

업무 양식에 근거한 객체 지향 역공학 방법론

유 천 수*, 이 희 석**

FORE: A Form-Driven Object-Oriented Reverse Engineering Methodology

Yoo, Cheon Soo, Lee, Hee Seok

Legacy applications are valuable assets that should be integrated into next generation business systems. To gain this advantage, progressive companies can reverse engineer the legacy business operations. This paper presents a form-driven object-oriented reverse engineering(FORE) methodology by the use of business forms to recover semantics of legacy applications. They retain the user-oriented contents of business and thus are easily understandable. Our form driven object-oriented reverse engineering methodology consists of five phases: form and usage analysis, form object slicing, object structure modeling, scenario design, and model integration. Knowledge about form structure and user interaction with legacy applications is used to capture the design semantics. An object model, which consists of an object structure model and scenario results from such form knowledge. The resulting object model is more likely to help reverse engineers understand and reuse legacy systems.

* 한국과학기술원 테크노경영대학원 경영공학 전공 박사과정

** 한국과학기술원 테크노경영대학원 부교수

I. 서 론

오늘날 기업의 업무는 급격히 변화하고 있다. 이러한 변화에 대응하기 위해 기업은 새로운 정보 아키텍처와 시스템을 개발하고 있다. 현재, 대부분의 조직에서는 기업의 운영을 효과적으로 지원하는 업무 응용 시스템(application system)을 이용하고 있다. 그러나 기업에서 안정적으로 운영되고 있던 기존 응용 시스템(legacy system)이 급변하는 업무 환경에 적합하지 않은 경우가 대부분이다. 이러한 기존 응용 시스템은 다음과 같은 전형적인 문제점들을 내포한다 [Aiken et al., 1993]. 첫째, 기존 응용 시스템은 어떠한 개발 문서(documentation)도 없는 경우가 대부분이다. 그 결과 기존 응용 시스템에 대한 알려진 업무 지식이 부족하고 시스템에 대한 이해가 미흡하기 때문에 성장과 발전이 어렵다. 둘째, 유지보수 하기가 복잡하며 비용이 많이 든다. 특히, 기능을 보다 관리가 용이한 구성 요소(component)들로 분할할 수 없다. 셋째, 대부분의 기존 시스템들은 1970년대에 개발된 기술을 활용하고 있다. 이러한 기술은 주장비(host) 중심인 닫힌 체계(closed system)이므로 통합이 어렵다.

기존 응용 시스템 문제점에 대한 해결 전략에는 폐기하고 무에서부터 새로이 개발하는 방안도 있다. 그러나 기존 응용 시스템은 차세대 시스템에 통합될 수 있는 가치 있는 자산으로서, 가까운 장래에 기존 시스템을 포기한다는 것은 대부분의 조직에서 적절하지 않다고 하겠다 [Kim, 1997]. 기존 시스템에 대한 단순 접근과 데이터웨어 하우스(data warehousing) 전략은 주로 새로운 시스템에서 기존 데이터를 계속 활용하고자 하는 방안이다. 마지막으로 기존 문제점에 대한 적극적 해결은 보다 유지보수가 용이한 응용 시스템으로 개선하는 일이다. 이를 시스템 이전 전략이라 하며 이 경우 기존 응용 시스템의 의미 있는 정보(semantic) 복구가 우

선이다. 즉, 새로운 기술을 활용하여 시스템을 전환하는 방법이다 [Umar, 1997]. 이러한 해결을 위해 역공학(reverse engineering)이 이용된다.

역공학의 주요 작업 중 하나는 기존 소프트웨어의 의미 있는 정보를 파악하는 것이며, 또 다른 작업은 복구된 의미 있는 정보를 보다 상위 수준의 추상화로 유도하는 것이다 [Chikofsky et al., 1990]. 역공학이라는 용어는 하드웨어 개발 분야로부터 빌려온 용어 [Waters, 1994]로서 다른 기업들의 제품이 어떻게 만들어졌는지를 발견하기 위해 일반적으로 적용되었던 기술이다. 본 논문에서는 물리적 시스템으로부터 보다 상위의 개념적 추상화를 획득하기 위한 관점에서 역공학을 정의한다. 또한, 객체 기술이 기존 및 차세대 응용 시스템 모두에 대해 일관성 있는 서비스를 제공해 주기 때문에 객체 지향 모형화 기법이 적용된다. 객체 지향 모형화의 핵심은 객체를 파악하는 것으로서 이렇게 파악된 객체를 이용하면 현 시스템을 리엔지니어링하는 것이 보다 용이할 것이다.

본 논문에서는 양식 기반 객체 지향 역공학 방법론인 FORE(Form Driven Object-Oriented Reverse Engineering Methodology)를 제안한다. 업무 양식에 관한 지식(form knowledge)을 파악하고 이를 이용하여 개념적 객체 지향 모형을 유도하는 것이 두 가지 핵심 목적이다. 양식은 최종 사용자를 위한 사용자 인터페이스 역할을 한다. 또한, 양식은 많은 조직에서 가장 널리 사용되는 공식적인 소통 수단이다 [Tsichritzis, 1982]. CODASYL의 EUFC(End User Facility Committee)에서는 양식의 중요성을 인식하고 사용자 인터페이스를 위한 핵심 수단으로서 양식 중심의 접근 방법을 권고하고 있다 [Choobineh et al., 1988 ; Choobineh et al., 1992]. 업무 응용 시스템들에 관한 대부분의 지식은 양식 자체 및 사용자/시스템 상호작용(interaction) 정보의 활용에 의해 수집될 수 있다. 따라서 FORE 방법이 양식을 수집함으로써 시작한다는 것은 매

우 자연스러운 일일 것이다. 역공학 엔지니어 (또는 설계자)는 방법론의 전 과정에 걸쳐 양식에 관한 지식을 이용하여 개념적 모형을 작성한다. 복구된 개념적 모형은 객체 모형과 시나리오로 이루어 진다.

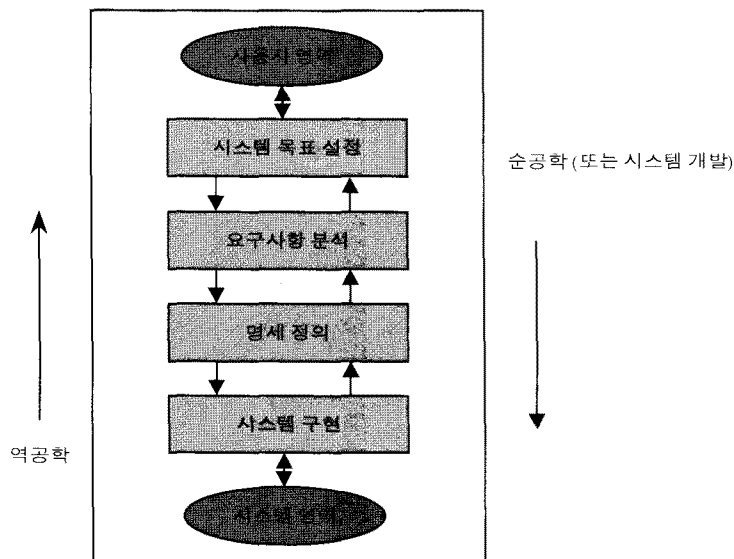
II. 역공학 기존 연구

본 장은 역공학과 관련된 기존 연구들을 요약 정리한다. 기존 연구들을 기술함에 있어 다음 네 가지 기준을 설정했다. 먼저, 소프트웨어 엔지니어링의 방향으로서 역공학과 순공학(forward engineering) 을 고려한다. 다음으로는 각 방법론에서 사용하는 입력 원천자료(input source)가 무엇인지를 기준으로 기존 연구들을 분석한다. 즉, 데이터, 원시 프로그램(source program), 양식 그리고 그 외 기존 시스템이 가지고 있는 모든 요소들을 입력 원천자료로서 활용하는 분류에 따라 방법론을 구분한다. 셋째, 방법론이 시스템에 대해 어떠한 관점을 채택하고 있느냐 하는 문제로서 시스템을 바라보는 관점을 통해

기존 연구들을 조명한다. 마지막으로, 복구된 의미 있는 정보를 모형화 하기 위해서 어떠한 개념적 모형을 사용하고 있는 지를 살펴 본다.

