정보기술응용연구
제 3 권 제 4 호
2 0 0 1 년 12 월

시스템 구축 과정에서 소프트웨어 자동생성 도구의 적용

정일주*

요 약

1970대 이후 CASE 도구들은 꾸준히 발전되어 오고 활용은 보편화되고 있다. 많은 CASE 도구들은 보다 확대된 기능을 제공하기 위하여 통합되고 연계되고 있다. 한편 E-R 모델은 모델링 도구로서 널리 사용되고 있으며 대부분의 CASE 도구들이 E-R 모델을 지원하고 있다. 본 연구에서는 E-R 모델에 내재한 여러 가지 특성과 응용시스템의 기본 요소들 사이에 관계를 설정하고 이를 이용하여 응용시스템을 자동적으로 생성할 수 있는 방법을 제시한다.

먼저 E-R 모델로부터 일정한 규칙에 의해 업무 프로그램, 프로그램간의 링크, 지속적 데이터 등으로 구성된 정형적 응용시스템이 생성된다. 정형적 시스템은 이용자를 위한 인터페이스 설계를 거쳐 최종적인 응용시스템으로 생성된다. 본 연구에서는 제안된 자동생성 과정의 실현가능성을 가상의 시스템을 대상으로 검증해 본다.

본 연구에서 제안된 응용시스템의 자동생성 방안은 CASE 도구들의 모델링 기능과 코드 생성 기능을 연계하고 통합할 수 있는 하나의 대안을 제시할 것으로 기대된다. 동시에 종래에 데이터모델링 도구로서 주로 인식되어 온 E-R 모델에 관한 연구를 응용시스템 전체에 대한 모델링의 관점으로 확대할 수 있을 것으로 기대된다.

*) 홍익대학교 상경대학 상경학부 chungic@hongik.ac.kr

1. 연구의 배경과 목적

본 연구의 목적은 E-R 모델을 이용하여 응용시스템을 자동생성하는 새로운 방법을 찾아보는 것이다. 구체적으로 본 연구에서는 E-R 모델의 잠재된 특성을 도출하여 이를 응용시스템의 구조적 특성에 체계적으로 연계할 수 있는 방안을 도출하고 이를 기반으로 CASE의 기본 명제인 응용시스템의 자동생성 과정을 새로운 시각으로 조명해 보고자 한다.

개체관계(E-R) 모델은 1970년대 중반 P. Chen에 의해 제안된 이래 시스템과 데이터의 논리적 구조를 표현하는 유용한 방식으로 광범위하게 이용되어 왔다.[1],[3],[4][5],[6],[9],[11] E-R 모델은 관계형 데이터베이스, 정보 모형, 요구분석 등 다양한 논리적 체계의 표현 도구로서 활용되고 있고, 최근에는 CASE 도구, 추론적 모델링, 전문가 시스템 및 객체지향 모델 등의 분야로 연구와 적용범위가 확대되고 있다.[9],[10] E-R 모델이 데이터 모델링의 주류로서 활용되고 있는 이유로서, 상대적인 사용의 용이성, 광범위한 CASE 도구의 지원, 그리고 개체와 개체들 간의 연관성이 실제 세계를 모델링 하는 자연스러운 개념이라는 믿음의 세 가지를 들 수 있다.[9]

한편, CASE(Computer-aided systems engineering)는 정보기술의 시스템 개발 활동, 테크닉 그리고 방법론에 대한 적용을 말한다.[15] CASE의 역사는 1970년대 중반에 미국 미시간 대학의 ISDOS 프로젝트에서 개발된 문제정의 언어와 분석기인 PSL/PSA에서 시작되었고 그 이후 Index Technology사에서 개발한 Excelerator의 큰 성공은 CASE의 발전의 획기적 전환점을 마련하였다.[8],[15] 오늘날에는 수많은 CASE 도구들이 다양한 시스템 개발자들에게 사용되고 있다.

CASE 도구들은 크게 정보요구, 시스템 분석, 설계 등에 활용되는 Popkin Software의 System Architect와 같은 Upper-CASE 도구와 상세 설계, 시스템 개발, 프로그램 생성 분야에 적용되는 PowerSoft의 Powerbuilder와 같은 Lower-CASE 등으로 구분할 수 있다. CASE 도구들은 또 도형화, 프로토타이핑, 데이터 설계, 프로그래밍, 프로젝트 관리, 유지보수, Reverse engineering, 코드 생성 등 시스템 개발 생명주기의 거의 모든 단계에 걸쳐 사용되고 있다.[8],[11],[14] 최근 CASE의 사용은 특히 데이터 모델링, 데이터베이스 스키마 설계, 프로세스 모델링 분야에 계속 확대되어 가고 있다. 그러나 CASE를 사용하는 시스템 개발자의 입장에서 보면 그가 필요로 하거나 사용하기 원하는 모든 CASE 도구들을 어느 한 소스(Source)에서 찾는다는 것은 거의 불가능하다.[15] Gane은 향후 CASE 발전의 중요한 경향으로서 광범위한 통합을 들고 있으며 많

은 CASE 개발자들은 코드생성기와 모델 정의 사이에 다리를 놓아 모델링과 코드 생성 과정을 더욱 더 밀접하게 통합하고 아울러 더 큰 모델로부터 응용시스템 전체를 생성하는 방향으로 움직이고 있다고 평가하고 있다.[8] 이 문제는 앞으로 상당 기간 CASE 분야 연구의 중요한 과제가 될 것으로 예상된다.

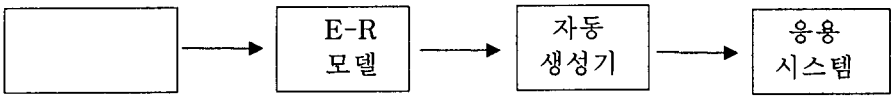
E-R 모델은 지속적으로 CASE의 대상이 되어왔다. 데이터 모델링을 수행하는 대부분의 CASE와 전문가 도구들은 E-R 모델에 기조하고 있다.[2] 많은 개발자들은 E-R 모델에 기반을 둔 CASE 도구들을 시스템개발 주기에서 특히 데이터 모델링, 데이터베이스 스키마 설계 단계에 주로 사용하고 있다. E-R 모델과 CASE를 연계한 연구에 있어서는 E-R 모델을 CASE의 관점에서 조명하면서 다양한 측면에서 CASE의 기능을 확대해 가는 방향의 연구가 활발히 진행되고 있다. Francett는 프로세스와 데이터 모델링을 연계하여 DFD에서 ERD 그리고 RDB로 설계하는 과정을 CASE 도구로 지원하는 Silverrun/ERX를 개발하였다.[7] Siau 등은 일반화, 추상화 등을 지원하는 EERD(Enhanced E-R Diagrammer)를 개발하였다.[12]

E-R 모델에 기초한 대표적인 CASE 도구들 중의 하나인 Logic Works의 ERwin은 초기의 데이터 구조의 설계에서 Microsoft의 Visual Basic을 이용한 클라이언트 화면 생성, Powerbuilder, Clipper, FoxPro 등과의 인터페이스 제공 등의 방향으로 진화하고 있다.[13] 한편 역시 CASE 도구인 InfoModeler에서는 ORM(Object role model)과 E-R 모델간의 자동변환을 지원하고 동시에 기존의 데이터베이스에서 E-R 모델을 도출하는 리버스 엔지니어링 기능으로 확장하고 있다.[13] 또 McChesney 등은 요구분석 및 획득에 E-R 모델과 CASE 도구를 활용하는 방안을 제시하였다.[10] 지금까지의 E-R 모델과 CASE를 연계한 연구와 개발 동향을 종합해 보면 모델링과 데이터베이스 설계를 연계하면서 DFD 등 다른 분석 기법과 연결하는 방향이 주종을 이루고 있는 가운데 E-R 모델링과 Powerbuilder 등 Lower CASE 도구와의 연계 및 통합이 부분적으로 시도되고 있는 것으로 평가된다.

이러한 배경에서 본 논문에서는 두 방향에서 CASE와 E-R 모델에 관한 연구에 접근한다. 첫째, CASE에 있어서 CASE에서 데이터 모델링과 데이터베이스 설계에 머물러 있던 E-R 모델의 역할을 응용 시스템의 모델링으로 확장하는 가능성을 점검해 보고, 둘째, 앞서 언급한 바와 같이 CASE 분야의 향후 과제로 떠오르고 있는 Upper CASE 와 Lower CASE로 분할되어 있는 CASE 도구들을 연결해 줄 수 있는 다리의 역할을 할 수 있는 방안을 탐색해 보는 관점에서 접근한다. .

만약 E-R 모델과 응용시스템 그리고 응용시스템의 생성기 사이에 체계적인 매핑 관계가 존재하고 이를 구현하는 생성기가 존재한다면 시스템 설계자는 [그

림 1]과 같이 먼저 현실세계를 E-R 모델을 통하여 표현하고 이를 자동생성기를 이용하여 하나의 응용시스템으로 생성함으로써 시스템 개발 과정 전체를 매우 효율적으로 마무리할 수 있을 것이다. 따라서 본 연구의 결과는 CASE 개발자들에게 데이터와 프로세스 모델링 단계 그리고 상세 설계 및 프로그램 생성단계를 체계적으로 연결시킬 수 있는 하나의 대안을 제시해 줄 수 있을 것으로 기대된다. 아울러 E-R 모델에 대한 연구에 있어서 E-R 모델을 역할을 데이터 모델링에서 프로세스의 모델링 그리고 궁극적으로 응용시스템 전체의 모델링으로 단계적으로 확대해 갈 수 있는 가능성을 제시해 줄 것으로 기대된다.



