 흐름도," 정보과학회 논문지, 24권 8호, pp.829-841, 1997년 8월.
 [7] 박재년, "기초전산학", 교학사, 서울, pp.337-366, 1997
 [8] Mark A. Weiss, Data structure and algorithm analysis, Benjamin cummings, pp.223-228, 1992
 [9] 김진수, 김정아, 이경환, "객체지향 동적 모델링 기법의 정형화", 한국정보처리학회 논문지, 4권 4호, pp.1013-1024, 1997년 4월.
 [10] J.Rumbaugh, Object oriented modeling and design, Prentice Hall, 1991.
 [11] Ivar Jacobson, The object Advantage : Business process reengineering with object technology, Addison-Wesley, 1994.
 [12] P.Coad, E.Yourdan, Object oriented analysis, 2nd ed., Prentice Hall, 1990.
 [13] 이서정, 박재년, "IFM에 기반한 한글 질의 처리기의 설계와 구현," 한국정보처리학회 논문지, 4권 10호, pp.2445-2452, 1997년 10월.

이 서 정

e-mail : sjlee@cs4000.dongduk.ac.kr
 1989년 숙명여대 전산학과 졸업
 1991년 숙명여대 대학원 전산학과 졸업(전산학, 이학석사)
 1998년 숙명여대 대학원 전산학과 졸업(전산학, 이학박사)

1998년 3월~현재 동덕여대 교양교직학부 대우전임
 관심분야 : 시스템 개발 방법론, Requirement Engineering, KMS 등

박 재 년

e-mail : jnpark@cs.sookmyung.ac.kr
 1979년~1989년 전남대학교 전산 통계학과 교수
 1983년~현재 숙명여대 정보과학부 교수
 관심분야 : 시스템 개발 방법론, 시뮬레이션, 메타DB 등