<그림 1> 은 역공학과 순공학의 개념을 도식화 한 것이다 [Berztiss et al., 1995]. 순공학은 시스템을 개발하거나 모형화 하기 위한 프로세스이다. 일반적으로 순공학은 개발 방법론이라고 일컬어지며 시스템 개발과정에 있어서 이전 단계의 산출물에 비해 보다 물리적인 산출물을 만들어 낸다. 역공학은 물리적인 환경으로부터 보다 상위의 개념적 요소들을 복구하기 위한 역방향 프로세스이다. 이는 구현된 시스템으로부터 시작하여 시스템 목표를 설정하는 최초의 과정에 이르는 일련의 흐름이다.

다음 <표 1>은 이전에 기술된 네 가지 기준을 기초로 기존 연구 및 방법론들을 비교 정리한 것이다. 역공학이 본 논문의 초점이므로 <표 1> 의 내용 중 순공학과에 대한 논의는 생략한다. 역공학과에 관한 관련 연구들을 <표 1>에 나타난 순서에 의거 상세히 설명하면 다음과 같다.



<그림 1> 역공학과 순공학

업무 양식에 근거한 객체 지향 역공학 방법론

<표 1> 역공학과 순공학 방법론 비교

방향	입력 원천 자료	시스템에 대한 관점	목표 모형	주요특징	방법론 명	참고 문헌	
역 공 학	관계형 DB 스키마	데이터	ER 또는 EER 모형	키 기반	Batini Navathe	Batini et al.[5] Navathe et al.[38]	
				내포 종속성 기반	Narjiwutz	Markowitz et al.[35]	
			내포 종속성과 키 종속성의 결합	Chiang	Chiang et al.[17]		
			바이너리 관계 모형	합수적 종속적 기반	Shivak	Shoval et al.[45]	
	네트워크 DB 스키마	데이터	ER 또는 EER 모형	OMT의 객체 구조 모형	- 보다 엄밀한 프로세스 - 경험을 반영	Premarlani	Premarlani et al.[40]
				사용자의 관여를 보다 많이 요구	Batini	Batini et al.[5]	
	계층형 DB 스키마	데이터	ER 또는 EER 모형	계층형 모형을 네트워크 모형으로 변환하는 선행 작업이 요구	Batini Navathe	Batini et al.[5] Navathe et al.[38]	
	원시 프로그램	프로세스	재사용 모듈(Reuse modules)	대상 기준(candidate criterion) 사용	RE2	Canfora et al.[12]	
				설계 프로그램 구조도	프로그래밍 계획 및 분할을 적용	Direct Slice	Cutillo et al.[22]
				Estelle(ISO표준명세 언어)	통신 소프트웨어의 프로토콜 역공학에 적용	Saleh	Saleh et al.[44]
기존 시스템의 모든 관련 자료	데이터 및 프로세스	SSADM(Structured Systems Analysis and Design Method)	- 특정 CASE 도구와 연계 - P나/PSA 리파지토리 기반	RECAST	Edwards et al.[23]		
			프로그램 세그먼트 모듈	지식 베이스에 기반을 둠	Cobol/SRE	Ning et al.[39]	
	객체	SSADM	- SSADM을 지원하는 CASE도구 활용 - 형식 언어와 SSADM표현 함께 활용	REDO	Lano et al.[29]		
구조적 분석 산출물 결과	객체	OMT	- 구조적 분석 결과로부터 객체모형 유도	Lee	Lee et al.[30]		
양식 구조 및 사용자 상호 작용 정보	객체	객체 지향 모형	- 응용 시스템 화면 양식 사용 - 객체 구조 및 시나리오 복구	FORE	본 연구		
순 공 학	양식	데이터	ER모형	- 업무 양식 모형 제안 - DB 설계 전문가 시스템을 개발하여 활용	Choobin도	Choobineh et al.[20] [21]	
				업무 문서 양식 및 사용자와의 대화를 이용	Kim	Kim et al.[26]	
				용어(glossary) 이용	Batini	Batini et al.[4]	
	업무 프로세스	데이터	FORMAL(Form Oriented Manipulation Language)을 이용한 양식 명세화	폭 넓은 영역의 데이터 프로세싱 활동을 전산화 할 수 있는 강력한 성능을 갖는 고 수준의 기능(facility) 제공	Shu	Shu et al.[46]	
				- BPR용 프로세스 모형과 방법 제안 - 정보처리 활동만을 다룸	EPRE(Enterprise Process Reverse Engineering)	Kim et al.[27]	
				최초의 개념적 데이터 모형화 기법	Chen	Chen[14]	
	자연어	데이터	ER 모형	ERD와 DFD 결합	Bailin	Bailin[3]	
				EDFD(Entity Data Flow Diagram)	Booch	Booch[9]	
	기존시스템의 모든 관련 자료	데이터 및 프로세스	데이터 흐름 모형	구문(lexical)분석 방법에 기반을 둠	Booch	Booch[9]	
				객체 모형, 동적 모형, 기능 모형	SSADM	Yourdon[54][55]	
객체		데이터 모형, 프로세스 모형, 데이터/프로세스 매트릭스	일반적인 구조적 방법론	Information Engineering	Martin[36]		
			- 일반적인 소프트웨어 개발 방법론 - 구조적 방법론에 ISP 단계 추가	MOODD	Silva et al.[47]		
객체	객체의 정적요소 (NER) - 객체의 동적 행위 (UPM)	간단한 형태의 중간 단계 객체 모형을 작성	공식적인 객체 지향 분석 및 설계 방법	OMT UML	Rumbaugh et al.[43] Quatrani[41]		

2.1 데이터베이스 스키마를 입력 원천자료로서 활용하는 역공학

데이터베이스의 역공학에 관한 연구는 다음 세 가지로 분류할 수 있다: 관계형 스키마를 ER 모형으로 변환하는 방식, 네트워크 스키마를 ER 모형으로 변환하는 방식 및 계층형 스키마를 ER 모형으로 변환하는 방식이다. 첫째, 관계형 데이터베이스 스키마를 ER 모형으로 역공학 하는 방법론들은 함수적 종속성(functional dependencies)과 내포 종속성(inclusion dependencies)에 기반을 두고 있다 [Azar, 1986][Kantola et al., 1992]. 관계형 DBMS(Database Management System)내의 카타로그(catalog) 정보와 기존 데이터베이스의 튜플(tuples)을 활용함으로써 본래의 데이터베이스의 개념적 모형을 복구한다. Batini et al. [1992] 과 Navathe et al. [1987] 는 릴레이션(relations)을 주 키(primary key) 에 의해서 분류한다. 이러한 분류는 사용자와의 상호작용을 거쳐 이루어진다. 다음으로는 릴레이션의 분류를 기초로 관계형 스키마를 해석하고 변환한다. 카디널리티(cardinality) 제약 사항은 함수적 종속성의 분석을 통해 유도한다. 이 때 릴레이션들은 제3정규형(normal form) 또는 제2정규형으로 가정한다. Markowitz et al. [1990] 의 접근 방법은 관계형 스키마로부터 ER 스키마를 유도하기 위해 내포 종속성을 이용한다. Chiang et al. [1994b]의 접근 방법에서는 유일한 전제 조건으로서 데이터베이스 인스턴스 레코드의 존재를 요구한다. 그리고 릴레이션들을 Navathe et al. [1987] 과 유사하게 분류하며 그 다음에 몇 가지 경험적 법칙(heuristics) 을 적용하여 내포 종속성을 생성한다는 점에 있어 키기반 과 내포 종속성-기반 접근 방법의 혼합된 형이라고 할 수 있다.