[그림 1] 소프트웨어의 생성 과정

본 논문의 2 장에서는 E-R 모델을 통해 응용시스템을 생성하는 기본적인 아이디어와 과정을 소개하고 이어서 3 장에서는 E-R 모델과 응용시스템 사이의 관계에 기초하여 응용시스템을 체계적으로 생성할 수 있는 자동생성기의 요소와 기능을 정의한다. 4 장은 케이스 연구로서 앞에서 제시한 응용시스템 자동생성기를 이용하여 하나의 응용시스템을 생성하는 과정을 가상의 시스템을 예로 들어 검증해 본다.

2. 개체관계 모델과 응용시스템의 연관성

E-R 모델은 어떤 대상을 표현하는 개념적 체계로서 개체, 개체타입, 관계, 관계타입, 속성, 값타입 등의 개념을 제공한다. E-R 모델은 이미 보편화되어 있으므로 자세한 설명은 생략한다.

응용시스템은 운영체제, 네트워크 관리 도구 등의 소프트웨어와 달리, 시스템 설계자에 의해 이용자의 정보 요구를 조사하여 이를 충족시킬 수 있도록 설계되고 구축된다. 본 연구의 관점에서 볼 때, 응용시스템은 다음의 네 요소로 구성된다. 첫째, 응용시스템의 기본적 요소는 업무 프로그램으로서 이용자의 정보 요구를 충족시켜주고 각종 거래를 수행할 수 있도록 해 준다. 둘째, 응용시스템의 업무 프로그램들은 서로 연결되어 상호간에 호출/피호출 혹은 데이터의 전달 등의 관계로 엮어진다. 이용자의 업무는 논리적 관계를 가진 일련의 프로그램들에 의해 수행된다. 이를 위해 개별적인 컴퓨터 프로그램들을 호출하거나 종료시키는 통제 프로그램이 존재한다. 셋째, 응용프로그램들을 전체적으로 논

리적으로 통합하여 이용자와 연결시켜주는 통상 "메뉴" 라고 부르는 요소가 존재한다. 마지막으로 응용시스템은 이용자가 사용하게 될 각종 정보와 데이터를 수록하는 데이터베이스를 가지고 있다.

E-R 모델을 통하여 응용시스템을 정의 및 설계하고, 이를 실제로 이용자의 업무 수행에 사용될 수 있는 시스템으로 구현하기 위해서는 E-R 모델과 응용시스템 양자 사이에 일관성 있고 체계적인 상호관계가 존재해야 하는 바 여기에는 두 가지 가정이 필요하다. 첫째, E-R 모델은 데이터뿐만 아니라 프로세스 모델링을 위한 개념들을 포함하고 있다. 그리고, 둘째, E-R 모델의 특성과 응용시스템의 특성들 사이에는 일정한 매핑 관계가 존재한다. 이러한 연관관계는 자동생성과정에서 규칙으로 표현된다. 구체적으로 말하면, 개체관계 모델의 개체, 관계, 개체/관계 타입, 속성, 값타입 및 개체들 사이의 기수성 등의 기본개념들과 앞서 정의한 응용시스템의 요소들을 체계적으로 연관시킬 수 있는 사상(寫象: Mapping) 관계가 설정되어야 한다.

자동생성 과정은 어떤 장치 하나를 의미하기보다는 자동생성에 관련된 일련의 절차와 도구들이 전체적으로 통합된 하나의 프로세스 즉, 과정으로 보는 것이 본 논문의 목적에 부합된다. 전체적인 응용시스템의 자동생성 과정은 다음과 같은 절차로 이루어진다.

스텝 1: 설계자는 대상이 되는 시스템을 E-R 모델로 표현한다.

스텝 2: 자동생성기는 E-R 모델을 입력받아 주어진 생성 규칙에 따라 응용시스템의 기본 요소들을 자동적으로 생성한다. 여기서 생성되는 시스템은 특정한 이용자를 가정하지 않는 정형적 시스템이다.

스텝 3: 자동생성기는 E-R 모델의 특성에 기초하여 업무프로그램들 사이의 링크 정보를 생성하고, 이어서 데이터베이스를 생성한다.

스텝 4: 개발자는 자동생성기와 상호작용을 통해 이용자 인터페이스를 설계한다. 여기에는 이용자의 업무에 적합한 업무 프로그램들을 자동생성된 프로그램의 풀(Pool)에서 선택하는 작업, 업무프로그램의 속성을 선택하는 작업 그리고 화면을 설계하는 작업등이 포함된다.

스텝 5: 자동생성기는 업무 프로그램, 링크, 데이터베이스, 인터페이스 설계 등 응용시스템의요소들을 통합하여 응용시스템의 생성을 완료한다.

3. 응용시스템 자동생성기

여기서 자동생성기는 일련을 규칙을 의미한다. 어떤 대상을 표현하는 E-R

모델을 보면 그것으로부터 도출될 수 있는 프로그램의 수나 형태에 있어서 매우 큰 가변성이 존재한다. 예컨대, 뒤에 제시된 [그림 7]의 개체관계도가 의미하는 정보 접근 및 처리 프로그램의 종류와 수는 엄청나게 많을 것이다. 어떤 막강한 능력을 가진 프로그램 한 개가 이 E-R 모델이 함축하고 있는 기능 전체를 포함할 수도 있고, 수백 개의 작은 프로그램들로 구분될 수도 있다. 따라서 응용시스템의 자동생성에 있어서 가장 중요한 문제는 주어진 E-R 모델로부터 응용시스템의 "프로그램"을 정의할 수 있는 규칙을 만들어 내는 일이다. 이러한 자동생성 규칙을 정의하기 위해서는 다음과 같은 4 가지 원칙이 필요하다.

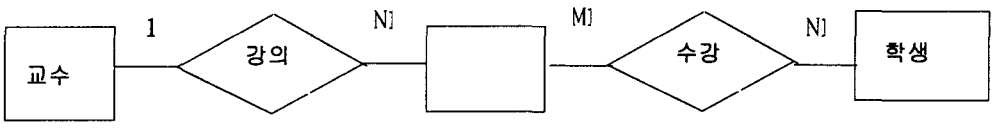
첫째, 자동생성기는 주어진 E-R 모델을 완전하게 표현할 수 있어야 한다. 이는 즉, 완전성의 원칙으로서 자동생성기에 의해 생성된 응용시스템은 주어진 E-R 모델이 표현하고 있는 정보와 기능의 범위를 완전하게 나타낼 수 있어야 한다는 의미이다.

둘째, 자동생성기가 생성하는 프로그램들은 효율적이어야 한다. 다시 말해, 프로그램의 구조는 단순한 형태가 되어야 한다.

셋째, 자동생성기는 이용자의 특성과 요구에 의해 수정될 수 있어야 한다. 즉, E-R 모델은 여러 이용자의 관점이 통합되어 있는 것으로 가정하고 설계된 E-R 모델의 일부분을 특정 이용자의 관점으로 응용시스템에서 구현할 수 있어야 한다.

마지막으로 자동생성된 시스템은 그 시스템에 대한 미래의 여러 가지 변경 요구를 효과적으로 수용할 수 있는 융통성을 가져야 한다. 다시 말해, 자동생성된 응용시스템은 생성과 함께 그대로 고정되어 버리는 것이 아니고 크고 작은 (수작업) 수정의 대상이 되기 때문에 향후 이용자에 의한 수정과 확장이 용이하도록 하는 것은 매우 중요한 원칙이다. 이러한 원칙에 의해서 자동생성 과정에서 필요한 생성규칙을 제시한다.

설명에 사용될 예로서 [그림 2]의 E-R 모델을 사용한다. [그림 2] E-R 모델에서는 개체타입 간의 기수성을 함께 보여 주고 있다. "1"은 하나를 나타내고 "N" 혹은 "M"은 다수를 나타낸다.



[그림 2] 학사관리 시스템의 개체관계도

3.1 정형적 응용시스템의 생성

3.1.1 업무 프로그램의 생성

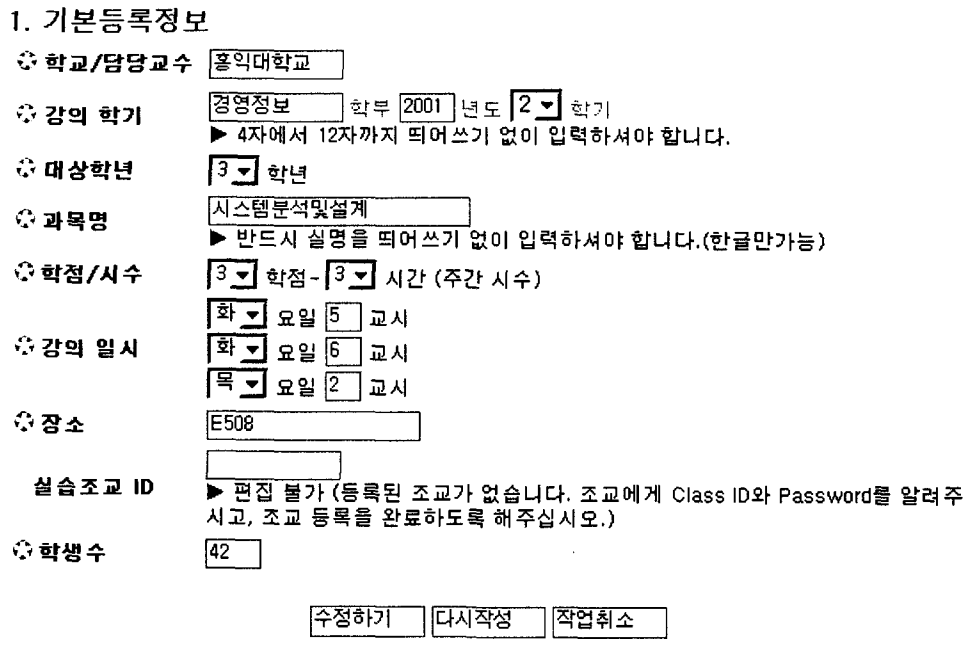
응용시스템의 가장 중요한 구성요소는 업무 프로그램이다. E-R 모델로부터 응용시스템의 업무 프로그램이 자동적으로 생성되기 위해서는 업무 프로그램의 특성과 E-R 개념 사이의 연계성이 체계적으로 설정되어야 한다. 이러한 연계성은 일련의 생성규칙들로 표현된다.

<규칙 1>: 자동생성되는 업무 프로그램은 등록, 삭제, 수정 및 조회의 네 가지로 세분된다.

기본적으로 E-R 모델과 동일한 표현력을 가진 응용시스템은 각각의 개체와 관계들을 생성, 삭제, 수정 및 조회하는 등의 처리할 수 있어야 한다. 또한 이 네 가지 작업은 별도의 프로그램으로 제공되어야 한다. 별도의 프로그램이 생성되어야 하는 이유는 예컨대, [그림 2]의 학사관리시스템에서 학생들은 등록된 클래스를 조회만 할 수 있고, 교수는 추가로 나머지 세 가지 작업을 할 수 있기 때문이다. <규칙 1>은 다음에 제시되는 <규칙 2와 3> 그리고 <확장규칙 1, 2, 3>과 결부되어 적용된다. 다시 말해, 생성되는 업무 프로그램들에게는 어느 것이나 <규칙 1>이 적용된다.

<규칙 2>: 각각의 개체타입은 독립된 업무 프로그램으로 변환된다. 개체타입의 모든 속성은 해당 프로그램에서 입출력의 대상이 될 수 있다.

E-R 모델에서 가장 기본적인 단위는 개체타입과 관계타입이다. <규칙 2>는 각각의 개체타입은 개별적인 프로그램으로 생성되도록 규정하고 있다. 예컨대, 학사관리 시스템에서 "과목"이라는 개체타입은 [그림 3]과 같은 업무 프로그램으로 생성될 수 있다. 과목 개체가 가진 속성들, 예컨대, "과목명", "강의 학기" 등은 화면에서 입출력 항목으로 정의된다.

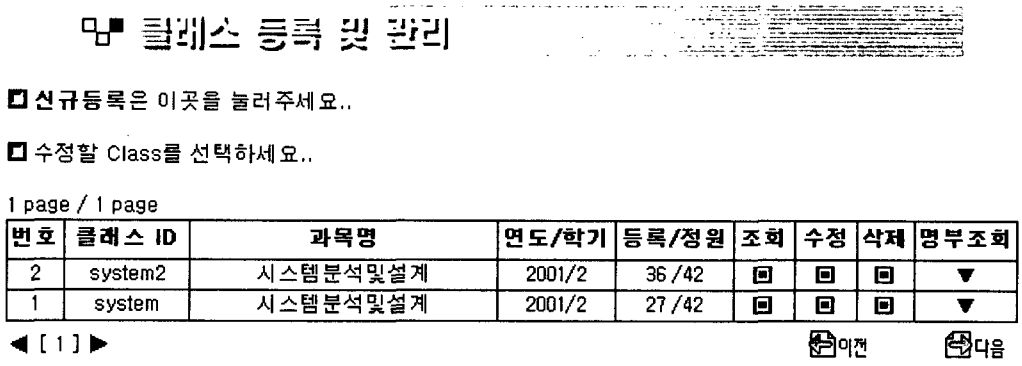


[그림 3] 과목 개체를 등록하는 프로그램

<규칙 2>에 의해 생성되는 프로그램은 한 화면에서 하나의 개체를 다루는 단수 개체에 대한 프로그램과 복수의 개체를 다루는 프로그램의 두 가지로 구분될 수 있으나 이러한 프로그램 내부의 특성은 생략하기로 한다.

<규칙 3>: 각각의 관계타입은 독립된 업무 프로그램으로 변환된다. 관계타입의 모든 속성은 입출력의 대상이 될 수 있다. 그 관계타입에 참여하는 개체타입의 속성들은 해당 프로그램에서 오직 출력의 대상이 될 수 있다.

예를 들면, 다음의 프로그램은 어떤 교수가 강의하는 과목의 목록 즉, 교수와 과목 개체타입 사이에 존재하는 "강의"라는 관계타입에 속한 하나의 관계를 보여주고 있다. 담당교수의 성명이 화면에 나타나지 않은 것은 현재 교수 ID로 로그인 한 상태이기 때문이다. <규칙 3>에서 중요한 것은 관계타입에 참여하는 개체타입의 속성들도 이 프로그램에서 다루지만 그 범위는 오직 조회의 대상이 될 뿐이라는 점이다. 예를 들어, [그림 4]에서 "과목명"은 개체타입 "과목"의 속성으로서 관계타입 "강의"의 속성이 아니며 따라서 오직 조회만이 허용된다.


클래스 등록 및 관리

신규등록은 이곳을 눌러주세요..

수정할 Class를 선택하세요..

1 page / 1 page

번호	클래스 ID	과목명	연도/학기	등록/정원	조회	수정	삭제	명부조회
2	system2	시스템분석및설계	2001/2	36 /42	▣	▣	▣	▼
1	system	시스템분석및설계	2001/2	27 /42	▣	▣	▣	▼

◀ [1] ▶ 이전 다음

[그림 4] 특정 교수가 강의하는 과목

E-R 모델에서 관계타입의 속성은 연관된 개체타입의 증명속성과 관계타입 자신의 속성들로 구성되어 있다. 따라서 응용시스템에서 관계타입에 참여하는 개체타입의 증명속성만이 화면에 나타난다면 ([그림 4]에서 클래스 ID) 그 개체가 무엇인지 쉽게 구분이 되지 않아 이용자 관점에서 불편할 것이다. 고로, 비증명 속성인 "과목명"을 아울러 제시한다. 따라서 <규칙 3>은 <규칙 2>와 달리 관계타입의 속성에 추가하여 참여 개체타입의 비증명 속성들도 조회하도록 하고 있다.

기본적으로 자동생성기는 모든 속성에 대한 조회를 가능하게 하고 있다. 그러나 이용자는 자동생성 과정의 마지막 단계에서 이러한 결정, 다시 말해 화면에 나타날 속성의 선택에 참여할 수 있다. 관계타입에 의해 생성된 프로그램은 개체타입에 의해 생성된 프로그램에 비해 훨씬 복잡할 것이다. 왜냐하면, 관계타입에 참여하는 해당 개체들을 모두 액세스해야 하기 때문이다.

3.1.2 업무 프로그램의 링크 생성

위의 세 가지 규칙에 의해서 E-R 모델은 일정한 수의 응용시스템 프로그램으로 변환될 수 있다. 다음 과제는 프로그램들을 서로 연결하는 일이다. 즉, 응용시스템은 개체관계도에 있는 관계타입에 속한 "관계"들을 통해 네트워크로 연결된 개체들 사이의 연결관계를 유지하고 관리할 수 있어야 한다.

<규칙 4>: 개체/관계 타입에서 생성된 업무 프로그램에 대하여 그 개체/관계 타입과 연결된 다른 개체/관계 타입에서 생성된 프로그램으로의 연결을

위한 링크를 생성한다.

<규칙 4>는 개체관계도에 있는 네트워크를 통해 한 개체에서 다른 타입의 개체로의 운행 요구를 반영하기 위한 것이다. 만약 이와 같은 링크가 없다면 이용자는 어떻게 될까? 이 경우 개체/관계 타입의 네트워크를 운행하는 것은 전적으로 수작업에 의존하게 된다.

[그림 7]의 개체관계도에서 각각의 개체/관계 타입이 앞의 세 규칙에 의해 프로그램으로 생성되고 동시에 프로그램의 목록이 생성되었다고 가정한다. 어떤 고객이 자신이 발주한 주문의 내역을 알아보고자 한다. 이를 위해 그는 먼저 프로그램 "고객"에서 고객 ID를 입력하여 해당 고객을 조회한 다음, 그 ID를 가지고 "발주"에서 그 고객의 주문 ID를 취한 다음 프로그램 "주문"을 기동시켜 주문 ID 입력박스에 입력하여 원하는 정보를 얻게 된다. 따라서 한 프로그램에서 다른 프로그램으로 운행하는 것은 전적으로 수작업으로 행하지 않으면 안된다. <규칙 4>를 [그림 2]의 개체타입 "주문"에 적용하면 이 개체타입으로부터 생성된 프로그램 "주문"에게는 발주, 사용, 의뢰, 지시, 결과의 5 개 링크가 생성된다.