다음은 네트워크 스키마를 ER모형으로 변

환하는 역공학 방법론에 관한 연구들이다 [Batini et al., 1992 ; Boulanger et al., 1989]. 이 접근 방법에 있어서의 어려운 점은 어떠한 레코드 타입이 엔티티(entity)를 나타내며 각각 관계(relationship)와 링크 레코드(link record)를 표현하는 지를 결정하는 것이다. 네트워크 스키마가 잘 정의된 형식(well-form) [Boulanger et al., 1989]인 경우라면 변환은 매우 단순하다. 그러나 네트워크 스키마가 잘 정의된 형식인 경우는 매우 드물며, 더구나 관련 정보 대부분이 응용 시스템 내에 내재한다. 따라서 이 접근 방법에서는 주어진 스키마의 의미 있는 정보를 밝혀내기 위해 사용자와의 상호작용이나 경험적 법칙이 활용되어야 한다.

마지막으로, 계층형 스키마를 ER모형으로 변환하는 방법 [Batini et al., 1992 ; Navathe et al., 1987] 은 계층형 스키마를 네트워크 스키마로 변환한 후 이를 다시 ER 모형으로 전환한다. 본 방법의 한 가지 어려움은 대부분의 무결성 제약 사항들과 구조적 정보들이 단지 사용자에 의해서만 분석이 가능한 응용 시스템들 속에 숨겨져 있다는 점이다.

관계형 스키마나 네트워크 데이터베이스 스키마로부터 ER모형을 유도하는 것은 상대적으로 단순하다. 계층형 스키마의 경우에 있어서는 먼저 계층형 데이터베이스 스키마를 네트워크 스키마로 전환하는 선행 작업이 요구된다. 또한, 데이터에 대한 대부분의 처리 규칙과 구조적 정보가 계층형 및 네트워크 데이터베이스를 활용하는 응용 시스템 내에 내재되어 있기 때문에 단지 데이터베이스 스키마만을 이용하여 개념적 모형을 복구한다는 것은 매우 어려운 일이다. 따라서 양질의 ER 모형을 획득하고자 할 경우 데이터베이스 스키마 정보만을 활용하는 것은 불충분하며 데이터베이스 스키마를 역공학 하는 방법 모두 사용자와의 상호작용을 요구한다.

2.2 원시 프로그램을 입력 원천자료로서 활용하는 역공학

원시 프로그램을 입력 원천자료로서 사용하는 대부분의 역공학 방법론들은 모듈 구조 및 프로그램 논리구조를 복구하는데 목적이 있다. 이러한 목적을 성취하기 위해 원시 프로그램을 분석하고 이를 의미 있는 정보 단위로 분할하는 방법을 적용한다.

RE2 (Reuse re-engineering)은 소프트웨어 재사용을 핵심 목적으로 하며 특별히 대상 기준(candidature criterion)을 정의하여 활용하는 실험을 시도하고 있다 [Canfora, 1993]. RE2 은 기존 시스템으로부터 재사용 가능한 모듈을 생성하는데 목적이 있다. 데이터 추상화에 초점을 두며 추상 데이터 타입을 찾아내기 위해 적용할 수 있는 수단으로서 처리규칙에 기초한 대상 기준이 제안되었다. 추상 데이터 타입을 찾기 위해 대상 기준이 필요하지만 기준을 설정하는 것이 난해하다.

Saleh et al. [1996] 은 본래의 시스템의 목적인 통신 서비스에 대한 정의와 설계 결정 및 개발 문서의 추출을 위해, 통신 소프트웨어에 적용할 수 있는 역공학 방법론을 제안했다. 이 방법에서는 통신 소프트웨어의 상위 수준 추상화를 Estelle에 기초하여 만들어 내는데, Estelle은 분산 시스템 프로토콜을 위한 ISO(International Standard Organization) 표준의 명세 언어이다. 이 방법에는 서비스를 제공하는 프로토콜 명세의 추출 외에 소프트웨어에 의해 분산 사용자에게 제공되는 서비스의 정의와 대응되는 매우 고 수준의 추상화가 포함되는데 두 가지 추상화 모두 Estelle을 이용해 공식적으로 기술된다. 그러나 Saleh et al. [1996] 의 방법은 특정 응용 시스템인 통신 소프트웨어에 중점을 두고 있으며 개념적 모형을 표현하는 Estelle 이 지나치게 물리적 구현에 가깝다.

Cutillo et al. [1993] 은 개발 문서화가 부족

한 대형 프로그램으로부터 업무 영역(domain) 독립적인 구성 요소들을 파악하고 추출하는 문제를 다룬다. 이 방법론은 Weiser의 분할(slice) [Cutillo et al., 1993]의 수정된 버전이라고 정의되며 특별한 종류의 코드 세그먼트(code segment)를 분류하고 추출하는데 목적이 있다. 정보 숨김(information hiding)의 원리에 따른 프로그램 구성 요소들을 개별화하고 이를 다시 재그룹화할 수 있는 장점을 제공한다. 이 방법론의 주요 산출물은 영역 독립적인 구성 요소를 위한 프로그램 구조도 (structure chart) [Yourdon et al., 1978]이다. 그러나, 어떠한 것이 영역 독립적이며 어떠한 것이 영역 종속적인 것인 지를 결정하는 것이 명확하지 않다.

2.3 기존 시스템의 모든 원천자료를 활용하는 역공학

시스템의 개념적 수준에서 의미 있는 정보를 복구하기 위하여 기존 시스템 환경 내에 존재하는 원시 프로그램, 데이터베이스, 사용자 인터페이스 및 다양한 유형의 개발 문서들을 모두 활용한다는 것은 매우 힘들다. 그 이유는 기존 응용 시스템들이 매우 오래된 시스템이므로 입력 원천자료가 제한되어 있거나 거의 남아 있지 않기 때문이다. COBOL/SRE(COBOL System Renovation Environment) 는 프로그램 세그멘테이션(program segmentation)이라고 불리는 소프트웨어 역공학 방법론으로서 오래된 코드로부터 재사용 가능한 자산의 복구를 목적으로 한다 [Ning et al., 1994]. 이 방법에서는 기존 시스템에서 획득 가능한 모든 자료 - 모듈, 파일, 프로그램, 서브 프로그램, 매뉴얼 및 JCL(job control language)를 활용한다. 일반적으로 복구된 구성 요소들이 본래 시스템에 비해 크기면에서 매우 작으므로 이들의 재사용성이 훨씬 높고 다른 플랫폼 요구사항에 맞춰 보다 쉽게 분산화가 가능하다. 이러한 접근 방법의 단점은 오래된 코드에 대

한 깊은 분석과 이해를 요구한다는 점이며 이는 사람에게 의존적이고 시간 소모성인 작업이다. 또한, COBOL/SRE는 기존 시스템내의 모든 원천자료를 필요로 하지만 기존 응용 시스템에서 이들 모두를 획득한다는 것은 어려운 작업이다.

REDO 방법론 [Lano et al., 1993]에서는 구조적 방법론과 Z [Pyle et al., 1991]와 같은 형식 언어 또는 객체 지향 방법론과 Z간을 상호 연계 하는 데에 상당한 노력을 기울인다. 형식 언어들은 구조적 방법론 및 객체 지향 방법론에서와 같은 비형식적 표기법에 의미 있는 정보를 부여하는 역할을 한다. 그리고 비형식 표기법은 형식언어에 의한 사양을 쉽게 이해할 수 있는 표현을 제공한다. REDO 방법론은 SSADM (Structured System Analysis and Design Method)을 이용하는 역공학 방법으로서 객체 지향 관점과 관계형 관점 간에 설정된 유사성에 기반을 두고 있다. REDO 방법론에서는 시스템에 대한 객체 지향 관점을 취하되 객체 지향 모형의 가장 중요한 요소인 클래스(class)를 관계형 테이블로 가정하고 있다. 이렇게 유도된 클래스는 수동적인 특성을 많이 내포하므로 객체의 동적인 행위와 연산(operation)과 같은 다른 중요 측면들을 반영하고 있지 못하다.