3.1.3 프로그램의 연계를 위한 데이터의 보존

이용자가 응용시스템에서 E-R 모델의 네트워크를 운행할 때 이 자동생성기에 의해 생성된 시스템에 의하면 개체관계도에서 새로운 개체와 관계타입으로 이동할 때마다 새로운 화면이 사용된다. 특별한 장치가 없는 한 한 화면(즉, 프로그램)에서 다른 화면으로 전환되면 이전 화면에서 다룬 데이터는 사라진다. 다음의 <규칙 5>가 이 문제를 해결한다.

<규칙 5>: 한 개체/관계 타입에서 생성된 업무 프로그램에서 다른 프로그램으로 연결될 때 개체타입의 증명속성 값이 자동적으로 다음 업무 프로그램에 입력되도록 한다.

<규칙 5>는 세션의 개념과 연계되어 고려되어야 한다. <규칙 5>는 현재의 세션이 종료되기 전에는 개체타입의 증명속성에 한하여 그 데이터가 이전 화면에서 이후 화면으로 자동적으로 전달되도록 보장한다. 이를 위해 응용시스템은 특정 데이터를 지속적으로 유지할 수 있는 장치를 제공하여야 한다.

<규칙 5>는 필수적인 규칙이라기보다 이용자의 편의를 위한 장치이다. 만약 <규칙 5>가 없다면 응용시스템에서 한 개체/관계타입 프로그램에서 다른 프로그램으로 전환할 때 증명속성의 값을 수작업으로 전달해야 한다.

3.1.4 기수성과 프로그램

개체관계도에 있는 기수성은 자동생성기에 의해 생성되는 업무 프로그램의 형태에 결정적 영향을 준다. 예를 들어, [그림 4]에서는 한 교수가 강의하는 모든 과목이 나열되어 있다. 개체타입 "교수"와 "과목"은 일-대-다수 기수성을 갖는다. 다시 말해, 한 교수는 여러 과목을 강의하고 한 과목은 오직 한 교수에 의해서 강의된다는 사실을 나타낸다. 이 경우에 <규칙 2>에 의해 생성되는 프로그램은 화면에 한 명의 교수에 대하여 복수의 "과목" 데이터를 나타내 주어야 한다. [그림 4]의 경우 2 개의 과목이 나열되어 있다. 일-대-다수 기수성의 경우 "다수"가 몇 개인지를 지정하지는 않기 때문에 응용프로그램은 불특정 다수의 출현에 대비하지 않으면 안된다.

3.2 자동생성 기능의 확장

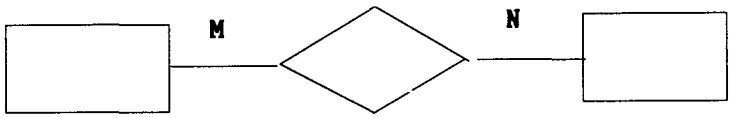
앞에서 제시한 자동생성 규칙을 통하여 응용시스템의 기본적인 구성요소들이 모두 생성될 수 있다. 그리고 이렇게 생성된 응용시스템은 개체관계도의 어느 개체타입에서 출발하여도 연결된 네트워크를 따라 다른 어느 개체타입의 개체에 대한 조회, 삭제, 수정 및 등록의 작업을 수행할 수 있다. 이러한 관점에서 이 생성 규칙은 완전하다고 하겠다. 동시에 이렇게 생성된 프로그램은 매우 단순하고 효율적이다. 다시 말해, 한 개체/관계 타입을 오직 한 프로그램에서 처리하기 때문에 그 논리는 정형화될 수 있다.

이러한 응용시스템 자동생성 기능을 보다 고도화 된 형태의 프로그램 생성을 위하여 확장하고자 하는 요구가 존재한다. 예를 들어, [그림 2]의 개체관계도에서 어떤 고객이 그가 발주한 주문 각각의 내역을 조회하거나 수정하고자 한다고 가정하자. 지금까지 소개한 5개의 규칙에 의해서 생성된 프로그램들을 이용한다면 그는 "고객" -> "발주" -> "주문"의 3 프로그램을 이용해서 그 정보처리 요구를 충족시킬 수 있다. 그러나 이제 그는 이 요구를 하나의 프로그램에 의해 처리함으로써 편의성을 높이고자 한다.

3.2.1 관계타입 중심의 확장

개체관계도를 보면 개체타입들은 관계타입을 통해 연결되어 있기 때문에 한 관계타입과 그 타입에 연결된 개체타입의 세트는 E-R 모델의 기본 단위가 된다. 이러한 개체/관계타입 단위는 자동생성의 대상이 될 수 있다. E-R 모델에서 관계타입은 참여하는 개체타입의 수에 따라 양방향, 3-방향... 등으로 구분된다.

양방향 관계타입: [그림 5]는 다수-대-다수 기수성을 가진 양방향 관계타입이다. 이 기수성이 의미하는 바는 A 개체 하나에 다수의 B 개체가 그리고 B 개체 하나에 다수의 A 개체가 연관될 수 있음을 나타낸다. 이 관계타입으로부터 자동생성기는 2 개의 프로그램을 생성할 수 있다.



[그림 5] 다수-대-다수 관계타입

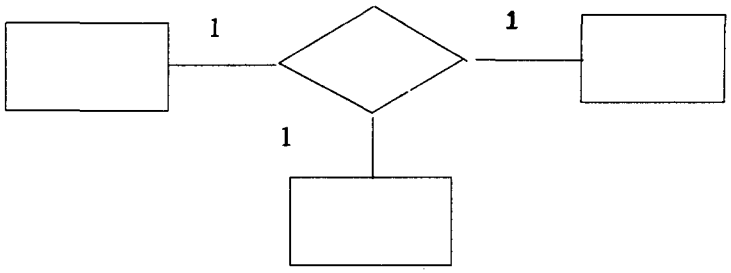
<확장규칙 1>: 양방향 관계타입의 A 개체에서 연관된 B 개체들에 대한 조회, 추가, 변경 및 삭제 작업 그리고 B 개체에서 연관된 A 개체들에 대한 조회, 추가, 변경 및 삭제 작업을 위한 업무 프로그램을 생성한다.

이 규칙은 한 A 개체에서 C 관계를 통해서 B 개체로 항해하는 비교적 복잡한 상황이다. 이를 위한 처리는 헤드(Head)와 바디(Body)가 모두 존재하는 화면의 형태를 필요로 한다. 이때 헤드에는 A 개체 한 개가 나타나고, 바디에는 복수의 B 개체들이 나타난다. 반면에 역으로 즉, B 개체를 중심으로 동일한 내용의 처리가 이루어진다.

양방향 일-대-다수 관계타입에 의해 정의되는 프로그램들은 기본적으로 앞의 경우와 유사하다. A 개체에서 연관된 B 개체들에 대한 작업은 앞의 경우와 동일하다. 그러나 B 개체에서 연관된 A 개체에 대한 작업은 단수 개체를 표현하는 형태가 된다.

일-대-일 양방향 관계타입과 참여하는 두 개체타입을 통해서 정의되는 프로그램들 역시 기본적으로 앞의 경우와 유사하다. 단지, A 개체에서 연관된 B 개체들에 대한 작업 그리고 그 역의 경우 작업은 역시 한 화면에 A와 B 개체가 각각 하나씩 나타나는 형태가 된다.

3-방향 관계타입: [그림 6]과 같은 3-방향 관계타입의 경우에는 <확장규칙 2>가 적용된다.



[그림 6] 3-방향 관계타입

<확장규칙 2>: 3-방향 관계타입에서 A 개체에서 연관된 B, C 개체들에 대하여, B 개체에서 연관된 A, C 개체들에 대하여 그리고 C 개체에서 연관된 A, B 개체에 대한 조회, 추가, 변경 및 삭제 작업을 수행하는 업무 프로그램을 생성한다.

E-R 모델에서는 4-방향 이상의 관계타입의 경우에도 이와 비슷한 확장규칙이 적용될 수 있다. 그러나 4-방향 관계타입은 존재할 수는 있지만 그 가능성은 그다지 높지 않기 때문에 더 이상 논의하지 아니한다.

3.2.2 개체타입 중심의 확장

<확장규칙 1>과 <확장규칙 2>에 의해 생성된 업무 프로그램을 가지고 이용자는 양방향 혹은 3-방향 개체-관계-개체의 연결된 정보의 요구를 충족시킬 수 있다. 그러나 이용자의 요구는 조금 더 확장될 수 있다. <확장규칙 3>을 설명하기 위해 근접개체타입을 먼저 정의한다. "근접(近接)개체타입"이란 한 개체타입에서 오직 하나의 관계타입을 통해서 연결된 개체타입을 말한다. <부록 1>의 E-R 모델에서 개체타입 주문의 근접개체타입은 고객, 꽃패키지, 화원 그리고 배달원의 4 개이다.