RECAST(Reverse Engineering into CASE Technology method) 는 기존 COBOL시스템의 원시 코드를 이용하여 SSADM 표현 형식으로 명세화된 시스템 표현을 유도한다 [Edwards, 1993]. 복구한 후 문서화된 시스템 설계도는 일부 시스템 개발 방법론과 연계 (CASE 툴의 활용을 통해) 되며 시스템의 수정 개발이나 시스템 대체를 지원하게 된다. RECAST는 기존 원시 코드로부터 시스템 설계의 완전한 복구를 위한 일련의 변환 규칙들을 제공한다. 변환은 절차적 형태로 정의되며 소프트웨어 툴에 의해 지원된다. 또한, PSL/PSA [Meta Systems Ltd., 1991] 리파지토리(repository)를 기반으로 하고 있으므로 특정 CASE 툴에 의존적이다. 방법론

의 자동화는 용이하게 성취하고 있지만 적용성이 약하다.

이외에 Lee et al. [1996]은 구조적 분석 산출물로부터 객체 모형을 유도하기 위한 연구를 시도했다. 이 연구는 구조적 방법론을 이용하여 개발된 시스템을 객체 지향 시스템으로 전환하거나 객체 모형을 설계하고자 하는 프로젝트들을 지원하는데 목적이 있다.

2.4 기존 연구의 요약 및 의의

지금까지 역공학과 관련된 주요 연구들을 살펴봐왔다. 이를 통하여 발견된 기존 방법론의 미비점을 기술하면 다음과 같다. 첫째, 기존 방법론은 프로그램 또는 데이터 관점 어느 한 측면만을 다룬다. 예를 들어, 프로그램을 입력 원천자료로서 활용하는 역공학 방법론들은 프로그램 모듈 구조와 이들 간의 상호관계를 발견하는데 중점을 두고 있다 [Cutillo et al., 1993]. 즉, 데이터 관점이 개념적 모형에서 제외되어 있다. 이에 반해, 데이터 구조나 데이터베이스 스키마를 이용하는 역공학 접근 방법은 개념적 데이터 모형만을 유도하는 것으로서 프로세스 또는 기능적 관점이 제공되지 않는다. 정보 시스템을 개발하기 위해서는 프로세스 모형과 데이터 모형 모두가 필요하다. 따라서, 현 역공학 방법론에 의해 획득되는 개념적 모형은 불완전하다고 할 수 있다.

둘째, 기존 역공학 방법론 대부분은 사용자 관점의 충실성 보다는 그들 방법론 자체의 자동화를 더 많이 지향하고 있다. 하지만 역공학이 사용자를 위한 것이며 그 결과도 결국은 사용자가 활용하는 것이다. 따라서 사람의 관여가 불가피한 상황이라면 사용자에게 보다 더 가까운 모형을 추구하는 것이 바람직하다고 판단된다. 즉, 역공학 방법론의 자동화가 중요하기는 하나 역공학 방법론의 가장 중요한 목적은 가능한 한 사용자에게 근접한 개념적 모형을 확

보하는 것이다.

셋째, 이들의 목표가 되는 개념적 모형(target conceptual model)이 ER모형 [Chen, 1976 ; Chen, 1985], 자료 흐름(data flow) 모형 [Yourdon, 1978 ; Yourdon, 1989] 및 이들의 변형된 형태와 같이 전통적 모형들이다. 전통적 모형들은 응용 시스템의 구조적인 요소와 행위 양면의 의미 있는 정보를 모두 표현함에 있어 최신의 객체 지향 모형에 비해 취약하다. 또한, 목표 개념적 모형들이 각 방법론들 간에 상이하다. 복구된 의미 있는 정보를 효과적으로 표현할 수 있는 표준 목표 모형이 제안될 필요가 있다.

넷째, 객체 지향 모형을 이용하는 문헌상의 많은 방법론들이 클래스, 속성 및 관계와 같이 객체 모형의 구조적인 측면의 표현을 주로 강조하고 있다. 이에 반해 시나리오와 같이 객체를 조작하는데 쓰이는 연산을 모형화 하기 위한 객체의 행위 측면의 설계에 대한 연구는 미진하였다. 이는 객체 모형의 표현만을 빌어 쓰는 것으로서 진정한 객체 모형의 장점을 활용하고 있지 못하다고 할 수 있다. 이러한 사실들을 고려할 때 현 역공학 방법론을 보다 발전시켜 업무의 동적인 면의 관리가 가능하도록 하여야 할 것이다.

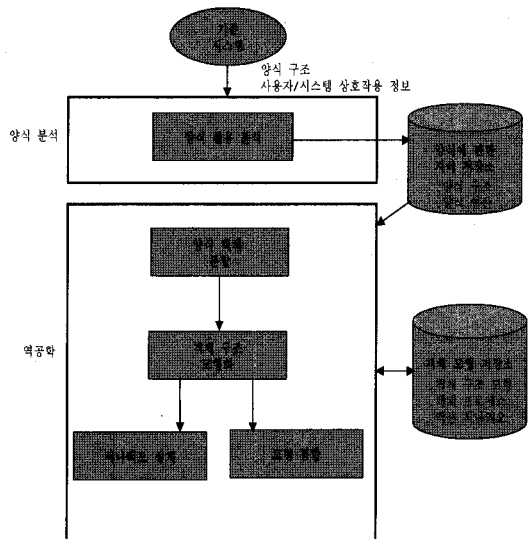
본 논문에서 제안된 FORE 방법론은 역공학 방법론 중의 하나로서 업무 양식을 입력 원천 자료로 이용한다. 특별히, 응용 시스템 화면 양식(electronic screen form)을 이용한다. 또한, 데이터와 프로세스의 의미 있는 정보 모두를 복구하고자 하며 데이터와 프로세스를 캡슐화(encapsulation)하는 객체를 시스템의 기본 관점으로 채택하게 되었고 이를 위해 목표 개념적 모형으로 객체 모형이 선택되었다. 복구된 의미 있는 정보는 기존 응용 시스템을 유지보수 하는데 유용하게 활용될 수 있다. 또한, 새로운 시스템을 객체 지향 기술을 이용하여 개발하고자 하는 경우 현재 운영되고 있는 시스템으로부터 복구된 의미 있는 정보를 개발 초

기의 시스템 모형으로서 활용할 수 있을 것이다. 한편, 기존 응용 시스템 서비스에 대한 접근이 요구될 때 복구된 의미 있는 정보 모형은 객체 래핑(wrapping) [Umar, 1997] 기술을 위한 지침을 제공할 수 있다. 또한, 복구된 객체 모형은 현 시스템의 유지보수 및 향후의 사용을 위한 객체 모형 리파지토리가 구축되면 그 내부에 축적되어 지속적으로 보완 발전될 수 있을 것이다.

Ⅲ. 역공학 방법론: FORE

3.1 아키텍처

FORE방법론은 5개 과정(phases) 으로 이루어진다 (양식 활용 분석, 양식 객체 분할, 객체 구조 모형화, 시나리오 설계, 모형 통합). <그림 2> 에서 FORE방법론의 아키텍처가 도식화 되었다. 단순 명료하게 도식화하기 위하여 각 과정들간의 환류는 표현하지 않았다. 그러나 환류는 각 과정에 있어서 산출물들의 개선을 위해 매우 중요한 요소이다.



<그림 2> FORE 방법론의 아키텍처

양식 활용 분석 과정에서는 양식 구조 및 기존 응용 시스템과 사용자 간의 상호작용 정보를 획득한다. 사용자에게 의한 기존 응용 시스템의 실행 동안 이벤트, 연산과 같은 사용자/시스템 상호작용에 의해 발생하는 정보와 양식 구조에 관한 정보들이 에이전트(agent) 프로그램에 의해 자동 수집된다. 에이전트 프로그램은 FORE 방법론의 지원을 위해 개발될 일종의 모니터링 프로그램이다. 이를 통해 획득된 정보들이 양식에 관한 지식 저장소(store)에 저장되고 다음 과정들에서 활용된다.