이용자는 한 개체를 시발점으로 해서 그 개체와 관계가 있는 다른 개체들을 (별도의 화면을 연속적으로 전환하여 조회하는 대신) 하나의 화면에서 이를 모두 조회하거나 수정하고자 하는 요구를 가질 수 있다. 이러한 요구는 위의 두 확장규칙에 의해 생성되는 업무 프로그램을 시발점 개체를 중심으로 기능적으로 통합함으로써 충족될 수 있다.

<확장규칙 3>: 하나의 개체타입에서 그 개체타입의 모든 근접개체타입에 대한 조회, 추가, 변경 및 삭제 작업을 수행하는 프로그램을 생성한다.

자동생성기는 이 확장규칙을 적용하는 과정에서 <확장규칙 3>에 의해 생성될 프로그램이 <확장규칙 1>과 2에 의해서 이미 생성된 업무 프로그램과 동일한 것인가 여부를 먼저 확인하여 만약 그렇다면 이를 중복적으로 생성하지 않는다.

<부록 1>의 개체관계도에서 <확장규칙 3>을 개체타입 "주문"에 적용하면 한 화면에서 이 개체관계도 전체가 포함하고 있는 정보를 한 번에 조회하는 프로그램을 생성할 수 있음을 알 수 있다.

3.3 데이터베이스의 생성

E-R 모델을 이용하여 관계형 데이터베이스를 설계하는 과정은 잘 알려져 있다. 다음의 데이터베이스 규칙에 의해 생성되는 데이터베이스는 (적절한 정규화 과정을 거쳐) 관계형 데이터베이스로 설계되는 것으로 가정한다. 기본적으로 E-R 모델에 있는 각각의 개체/관계 타입은 데이터베이스 테이블로 생성된다.

<데이터베이스 규칙>: E-R 모델에 있는 각각의 개체/관계 타입은 관계형 데이터베이스의 테이블로 생성되며 개체/관계 타입에 속한 속성들은 각각 칼럼으로 정의된다.

3.4 인터페이스의 설계

앞서 5 개의 규칙과 3 개의 확장규칙에 의해 생성된 시스템은 정형적(定型的) 즉, 스테레오타입(Stereotype) 시스템이다. 다시 말해, 이 용용시스템은 특정한 이용자를 위해 조정되지 않은 상태에 있다. 따라서 이용자들의 특성과 요구에 기초하여 정형적 응용시스템을 이용자 시스템으로 구체화하는 일이 수행되어야 한다.

이 절에서 사용하는 "이용자"라는 용어는 최종적으로 완성된 응용시스템을 사용하게 될 사람들을 의미한다. 그러나 자동생성 과정의 "이용자 입력" 단계에서 여러 가지 시스템 설계 결정을 내리고 그 내용을 입력하여 정형적 시스템을 수정하는 주체로서의 "이용자"는 최종 이용자의 입장을 대변하여 자동생성 과정을 수행하는 응용시스템 개발자 혹은 설계자를 지칭한다. 물론 응용시스템의 이용자와 개발자는 동일인일 수 있다.

앞서 E-R 모델을 이용한 응용시스템의 자동생성 과정에서 고려해야 할 중요한 원칙으로서 이용자의 특성과 요구에 맞추어 응용시스템이 생성되어야 함을

강조하였다. 이러한 원칙을 실현하기 위해서는 지금까지 제시한 규칙들에 의하여 E-R 모델로부터 생성된 정형적 시스템을 이용자 입력 및 수정 과정을 통하여 응용시스템의 설계가 최종적으로 확정될 수 있도록 하는 "인터페이스(Interface) 설계" 과정이 불가피해진다. 그러나 이 과정이 과도하게 복잡하면 자동생성의 효과가 떨어지게 되기 때문에 자동생성 과정에서 허용할 수 있는 이용자 개입의 적절한 수준을 결정하는 것이 중요하다.

이용자는 다음과 같은 관점에서 응용시스템의 설계를 결정할 수 있다. 첫째, 이용자는 생성된 정형적 업무프로그램들 중에서 그가 필요로 하는 것들을 선택할 수 있다. 둘째, 이용자는 실제로 화면에 나타나게 될 속성을 결정할 수 있다. 셋째, 이용자는 화면에 나타나는 속성들의 위치를 결정할 수 있다. 마지막으로 넷째, 이용자는 화면에 나타나는 (다른 프로그램으로의) 링크를 선택할 수 있다.

먼저 개발자는 응용시스템의 이용자 그룹을 정의하고 이를 자동생성기에 등록한다. 이용자는 이름, 사용 아이콘 등 여러 가지 특성을 갖는다. 보통 한 응용시스템에는 복수의 이용자 그룹이 존재할 가능성이 높다. 개발자는 등록된 각각의 이용자 그룹에 대하여 다음의 4 규칙을 적용하여 정형적 시스템을 구체적으로 실현한다. 자동생성기는 이용자 그룹의 목록을 가지고 응용시스템의 상위 메뉴를 생성한다.

<인터페이스 설계 1>: 이용자는 정형적 시스템이 제공하는 업무 프로그램들 중에서 그가 필요로 하는 것들을 선택한다.

자동생성기는 <인터페이스 설계 1>을 적용하기 위해서 앞에서 자동생성기에 의해 생성된 모든 업무프로그램들의 목록을 제시한다. 이 목록에 제시된 (정형적) 업무프로그램들은 응용시스템의 최대한의 범위를 정의한다. 업무프로그램의 선택에는 이용자의 업무 수행에 필요한 실용성이 가장 중요한 기준이 되지만 이용자 그룹에 대하여 일정한 범위의 이용 권한을 부여하는 접근통제 차원의 고려도 포함된다.

시스템 개발자는 제시된 목록으로부터 응용시스템의 이용자가 필요로 하는 업무기능을 지원할 수 있는 업무프로그램들을 선택한다. 자동생성기는 이 입력 데이터를 기초로 하여 응용시스템 상에 해당 이용자 그룹을 위한 하위 메뉴를 생성한다.

<인터페이스 설계 2>: 이용자는 업무 프로그램에 포함된 속성들 중에서 그가 필요로 하는 것들을 선택한다.

시스템 개발자는 <인터페이스 설계 1>에 의해 선택된 업무 프로그램을 검토하여 각각의 프로그램에 포함될 속성들을 선택한다. 이를 위해 자동생성기는 이용자에게 업무프로그램 별로 그 프로그램에서 다루게 될 속성들의 목록을 제시한다. 시스템 개발자는 (특정 이용자 그룹을 대신하여) 제시된 속성 목록으로부터 그 이용자가 사용하기를 원하는 속성만을 선택하게 된다.

<인터페이스 설계 1, 2>에 의해 응용시스템이 제공할 수 있는 기본적인 정보의 내용과 업무 처리의 범위는 이용자 그룹별로 최적화 된다. 자동생성기에 의해 생성된 업무 프로그램의 풀(Pool)에는 많은 중복이 존재한다. 기본적으로 앞서 제시한 3 개의 확장규칙들은 이용자 편의를 위해 단위 프로그램들의 기능을 통합한 업무 프로그램을 생성하기 때문이다. 따라서 이용자는 많은 대안 중에서 최적의 선택을 할 수 있도록 시도할 것이다. 이러한 선택에는 이용자의 개인적인 선호가 중요한 역할을 할 수도 있기 때문에 시스템 개발자는 이용자 스스로 이 결정을 내리고 또 직접 입력하도록 할 수도 있다.

<인터페이스 설계 3>: 이용자는 업무프로그램의 화면 설계를 선택한다.

자동생성기가 제안하는 화면의 설계는 임시적인 것이다. 이용자는 이 단계에서 화면의 설계를 최종적으로 결정할 수 있다. 이를 위해 자동생성기는 업무프로그램별로 선택된 속성들과 미리 정해진 규칙을 이용하여 생성된 정형적 화면설계를 이용자에게 제시한다. 이용자는 화면에 나타날 속성 항목의 위치, 폰트의 크기, 색상 등 화면 표시 형식에 대해서 최종적인 결정을 내린다.

<인터페이스 설계 4>: 이용자는 업무프로그램의 링크(Link)를 선택한다.

마지막으로 이용자는 한 업무프로그램 화면에서 다른 화면으로 이동을 통제하는 링크 즉, 통제프로그램에 대한 최종적인 결정을 내린다. 자동생성기는 앞서 E-R 모델로부터 <규칙 4>를 적용하여 개체/관계 타입 사이에 존재하는 연결선을 모두 정형적 시스템 상에 업무프로그램의 링크로서 생성해 놓았다. 링크를 통해 서로 연결된 일련의 업무 프로그램들은 이용자의 업무 프로세스를 나타내기 때문에 링크의 선택은 이용자 업무 프로세스의 최적화 된 설계에 중요한 요인이 된다.