다음의 네 과정들은 역공학을 통한 객체 지향 개념적 모형화에 속한다. 이들 과정에서는 이전 과정에서 수집한 양식에 관한 지식을 입력으로 활용한다. 양식 객체 분할 과정에서는 양식에 관한 지식 저장소로부터 양식에 관한 지식을 읽어 들인다. 그 다음에 그룹 필드(group field), 블록 필드(block field) 및 연산과 같은 정보의 유형에 따라서 양식에 관한 지식을 의미 있는 정보 단위로 분할한다. 객체 구조 모형화 과정의 핵심 목적은 이전 과정의 결과로부터 객체들을 파악하는 것이다. 또한, 객체들 간에 기본적인 구조적(structural) 관계와 협업(collaboration) 관계가 설정된다. 이 과정에서의 주요 산출물은 객체 이름, 객체 속성 및 구조적 관계로 이루어지는 객체 구조 모형이다. 시나리오 설계 과정에서는 양식에 관한 지식 저장소 내의 연산 정보들을 활용한다. 우선, 어떠한 객체가 사용자 키인 연산, 시스템에서 담당하거나 처리하는 시스템 제공 처리 및 수식의 계산과 같은 연산들을 수행하는 지를 파악한다. 그 다음에는 객체들 간에 발생하는 일련의 연산들을 시간 순서에 의해 기술한다. 시나리오 설계 과정의 주요 산출물은 객체 프로세스 액션 시나리오 다이어그램(object process action scenario diagram)이다. 마지막으로 모형 통합 과정에서

는 단위 모형들을 통합하여 보다 상위 수준에서 통합 모형을 개선한다. FORE 방법론의 일련의 과정을 거쳐 출력되는 최종 산출물은 객체 구조 모형과 객체 프로세스 액션 시나리오이다.

모든 작업은 출발 전에 시작을 위한 준비가 요구된다. 선행 준비 과정은 FORE 방법론의 기본 프로세스에는 속하지 않지만, 기존 응용 시스템의 역공학을 위한 프로젝트를 준비하기 위해 설정되어야 한다. 역공학의 실제 활동이 수행되기 전에 프로젝트를 위한 결정사항 및 필요한 내용들이 정해진다. 첫째, 역공학 대상 응용 시스템을 선정한다. 또한, 대상 범위와 관련 정보들을 정하고, 응용 시스템의 주요 사용자를 파악한다. 둘째, 기존 응용 시스템 내에서 양식을 이용하는 양식 프로세스가 정의되어야 한다. 하나의 양식을 처리하는 업무(task)를 양식 프로세스로 가정한다. 셋째, 양식 프로세스들간의 종속성이 파악되어야 한다. 즉, 어떠한 양식을 먼저 처리하고 그 다음에 어떠한 양식을 처리해야 하는지의 선후 관계가 정해져야 한다.

3.2 방법론 세부 사항

3.2.1 사례

본 연구에서는 FORE 방법론의 응용성 예시를 위해 잘 알려진 예를 사용하였다. <그림 3>은 Choobineh et al.[1992]에서 사용된 예로서 입력 자료가 되는 2개의 양식을 보여 주고 있다. 그림은 WorkOrder 양식과 JobAssignment 양식의 구성과 특정한 경우의 값을 가지고 있는 WorkOrder 양식의 사용 예를 보여 준다. Choobineh의 방법론은 사용자의 문서 양식과 응용 시스템 화면 양식을 사용하여 ER 다이어그램을 유도하였다.

업무 양식에 근거한 객체 지향 역공학 방법론

Date **2-5-90** Work Order Number **153**

Bill To: **XYZ Headquarters** Job Location: Name **XYZ Street #5**

Name **1282 Main Street** Address **65 Railroad Street**

Address **Houston, Texas 70000** **Somecity, Texas 71000**

Date Required **3-5-90**

Task	COST/SQFT	SQFT	Amount
Wall Insulation	15.00	40	600.00
Ceiling Acoustics 25	12.0	25	300.00
Total Before Tax			900.00
Tax(5%)			45.00
Total			\$945.00

Customer Type **Goodman** Sales Person **Goodman**

Work Order 양식

Date of Job **2-20-90** Assignment No **5**

Job Location: Name **XYZ Street #5**

Address **65 Railroad Street**
Somecity, Texas 71000

Crew Foreman **Joe Foreman** Work Order No **153** Vehicle No **--**

Task	QFT	Material	Bin-No	# of Bags
Wall Insulation	40	Fiber Glass	4	8
			5	2
Ceiling Acoustics 25		Rock Wool	10	3
			11	5
		Foam Board	25	8

Authorized By **Sally Supervisor**

Job Assignment 양식

<그림 3> 사용 양식 예

FORE 방법론이 객체 모형을 유도하는 것이 때문에 FORE 방법론에서는 양식의 구조뿐 아니라 연산 부분도 필요하다. 따라서, 양식 자체 외에 각 양식을 이용한 작업 절차를 다음과 같이 가정해서 설계했다.

WorkOrder 프로세스:

사용자인 세일스 맨은 WorkOrder 프로세스를 요구한다. WorkOrder 프로세스는 사용자 ID의 제출을 요구한다. 사용자 자신의 ID와 패스워드

를 제출할 때마다 WorkOrder 프로세스는 그것을 체크 한다. 사용자 ID와 패스워드가 통과되면 WorkOrder 프로세스는 자신의 양식을 제시하고 이를 사용자에게 제공한다. 사용자 ID와 패스워드가 틀리면 세 번까지 허용하고 그 이상이면 종결한다. WorkOrder 양식이 제공되면 주문 날짜(date)와 주문 번호(workOrder Number)가 시스템에 의해 자동적으로 제공된다. 다음으로 는 사용자가 대금을 지불할 고객을 입력한다. 그리고 WorkOrder 에 대한 위치도 입력한다. 주

문이 된 각 태스크에 대해서 사용자는 다음과 같이 처리한다. 각 태스크에 대해 사용자가 필요한 양을 정의하면 WorkOrder 프로세스는 데이터베이스로부터 단위 비용(cost/ sqft)을 가져와서 비용(amount)을 계산한다. 주문한 모든 태스크가 완료되면 WorkOrder 프로세스는 totalBefore Tax, tax 및 total을 계산하고 이를 화면 양식 상에 출력한다. 마지막으로 WorkOrder 양식의 사용자가 확인 서명한다.

JobAssignment 프로세스:

사용자인 작업 배당 관리자(supervisor)는 자신의 ID와 패스워드를 입력한다. 시스템은 사용 권한을 확인한다. 3회까지 재시도할 수 있다. 만약 승인되면 시스템은 JobAssignment 양식을 제공한다. 먼저, 사용자는 workOrderNumber를 입력한다. JobAssignment 프로세스는 workOrder Number를 사용하여 데이터베이스로부터 관련 값을 가져와서 대응되는 양식 필드에 값을 채운다. 사용자는 주문된 WorkOrder를 수행할 기술자(foreman)를 지정한다. 각 태스크에 대해 사용자는 필요한 재료, bin_no, # of bags들을 지정한다. 마지막으로 사용자가 서명하고 종료한다.

3.2.2 양식 활용 분석

이 과정에서는 양식에 관한 모든 정보가 수집된다. 양식은 크게 문서 양식과 응용 시스템 화면 양식으로 구분할 수 있는데 현재 동작 중인 기존 응용 시스템을 대상으로 한다는 점을

고려하여 응용 시스템 화면 양식을 중점으로 다룬다. 양식은 주문 양식, 세금 양식, 제품 송장 및 공급 요청 양식 등과 같이 다양한 종류의 정보를 처리하는 방법을 단순화 하고 표준화 하기 위해 가장 보편적으로 사용된다. 양식의 구조적인 정보는 양식의 상태를 나타내며 양식에 대한 연산은 보통 조직의 의사 결정을 반영한다. 양식의 연산은 시간에 따라 업무를 제어하며 양식의 흐름은 업무에 있어서 데이터 흐름이 된다. 본 논문에서는 양식이 가지고 있는 이와 같은 객체적인 특성 때문에 양식을 가장 원시적인 형태의 객체로 간주한다. FORE 방법론에서는 양식이 가지고 있는 이러한 특성들을 기초로 객체 구조와 메쏘드(method)를 찾는다. 양식의 행위는 매우 동적이며 모험화 관점에서 복잡하므로 보다 많은 의미 있는 정보를 표현하기 위한 수단으로서 객체 모형이 제안될 필요가 있다.

본 과정에서는 모니터링 툴을 개발하여 양식 구조와 사용자/시스템 상호작용에 의해 발생하는 정보를 수집하는데 활용한다. 이 툴은 기존 응용 시스템이 실행되는 동안 이벤트의 종류에 따라 양식 구조, 사용자/시스템 상호작용 및 데이터베이스 관련 접근 정보들을 획득하게 된다. 이렇게 수집된 정보들은 양식에 관한 지식 저장소에 축적되고 후속 과정들에서 활용된다. 이 과정의 목적은 양식에 관한 가능한 한 많은 정보들을 모으는데 있다. 이 과정의 수행 절차는 다음 <표 2> 에 기술되어 있다.

<표 2> 양식 활용 분석 과정

과 정	단계별 절차	산출물
양식 활용 분석	1. 대상 응용 시스템을 단일 양식을 갖는 최하위 프로세스로 분할 2. 시스템 동작 중 사용자와의 상호작용을 통해 생성되는 모든 정보 수집 3. 양식에 관한 지식 저장소에 수집된 정보 저장 4. 대상 응용 시스템 내의 양식을 이용하는 모든 프로세스가 완료될 때까지 단계2에서 단계 3 반복	양식에 관한 지식 저장소 · 양식 구조 정보 · 사용자/시스템 상호작용에 의해 발생하는 정보

지원 틀은 이 과정에 있는 모든 단계들을 자동적으로 수행할 수 있도록 도와준다. 이 과정의 주요한 산출물은 지원 틀에 의해 수집될 양식에 관한 지식이다. 양식에 관한 지식은 두 부분으로 이루어져 있는데 하나는 양식 구조에 관한 것이며, 다른 부분은 사용자 연산과 데이터베이스 접근 연산에 관한 것이다. 다음 <표 3>은 WorkOrder 양식에 관한 지식의 한 예로서 이 것은 <표 2>의 절차를 기초로 개발될 지원 틀을 활용할 경우 획득하게 될 가상의 지식이다. <표 3a>는 양식에 관한 지식 저장소에 저장된 양식 구조에 관한 사용 예이다. 양식 자체와 양식 필드 두 가지 형태의 양식 구조가 존재한다. 양식 구조에서 캡션(caption) 이름은 양식 필드의 의미 있는 정보를 알기 쉽게 설명해주는 역할을 하는 헤더 필드(header field)이다. 기존 응용 시스템이 실행되는 동안 양식 필드의 값이 변수 이름에 실질적으로 할당된다.

다섯 가지 유형의 양식 필드는 다음과 같다. 첫째, 양식 자체는 서브 객체들로 이루어진 복

합(complex) 객체이다. 둘째, 원자 타입(atomic type)은 더 이상 분할될 수 없는 필드이다. 셋째, 블록 타입은 단일한 캡션 필드 이름하에 그 이름과 연계되는 여러 개의 양식 필드들이 함께 모여 있는 필드의 형식이다. 넷째, 그룹 타입은 하나의 캡션 필드 이름하에 동일한 의미의 양식 필드들이 배열 형식으로 반복해서 나타나는 필드들의 모임이다. 다섯째, 컴퓨테이션(computation)은 값이 어떤 수식에 의해 계산되는 필드이다. 양식의 연산에 관한 정보의 사용 예가 <표 3b>에 제시되어 있으며 <표 3b>의 각 칼럼들은 다음과 같다: 연산의 순서는 시스템과 사용자와의 상호작용의 순서이다. 본래의 양식 구조 내의 필드 배치 순서에 따라 만들어진 테이블에 연산의 순서가 추가되었다. 연산의 타입은 사용자 키인, 시스템 제공 및 데이터베이스 접근과 같은 연산의 종류를 의미한다. 대응하는 양식 필드는 연산자(operator)에 의해 활용되는 변수(operand)이다. 결과는 프로세스가 완료된 후 변수에 배정된 값을 의미한다.

<표 3a> WorkOrder양식의 양식에 관한 지식 예: 양식 구조

내 용		양식 필드 타입		
타 입	캡션 이름	변수 이름		
Form Name	WorkOrderForm	WorkOrder_v	Complex Object	
Form Field	WorkOrderNumber	WorkOrderNumber_v	Atomic	
	Date	Date_v	Atomic	
	DateRequired	DateRequired_v	Atomic	
	BillTo-Name	- BillToname_v	Block	
	BillTo-Address	- BillToaddress_v		
	JobLocation-name	- JobLocationname_v	Block	
	JobLocation-address	- JobLocationaddress_v		
	Task	Cost/sqft	Task_v1, task_v2 ...	Group
		Sqft	Cost/sqft_v1, cost/sqft_v2 ...	Group
		Amount	Sqft_v1, sqft_v2 ...	Group
	TotalBeforeTax	Amount_v1, amount_v2 ...	Group, Computation	
	Tax	TotalBeforeTax_v	Computation	
	Total	Tax_v	Computation	
SalesPerson	Total_v	Computation		
	SalesPerson_v	Atomic		

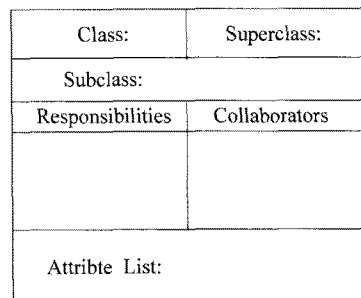
<표 3b> WorkOrder 양식에 관한 지식 예: 양식 프로세스의 연산

연산 순서	연산 타입	대응 양식 필드	결 과
2	User Keyin	WorkOrderNumber_v	153
3	User Keyin	BillToName_v	XYZ Head Quarters
4	Database Access	BillToAddress_v	1283 Main Street Houston, Texas 70000
5	User Keyinv	JobLocationName_v	XYZ Street #5
6	User Keyin	Job_locationAddress_v	65 Railroad Street Somecity, Texas 71000
8	User Keyin	Task_6v1, task_v2 ...	Wall insulation
10	User Keyin	Sqft_v1, sqft_v2 ...	40
7	User Keyin	DateRequired_v	3-5-90
16	User Keyin	SalesPerson-v	Goodman
1	System Provided	Date_v	2-5-90
9	Database Access	Cost/sqft_v1, cost/sqft_v2 ...	15.00
13	Database Access	TaxRate_v	5%
11	Computation	Amount_v1, amount_v2 ...	600.00
12	Computation	TotalBeforeTax_v	900.00
15	Computation	Total_v	945.00
14	Computation	Tax_v	45.00

3.2.3 양식 객체 분할

양식 객체 분할 과정은 양식에 관한 지식 저장소에 축적된 양식에 관한 지식을 분석하고 이를 보다 의미 있는 정보 단위로 분할한다. 이 과정의 목적은 복합 양식 객체를 보다 단순한 객체로 나누는 것이다. 복합 양식 객체는 블록 필드, 그룹 필드 및 데이터베이스 접근 필드와 같은 요소들로 구성된다. 이 요소들은 중요한 객체 후보가 된다. FORE 방법론에서 양식에 관한 지식 저장소 내의 모든 양식에 관한 지식에 대해 분할 개념이 적용된다 [Cutillo et al., 1993]. 객체를 표현하기 위해 CRC(Class, Responsibilities, Collaborators) 카드[Wilkinson, 1995]가 도입되었고 이를 FORE 방법론에 맞춰 확장하여 사용한다. 본 논문에서는 이 확장된 방법을 ECRC(Extended CRC)라고 칭하겠다. 다음 <그림 4>는 ECRC의 기본 표현 형식이다. 실선은 구조적 관계로서 각 유도된 객체와 본래 위치가 되는 컨테이너 객체 간의 관계에서 파악되며, 협업

관계는 컴퓨테이션 필드들을 분석함으로써 정의되며 점선으로 표현된다. 슈퍼/서브 타입 관계는 입반화 객체와 상세화 객체간의 관계를 나타내는 표현이다. FORE 방법론을 지원하기 위해 개발 중인 ECRC 도구(ECRC Tool for FORE Methodology)를 이용하여 객체 모형을 작성했다.