4. 사례 소개

4.1 대상시스템의 E-R 모델

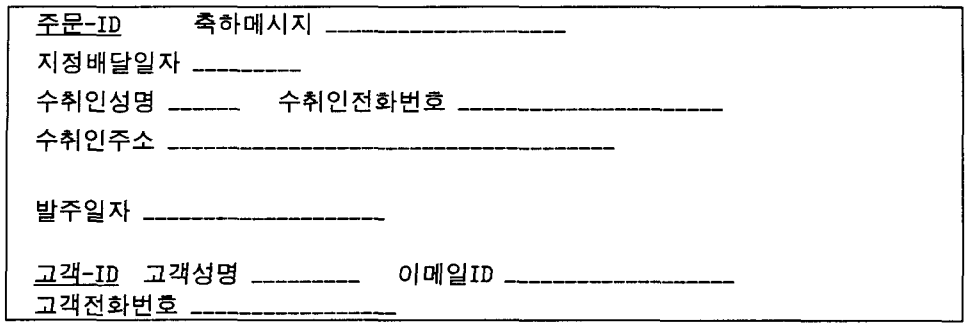
그러면 E-R 모델을 통하여 응용시스템을 설계하고 생성하는 시스템의 사례를 검토하기로 한다. 자동생성 과정를 시작하기 위한 준비물은 기본적으로 대상 시스템에 대한 개체관계도와 속성의 목록이다. 이 사례에서 다룰 대상은 가상의 꽃배달 시스템으로서 자동생성기가 어떻게 작동하는지 보여 주기 위하여 비교적 단순한 시스템을 가정하였다. <부록 1>에 사례로 사용될 개체관계도가 주어져 있다. <부록 2>는 개체타입과 관계타입들의 속성 목록을 보여 주고 있다. 속성들 중에 밑줄이 그어진 것은 증명속성 즉, 그 타입에서 각각의 개체 혹은 관계를 유일하게 증명하는 속성을 의미한다.

4.2 정형적 시스템의 생성

그러면 이제 본격적인 응용시스템의 자동생성 단계로 이행하여 응용시스템의 정형적 모델이 생성된다. 여기서 생성된 정형적 시스템은 다음 단계에서 이용자를 위한 인터페이스 설계를 거쳐 완전한 시스템으로 구축된다.

생성 규칙 1, 2 및 3의 적용: <규칙 1>과 함께 <규칙 2> 그리고 <규칙 3>을 적용하면 <부록 3>과 같이 도합 40개의 정형적 프로그램이 생성된다. 개체와 관계타입 10개에 대하여 <규칙 1>이 적용되면 각각의 타입은 조회, 추가, 삭제 그리고 수정 프로그램으로 세분화된다. (편의상 등록, 삭제, 수정 그리고 조회 업무 프로그램의 이름을 A, D, M 그리고 I 부호를 첨가하여 부여한다.)

<규칙 3>에 의해서 생성되는 관계타입 프로그램은 해당 관계타입과 연관된 개체타입에 대한 조회를 제공한다. 예컨대, 프로그램 "발주"는 [그림 7]과 같은 내용의 화면을 보여 줄 것이다. 관계타입 "발주"의 속성인 주문-ID, 발주일자 그리고 고객-ID는 수정, 삭제, 추가의 대상이 되지만 이 관계타입과 연관된 개체타입의 속성들은 오직 "조회" 목적으로만 제시된다.



[그림 7] 개체/관계 타입으로부터 생성되는 정형적 프로그램

규칙 4의 적용: <규칙 4>를 적용하면 앞서 생성된 40개의 업무 프로그램들 각각에는 그 프로그램으로부터 연결될 다른 업무 프로그램의 링크들의 세트가 생성된다. 다시 말해, 이용자는 40개의 프로그램 각각에서 이 링크를 클릭하여 다른 프로그램으로 직접 전환할 수 있다. 예를 들어, **주문-I** 프로그램은 다음과 같이 23 개의 링크를 갖는다. (업무 프로그램과 구분하기 위하여 { } 부호를 사용하여 링크를 표시한다.)

{주문-A}, {주문-D}, {주문-M}, {발주-A}, {발주-D}, {발주-M}, {발주-I},
{사용-A}, {사용-D}, {사용-M}, {사용-I}, {의뢰-A}, {의뢰-D}, {의뢰-M},
{의뢰-I}, {지시-A}, {지시-D}, {지시-M}, {지시-I},
{결과-A}, {결과-D}, {결과-M}, {결과-I}

이렇게 많은 링크가 생성되는 이유는 <부록 1>의 E-R 모델에서 주문 개체타입에 연결된 관계타입 5 개에 각각 4 개씩의 프로그램이 생성되었기 때문이다. **주문-I** 프로그램은 {주문-A}, {주문-D}, {주문-M} 링크를 갖는데 그 링크들은 어떤 주문에 대해 조회한 다음 그 주문을 삭제 혹은 수정할 수 있도록 하기 위해 사용될 수 있다.

규칙 5의 적용: 다음으로 자동생성기는 하나의 세션에서 화면 전환시 이전 화면의 데이터값이 계속 유지되도록 지속적 데이터를 생성한다. <규칙 5>를 적용하면 <주문-ID, 고객-ID, 회원-ID, 배달원-ID, 꽃패키지-ID>의 증명속성 값들이 한 화면에서 다른 화면 프로그램으로 전환될 때에 지속적으로 유지될 수 있는 장치를 생성한다.

확장 규칙 1, 2 및 3의 적용: <확장규칙 1>을 적용하면 <부록 5>와 같이 도합 8개의 확장된 업무프로그램들이 생성된다. 여기서 각각의 프로그램에서 선두에 표시된 것이 기동(起動) 개체타입이다. <확장규칙 2>를 적용하면 <부록

5>와 같이 3 개의 3-방향 관계타입에 대한 프로그램들에 대해서 다시 4 개씩의 도합 12 개의 업무 프로그램이 생성된다. <확장규칙 3>을 적용하면 주문, 배달원 그리고 화원 각각에 대하여 조회, 추가, 삭제 및 수정 등 4 개씩 도합 12 개의 확장된 업무 프로그램들이 생성된다. <부록 5>의 [ ] 괄호 속에 있는 것이 하나의 확장된 업무프로그램에 참여하는 개체/관계타입이다. [ ] 속에서 선두에 표시된 것이 기동 개체타입이다. 예컨대, 첫 번째 확장 프로그램에서는 한 "주문" 개체를 중심으로 관련된 모든 개체들을 처리하는 기능을 제공한다.

데이터베이스 규칙의 적용: <데이터베이스 규칙>을 적용하면 <부록 4>와 같이 도합 10개의 관계형 데이터베이스 테이블이 생성된다. 속성 중에서 밑줄이 쳐진 항목은 주키를 가리킨다.

4.3 인터페이스 설계

먼저 <인터페이스 설계 1>에서는 다음과 같은 4 종류의 이용자 그룹이 등록된다. < > 속에는 응용시스템 상위 메뉴의 텍스트박스(Text-box)에 나타날 명칭을 명시한다.

고객: <고객서비스>
고객서비스매니저: <고객주문관리>
배달원: <배달관리>
화원경영자: <화원괸리>

이용자 그룹이 등록되면 다음 단계로서, 이용자 그룹별로 그 그룹의 이용자들이 사용할 업무프로그램을 선택하는 작업이 수행된다. 시스템 개발자는 고객에 대하여 다음과 같은 업무 프로그램을 선택한다.

고객-I, 고객-A, 고객-D, 고객-M, 주문-I, 주문-A, 주문-D, 주문-M,
발주-I, 발주-A, 발주-D, 발주-M, 고객->발주->주문-I, 주문->사용->꽃패키지-I,
주문->결과->배달원-I

일견, 이 프로그램들을 모두 사용할 수 있다면 고객은 이 응용시스템으로부터 상당한 수준의 정보서비스를 받을 수 있음을 예상할 수 있다. 자동생성기는 위의 정보를 가지고 상위 그리고 하위 메뉴를 생성한다.

고객서비스	고객주문관리	배달관리	화원관리

위의 상위 메뉴에서 **고객서비스**를 선택하면 다음과 같은 하위 메뉴가 팝업된다.

고객-I, 고객-A, 고객-D, 고객-M,
주문-I, 주문-A, 주문-D, 주문-M,
발주-I, 발주-A, 발주-D, 발주-M,
고객->발주->주문-I
주문->사용->꽃패키지-I
주문->결과->배달원-I

메뉴의 생성과 이용자의 요구에 기초한 업무프로그램의 선택이 완료되면 시스템 개발자는 <인터페이스 설계 2> 즉, 속성 선택 작업으로 넘어간다. 자동생성기는 이용자에게 업무프로그램 별로 그 프로그램에서 다루게 될 속성들의 목록을 제시한다. 이용자는 제시된 속성 목록으로부터 원하는 속성을 선택하게 된다. 예컨대, **주문->결과->배달원-I** 프로그램의 속성들은 다음과 같다.

주문-ID, 축하메시지, 지정배달일자, 수취인성명, 수취인주소, 수취인전화, 배달원-ID, 배달일자, 배달결과, 배달원성명

이용자인 "고객"은 위의 속성 목록에서 불필요하다고 생각하는 **축하메시지, 수취인주소, 수취인전화, 배달원성명**의 5 속성들을 삭제하고, 최종적으로 **주문->결과->배달원-I** 프로그램의 속성들을 다음과 같이 결정한다.