구조적 관계: _____
 협업관계:
 슈퍼/서브 타입 관계: _____➔

<그림 4> ECRC 표현 형식

<표 4> 양식 객체 분할 과정

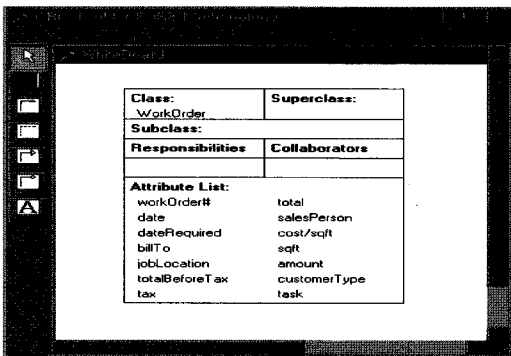
과 정	단계별 절차	산 출 물
양식 객체 분할	<ol style="list-style-type: none"> 1. 복합 양식 객체 작성 <ol style="list-style-type: none"> 1.1 양식에 관한 지식 저장소의 양식 구조를 이용하여 복합 양식 객체를 만듦 1.2 양식 이름을 이용하여 객체 명명 1.3 양식 필드들을 객체의 속성으로 할당 2. 복합 객체를 평평하게 만듦 <ol style="list-style-type: none"> 2.1 복합 객체 내의 그룹필드와 블록 필드를 찾음 2.2 그룹 필드와 블록 필드를 새로운 객체로 변환 2.3 새로 유도된 객체 명명 3. 데이터베이스 연산을 기반으로 양식 객체를 분할 <ol style="list-style-type: none"> 3.1 데이터베이스로부터 값을 가져오는 양식 필드를 파악 3.3 대응하는 데이터베이스 테이블 이름을 이용하여 새로운 객체로 작성 3.4 테이블의 관련 필드들을 객체 속성으로 부여 4. 추가적인 양식 객체 유도 <ol style="list-style-type: none"> 4.1 나머지 후보 객체를 찾음 4.2 후보가 객체가 될 수 있는지 없는지 결정 4.3 새로 유도된 객체를 객체 목록에 추가 	분할 된 새로운 객체

이 과정은 <표 4>와 같은 절차를 갖는다. 이 과정에서는 이전 과정에서 수집된 양식에 관한 지식을 다음과 같이 분석한다. 단계1에서는 WorkOrder 양식을 이용해 복합 양식 객체를 만든다. 복합 양식 객체는 WorkOrder 양식의 필드들 모두를 속성으로 가지며 <그림 5>는 ECRC 도구를 이용하여 작성된 WorkOrder 양식 객체를 보여 준다.

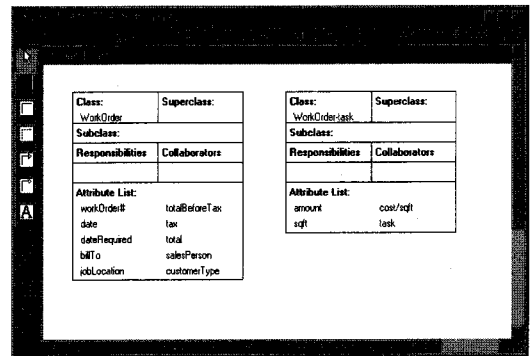
단계2에서는 WorkOrder 양식 객체 내의 속성 중 그룹 필드들을 외부로 추출하여 복합 양

식 객체가 평평한 (flat) 객체가 되도록 한다. <그림 6>에서 WorkOrder-Task는 WorkOrder 양식 객체로부터 새로이 생성된 객체이다. WorkOrder 양식 객체를 분할한 후 객체가 단순화 되었음을 확인할 수 있다. 객체 이름의 중복과 혼동을 피하기 위하여 새로운 객체의 이름은 본래 객체의 이름에 새로 추출된 객체 이름(양식 필드를 이용) 을 합성하여 명명한다.

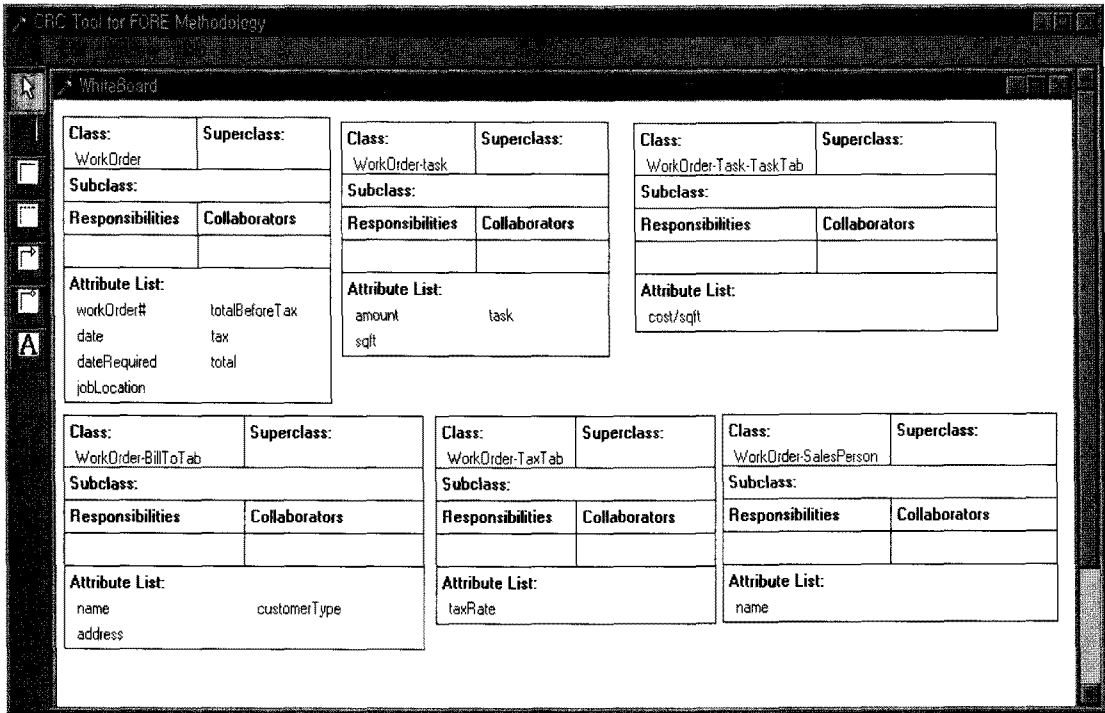
단계3에서는 각 객체 내에 있는 어떠한 양식 필드가 데이터베이스를 접근하는지를 파악한다.



<그림 5> WorkOrder 양식 객체



<그림 6> Work Order 양식 객체가 평평하게 된 후의 객체



<그림 7> Work Order 양식 객체에 대해 분할 적용 및 추가로 파악된 객체

WorkOrder-Task 내의 cost/sqft 필드의 값은 WorkOrder-Task-TaskTab 이라는 데이터베이스 테이블로부터 가져 오게 된다. Work Order 객체 내의 billTo 필드 또한 Work Order-BillToTab 이라는 데이터베이스 테이블로부터 값을 접근한다.

단계4에서는 추가적으로 객체 후보들을 파악한다. WorkOrder-SalesPerson 객체는 양식에 대한 작업자로서 동적 객체의 역할을 한다. Work Order 객체의 tax 계산시 taxRate가 필요하며 이는 데이터베이스의 tax 테이블 WorkOrder-TaxTab 으로부터 값을 가져 온다는 것을 알 수 있다. <그림 7>은 추가적으로 파악한 Work Order-SalesPerson, WorkOrder-Task-TaskTab, Work Order-BillToTab, WorkOrder-TaxTab 객체를 보여 준다.

3.2.4 객체 구조 모형화

객체 구조 모형은 이전 과정에서 분할된 객체들을 이용해서 만들어 진다. 객체 구조 모형은 UML [Quatrani, 1998][Rational Software Corp., 1996]의 클래스 다이어그램이나 OMT에서의 객체 모형과 유사하다. 이전 과정에서 객체의 속성들은 이미 배정되었지만 객체는 그 자신의 메소드를 가지고 있지 않다. 객체가 데이터와 연산 부분으로 구성되기 때문에 객체를 구성하기 위해서는 두 부분 모두가 요구된다. 그러나 객체 모형을 활용하고 있는 대부분의 기존 방법론들은 단지 객체의 구조적인 측면에만 초점을 두고 있다 [Blahe et al., 1988]. 이러한 방법론은 객체 지향 모형의 표현만을 단지 활용하고 있다고 할 수 있다.