주문-ID, 지정배달일자, 수취인성명, 배달원-ID, 배달일자, 배달결과

<인터페이스 설계 3>에서 시스템 개발자(혹은 이용자)는 업무 프로그램의 화면 설계를 최종적으로 설계한다. 개발자는 계속해서 자동생성기에 의해 제시된 화면 설계를 검토하고 이를 수정한다. 예를 들면, 방금 전에 속성을 선택한 **주문->결과->배달원-I** 프로그램의 화면 설계는 다음과 같이 수정될 수 있다.

주문-ID : _____ 지정배달일자 _____
수취인성명 _____ 배달원-ID _____
배달일자 _____ 배달결과 _____

마지막으로 <인터페이스 설계4>에서는 그가 선택한 업무프로그램과 연계될 링크를 취사선택한다. 예를 들어, 방금 전에 속성을 선택한 **주문->결과->배달원-I** 프로그램의 링크들 중에서 다음의 4 개만을 선택한다.

{발주-I}, {사용-I}, {의뢰-I}, {지시-I}

추가로 <확장규칙 1, 2>를 사용하여 생성된 다음의 4 개 업무 프로그램 링크들 중에서 **{주문-사용-꽃패키지-I}** 링크만을 선택한다.

{주문->발주->고객-I}, {주문->사용->꽃패키지-I}, {주문->의뢰->회원-I}, {주문->지시->배달원/화원-I}

<확장규칙 3>을 적용하여 생성된 업무프로그램으로의 링크는 선택하지 아니한다. 이렇게 하여 최종적으로 다음의 5 개 링크만을 선택한다.

{발주-I}, {사용-I}, {의뢰-I}, {지시-I}, {주문-사용-꽃패키지-I}

이렇게 선택된 링크가 추가된 앞의 **주문->결과->배달원-I** 프로그램의 화면 설계는 [그림 8]과 같이 수정된다. 화면의 하단에 최종적으로 이용자가 선택한 링크가 진한 고딕체로 나타나 있다. 상단에 이 업무프로그램의 이름이 주어진다. 이렇게 하여 E-R 모델로부터 출발하여 응용시스템을 자동생성하는 과정이 모두 종료된다.

주문->결과->배달원-I

주문-ID : _____ 지정배달일자 _____
수취인성명 _____ 배달원-ID _____
배달일자 _____ 배달결과 _____

[발주-I] [사용-I] [의뢰-I] [지시-I] [주문-사용-꽃패키지]

[그림 8] 링크가 추가된 이용자 시스템의 업무 프로그램

5. 토론 및 결론

5.1 연구 내용의 요약 및 토론

지금까지 본 논문에서는 E-R 모델로부터 출발하여 응용시스템을 자동적으로 생성하는 전체적인 과정에 대해서 검토하였다. 응용시스템의 자동생성 규칙들이 제안되었고, 이러한 규칙들이 적용될 수 있는 절차를 제시하였다. 그 결과 응용시스템은 E-R 모델을 통하여 거의 자동생성에 가까운 효율성을 가지고 생성될 수 있음을 보여주었다. 아울러 생성되는 응용시스템이 이용자의 특성에 맞추어 조정될 수 있도록 "이용자 인터페이스 설계"라는 과정을 도입하였다.

또 이용자에 응용시스템이 적응할 수 있도록 하기 위하여 자동생성 과정을 기본적으로 정형적(즉, 스테레오타입) 시스템을 생성하는 단계와 결정적 이용자 입력과정을 통해서 정형적 시스템이 이용자 시스템으로 맞추어지는 2 단계의 접근 방식을 제안하였다.

이러한 자동생성의 원리가 실제로 작동할 수 있는지 그 실현 가능성을 점검하기 위하여 가상의 꽃배달 시스템을 대상으로 하여 자동생성 과정을 수행하였다. 즉, 이 대상 시스템의 E-R 모델은 이미 설계가 완료된 것으로 가정하고 여기서 출발하여 본 연구에서 제안한 규칙과 절차를 적용하여 최종적으로 작은 응용시스템을 생성하는 전체적인 과정을 실제로 따라서 가상의 응용시스템을 생성하였다.

본 연구에서는 E-R 모델의 관계타입 중에서 3-방향 관계타입까지를 확장규칙에 포함시켰다. 매우 드물기는 하지만 4-방향 이상의 관계타입도 존재할 수 있기 때문에 자동생성기가 어느 정도까지 복잡한 관계타입들을 모두 자동생성의 대상으로 할 것인가의 문제가 남는다.

이용자의 특성과 요구에 부응하기 위한 인터페이스 설계의 범위는 앞서 제안한 것보다는 훨씬 더 다양하고 복잡할 수 있다. 특히 멀티미디어 데이터의 처리, 화면 설계의 다양한 가능성 등이 추가로 고려된다면 자동생성기와 화면 에디터의 구분이 희미해 질 수도 있다.

본 연구에서 제안한 자동생성의 규칙과 원리를 실제의 소프트웨어로 구현하는 데에는 포함하여 여러 가지 대안이 존재할 것이다. 이와 관련하여서는 어떤 특정한 도구를 가정하지 않고 있다. 자동생성된 업무 프로그램들이 작동할 플랫폼, 인터넷 웹 시스템과 기존의 클라이언트-서버 아키텍처, 미들웨어 등 다양한 차원의 문제들이 존재할 것이다.

5.2 본 연구의 의의

본 연구는 응용시스템의 자동생성이라는 주제를 다루었고, 소프트웨어의 생성 과정의 효율성과 생산성의 문제를 다루는 소프트웨어 엔지니어링 분야에서 한 가지 새로운 시각을 제시하였다는데 그 의의를 찾을 수 있다. 지금까지 제시된 자동생성 과정이 실제로 구현되고 적용된다면 시스템 개발자는 상당한 수준의 소프트웨어 생산성 향상을 얻을 수 있을 것으로 기대된다. 기본적으로 E-R 모델만이 설계된다면 그 다음 단계는 체계적이고 자동적으로 진행될 수 있기 때문이다.

마찬가지로, 개체관계 모델에 대한 연구에도 다소 기여한 것으로 생각된다. 지금까지 개체관계 모델은 데이터베이스의 모델로서 인식이 되어 왔고, 부분적으로 일반적인 시스템 분석 및 설계의 도구로서 그 역할이 인식되어 왔고 따라서 E-R 모델에 대한 연구도 이러한 분야에 집중되어 왔다. 본 연구는 앞으로 더 활발한 연구가 뒷받침되어야 함에도 불구하고 이러한 E-R 모델 분야의 연구에 새로운 방향을 제시하는데 의의가 있다고 생각된다.

5.3 본 연구의 제약점과 앞으로의 연구 방향

본 연구는 몇 가지 제약 조건을 가지고 있다. 첫째, 본 연구에서 제시한 응용시스템의 자동생성 과정은 기본적으로 "잘 설계된" E-R 모델을 전제로 하고 있다. 다시 말해, 응용시스템이 대상으로 하는 시스템을 E-R 모델을 가지고 잘 표현할 수 있다면 그 다음 단계는 자동생성기의 도움을 받아 매우 효율적으로 진행할 수 있다는 것이다. 그러나 역으로 E-R 모델이 대상을 적절히 반영하기 못하고 있다면 결과적으로 생성되는 응용시스템 역시 이용자의 정보와 업무 요구를 충분히 소화해 내지 못할 것이다. 아울러 아무리 잘 설계된 E-R 모델이라고 할지라도 이것에 기반을 두고 생성되는 응용시스템은 그 E-R 모델이 표현하고 있는 수준 이상으로 만들어 질 수는 없다. 이 점은 이 접근 방식이 가진 근본적인 제약점이다.

둘째, 본 연구에서 제안한 방법에 의해 이용자의 모든 정보요구가 충족될 수 있다고 생각하는 것은 성급한 생각이다. 어느 응용시스템이든지 자동생성 도구가 접근할 수 없는 영역이 언제나 존재하기 때문이다. 따라서 E-R 모델을 통해 논리적으로 강화된 자동생성 기능은 언제나 자동생성된 프로그램의 수작업 수정이라는 과정을 통해 부분적으로 보완될 수 있는 가능성을 염두에 두어야 한다. 이런 관점에서 자동생성 되는 응용시스템의 융통성을 제고시키는 방향의 연구가

필요하다. 생성된 응용시스템이 주변 환경의 변화에 융통성 있게 적응해 갈 수 있는 체계와 장치를 자동생성 되는 시스템과 프로그램이 어떻게 갖도록 확보하는가 하는 것은 모든 CASE 도구들의 과제이다.

셋째, 본 연구에서 제시한 것은 응용시스템의 자동생성기 그 자체가 아니고 자동생성기의 원리와 절차이다. 실제에서 이러한 제안한 아이디어를 실현하는 것은 앞으로의 연구 과제가 될 것이다.

참고문헌

[1] 정일주, 데이터베이스, 모델, 언어 및 설계, 시그마프레스, 1999.

[2] Batra, Dinesh, Hoffer, Jeffrey A., "An ER based methodology for modeling user views and detecting derived relationships," Journal of Database Management, Winter 1994, 5(1): 3-16.

[3] Chen, Peter P. "The Entity-Relationship Model: Towards a Unified View of Data," ACM Transaction on Database Systems, Vol. 1, No. 1, March 1976, pp. 9-37.

[4] Chen, Peter P. The Entity-Relationship Approach to Logical Database Design, Q.E.D. Information Sciences, Inc., 1977.

[5] Chen, Peter P. (Ed.) Entity-Relationship Approach to Systems Analysis and Design. North Holland, October, 1983.

[6] Chen, Peter P., Ilchoo Chung & Dennis Perry. A Logical Database Design Framework, NBS Publication, NIST, 1982.

[7] Francett, Barbara, "Companies mind their business with integrated modeling tools," Software Magazine, Dec 1994, 14(12): 57-63

[8] Gane, Chris. Computer-Aided Software Engineering, The Methodologies, the products, and the future, Prentice-Hall, 1990.

[9] McFadden Fred R. and Jeffrey A. Hoffer, Modern Database Management. The Benjamin Cummings Publishing Company, 1994.

[10] McChesney, I., Glass, D., Hughes, J. G., "CASE Tool Support for Requirements Capture and Analysis," Microprocessing & Microprogramming, Aug 1990, 30(1-5): 281-288.

[11] Shelly, Gary B., Thomas J. Cashman, Judy Adamsky and Joseph J. Adamski. Systems Analysis and Design (Second Ed.), boyd & fraser publishing company, 1995.

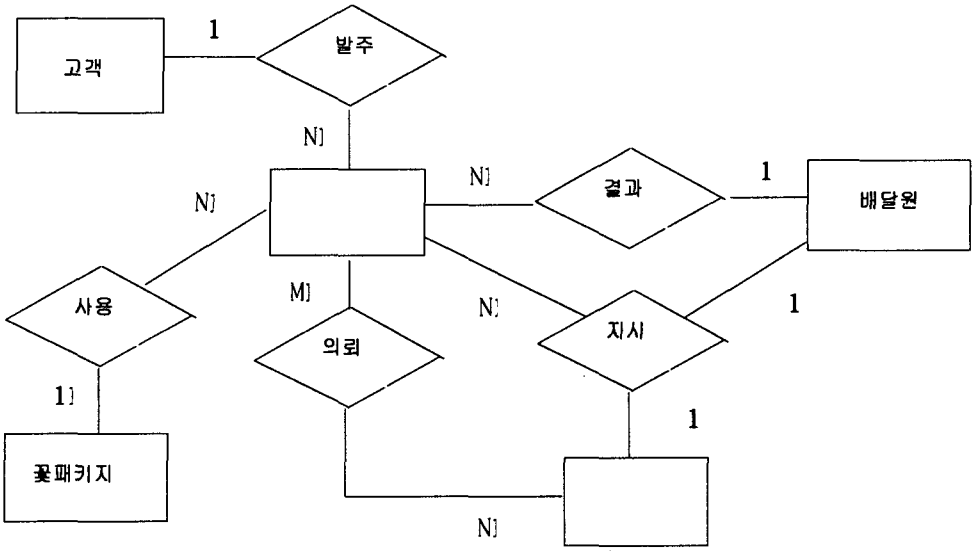
[12] Siau, K.L., Chan, H.C., and Tan, K.P. "A CASE too for conceptual database design," Information & Software Technology, Dec 1992, 34(12): 779-786.

[13] Stoughton, Alan M., "ERwin streamlines GUI and adds versatile features," InfoWorld, July 1997, 19(27): 96-98.

[14] Tate, G., Verner, J. and Ross, J. "CASE: a testbed for modeling, measurement and management, Communications of the ACM, v35 n4, April 1992, pp 65-72.

[15] Whitten, Jeffrey L. and Lonnie D. Bentley. Systems Analysis and Design Methods (4th Edition), McGraw-Hill, 1998.

부 록 1: 꽃배달 시스템의 E-R 모델



부 록 2: 개체/관계타입의 속성 목록

개체타입	속성	값타입
주문	주문-ID	문자
	축하메시지	문자
	지정배달일자	일자
	수취인성명	문자
	수취인주소	문자
	수취인전화	전화번호
고객	고객-ID	문자
	고객성명	문자
	이메일ID	문자
	고객전화번호	전화번호
꽃패키지	꽃패키지-ID	문자
	패키지이름	문자
	가격	숫자
화원	화원-ID	문자
	화원이름	문자
배달원	배달원-ID	문자
	배달원성명	문자

관계타입	속성	값타입
발주	주문-ID	문자
	고객-ID	문자
	발주일자	일자
사용	주문-ID	문자
	꽃패키지-ID	문자
의뢰	주문-ID	문자
	화원-ID	문자
	의뢰일자	일자
지시	주문-ID	
	화원-ID	문자
	배달원-ID	문자
	배달지시일자	
결과	주문-ID	문자
	배달원-ID	문자
	배달결과	문자
	배달일자	일자

부 록 3: 규칙 2, 3에 의해 생성된 업무프로그램

적용규칙	E-R타입	업무 프로그램			
		조회	추가	삭제	수정
규칙 2	고객	고객-I	고객-A	고객-D	고객-M
	주문	주문-I	주문-A	주문-D	주문-M
	꽃패키지	꽃패키지-I	꽃패키지-A	꽃패키지-D	꽃패키지-M
	화원	화원-I	화원-A	화원-D	화원-M
	배달원	배달원-I	배달원-A	배달원-D	배달원-M
규칙 3	발주	발주-I	발주-A	발주-D	발주-M
	사용	사용-I	사용-A	사용-D	사용-M
	의뢰	의뢰-I	의뢰-A	의뢰-D	의뢰-M
	지시	지시-I	지시-A	지시-D	지시-M
	결과	결과-I	결과-A	결과-D	결과-M

부 록 4: 데이터베이스 규칙에 의해 생성된 스키마

고객(고객-ID, 고객성명, 이메일ID, 고객전화번호)
주문(주문-ID, 축하메시지, 지정배달일자, 수취인성명, 수취인주소, 수취인 전화)
꽃패키지(꽃패키지-ID, 패키지이름, 가격)
화원(화원-ID, 화원이름)
배달원(배달원-ID, 배달원성명)
발주(주문-ID, 고객-ID, 발주일자)
사용(주문-ID, 패키지-ID, 발주일자)
의뢰(주문-ID, 화원-ID, 발주일자)
지시(주문-ID, 화원-ID, 배달원-ID, 배달지시일자)
결과(주문-ID, 배달원-ID, 배달일자, 배달결과)

부 록 5: 확장규칙 1, 2, 3에 의해 생성된 업무프로그램

적용 규칙	중심 관계타입	업무 프로그램			
		조회(I)	추가(A)	삭제(D)	수정(M)
확장 규칙 1	발주	고객->발주->주문-I	...-A	...-D	...-M
		주문->발주->고객-I	...-A	...-D	...-M
	꽃패키지	주문->사용->꽃패키지-I	...-A	...-D	...-M
		꽃패키지->사용->주문-I	...-A	...-D	...-M
	의뢰	주문->의뢰->회원-I	...-A	...-D	...-M
		화원->의뢰->주문-I	...-A	...-D	...-M
	결과	주문->결과->배달원-I	...-A	...-D	...-M
		배달원->결과->주문-I	...-A	...-D	...-M
확장 규칙 2	지시	주문->지시->배달원/화원-I	...-A	...-D	...-M
		화원->지시->주문/배달원-I	...-A	...-D	...-M
		배달원->지시->주문/화원-I	...-A	...-D	...-M
확장 규칙 3	주문 지시	[주문->발주->고객, 주문->사용->꽃패키지, 주문->결과->배달원, 주문->의뢰->화원, 주문->지시->배달원/화원]-I	...-A	...-D	...-M
	배달원 지시	[배달원->결과->주문, 배달원->지시->주문/화원]-I	...-A	...-D	...-M
	화원 지시	[화원->의뢰->주문, 화원->지시->주문/배달원]-I	...-A	...-D	...-M

Automated Generation of Software Systems in Systems Construction

Ilchoo Chung

Abstract

This paper makes an attempt to look at the process of automatically generating an application software system based on the Entity-Relationship (E-R) model. Basically, the process consists of five steps as follows: First, the designer develops an E-R model of an real-world system. Second, the software generator automatically generates a stereo-type application system. Third, the generator produces database schema and link information between application programs. Fourth, the designer designs the user interface including menu, screen design and so on. Finally, the generator completes the process integrating all the elements of an application system. Five basic program generation rules, three extended rules and a database generation rule have been suggested. By following each rule with the generator, the designer can build an application with an extremely efficient manner compared with traditional approaches.

A case study has been included in order to show the applicability of the automated software generation process suggested in this paper. It has been demonstrated from the case study that the idea of applying an automated generator in systems development based upon the E-R model worked well.

◆ 저자소개 ◆

정 일 주 (Chung, Ilchoo)

현재 홍익대학교 상경대학 상경학부 교수로 재직중이다. 서울대학교 상과대학 학사(1968년), University of Oregon 경영학 MBA(1978년), UCLA 경영학 박사 (1983년) 학위를 취득하였다 관심분야는 데이터베이스, 소프트웨어 자동생성, 시스템 분석 및 설계 이다.

E-mail : chungic@hongik.ac.kr
Tel : 041-860-2362