FORE방법론에서는 컴퓨테이션 필드로부터 객체의 연산 부분을 파악하고자 시도하는데 그

이유는 양식에 대해 행해지는 많은 연산들이 컴퓨테이션 필드 속에 숨겨져 있기 때문이다. 그리고 양식에서의 연산들이 메소드의 주요한 후보가 되기 때문이다. 컴퓨테이션 필드는 연산자와 변수로 이루어진 수식으로부터 반환된 값을 갖는다. 수식에서 연산자는 수리적인 부호로서 더하기, 빼기, 곱하기, 나누기 등을 의미한다. 변수는 수식에서 활용되는 변수로서 객체의 속성이 된다. 기존 응용 시스템의 원시 프로그램 없이 지원 틀을 통해 시스템에 의해 제공되는 컴퓨테이션 필드와 양식 필드들로부터 자동적으로 메소드를 완벽하게 찾는 것은 어렵고 불가능하기까지 하다. 이는 FORE 방법론의 중요한 입력 원천자료로서 양식 구조와 사용자/시스템 상호작용에 의해 발생하는 정보만이 활용

되는데 기인한다. 사용자 키인과 데이터베이스 접근에 의해 값을 배정받는 필드 외의 모든 양식 필드는 컴퓨테이션 필드라고 볼 수 있다. 이러한 컴퓨테이션 필드들을 찾아내기 위해서는 사용자나 설계자의 관여가 요구된다.

이 과정에서, 우리는 컴퓨테이션 필드들을 분석함으로써 객체의 메소드를 찾고, 그 다음에 객체간의 관계를 찾는다. 대부분의 구조적 관계는 분할된 객체와 본래의 컨테이너(container) 객체에 관한 정보로부터 쉽게 유도될 수 있다. 협업 관계는 연산에 참여하고 있는 필드들 간의 연관 관계로부터 유도가 가능하다 [Lee et al., 1998]. 다음 <표 5>는 상기한 내용에 의거 객체 모형을 유도하기 위한 절차를 기술한 것이다.

<표 5> 객체 구조 모형화 과정

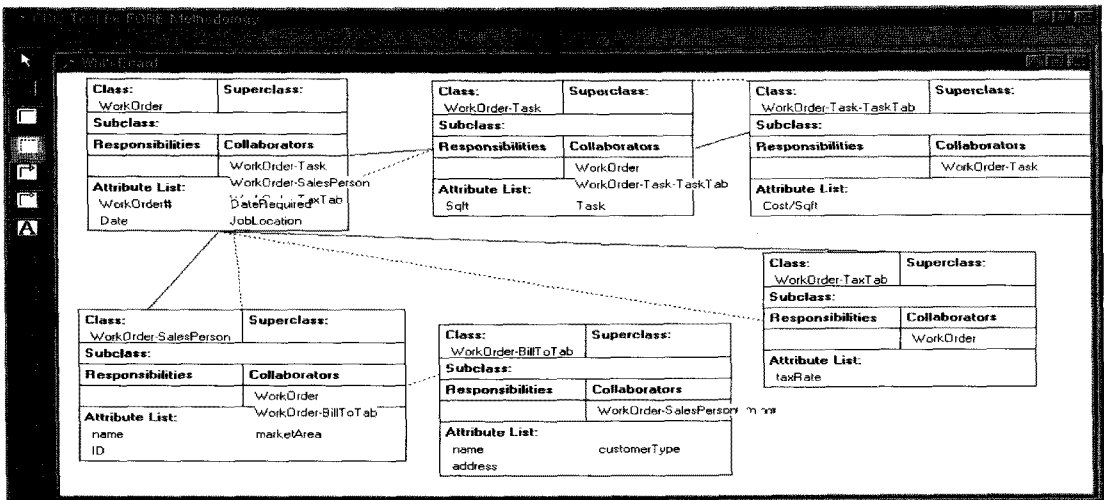
과 정	단계별 절차	산 출 물
객체 구조 모형화	<ol style="list-style-type: none"> 1. 컴퓨테이션 필드 발견 <ol style="list-style-type: none"> 1.1 필드 타입을 기초로 컴퓨테이션 필드 후보를 찾음 1.2 컴퓨테이션 필드 여부 판단 1.3 컴퓨테이션 필드 표시 2. 역공학 엔지니어의 지원에 의해 컴퓨테이션 필드 분석 <ol style="list-style-type: none"> 2.1 컴퓨테이션 필드로부터 수식 유도 2.2 수식을 최하위 연산과 원자 필드로 분할 2.3 모든 원자 필드에 대한 값의 원천을 찾음 3. 객체 메소드 정의 <ol style="list-style-type: none"> 3.1 기본 연산과 원자 필드만으로 구성된 수식 선택 3.2 수식을 기초로 단위 기능 구성 3.3 중첩된 기능이 존재한다면 중첩된 기능으로부터 단위 기능을 유도 3.4 단위 기능을 이용하여 객체의 메소드 작성 4. 객체 메소드 할당 <ol style="list-style-type: none"> 4.1 객체 역할 파악 4.2 객체를 서버 객체와 클라이언트 객체로 구분 4.3 객체 메소드를 서버 객체에 할당 5. 객체간 관계 설정 <ol style="list-style-type: none"> 5.1 본래 객체와 필드로부터 유도된 객체간에 구조적 관계 정의 5.2 서버 객체와 클라이언트 객체간에 메소드를 통한 협업 관계 정의 	객체 구조 모형

컴퓨터이션 필드들의 분석을 통해 Work Order 객체로부터 세개의 컴퓨터이션 필드를 찾아냈다. 이들은 amount, totalBeforeTax 및 total 필드이다. 이 컴퓨터이션 필드들은 최하위 (primitive) 연산과 원자 필드(atomic field)들로 분할된다. 먼저, amount는 cost/sqft와 sqft 필드의 곱셈에 의해 계산된다. 다음으로 totalBeforeTax는 태스크별 amount 필드의 합이다. 즉, 최하위 수준에서 totalBeforeTax 필드는 sum(cost/sqft×sqft) 이다. 마지막으로 total은 totalBeforeTax와 tax의 합이다. 다시 tax는 taxRate와 totalBeforeTax의 곱셈에 의해 계산된다. taxRate의 값은 관련 데이터베이스 테이블 WorkOrder-TaxTab 으로부터 값을 접근 한다. 이 필드들이 각각 최하위 수준의 필드들이다.

객체 메소드는 최하위 연산에 의해 정의된다. 이들의 프로세스 논리구조는 필드 이름을 분석함으로써 유추되는데 그 이유는 FORE 방법론에서 기존 응용 시스템 내부에 내장되어 있는 프로세스 논리구조를 추출하기가 어렵기 때문이다. 따라서, 컴퓨터이션 필드의 의미 있는 정보 이름으로부터 프로세스 논리구조를 추론하는 것은 사용자나 역공학 엔지니어에 의존하게

된다. 응용 시스템의 원시 프로그램 대신에 양식을 이용하는 본 방법론의 경우에 있어서 컴퓨터이션 필드 이름을 분석함으로써 프로세스 논리구조를 추론하는 것은 불가피한 선택이다. 그 이유는 프로세스 논리구조 대부분이 원시 프로그램 내에 명령어로 구현되지만 양식 필드 또는 변수에 그 값을 배정하는 형식으로 수행하므로 변수 이름에 해당하는 컴퓨터이션 필드들이 프로세스 논리구조를 함축하고 있기 때문이다. 단계1로부터 네개의 객체 메소드 totalBeforeTax(), amount(), total(), tax()를 유도해냈다. 각 메소드는 연산과의 혼동을 피하기 위해 함수의 형식으로 표현한다.

객체 메소드는 기본적으로 서버 객체에 할당되게 된다. 따라서 어느 객체가 서버 객체인지를 결정하는 것이 중요하다. 이 때 객체의 책임(responsibility)을 중요 변수로서 고려할 수 있을 것이다. 예를 들어, sum(amount), total(), totalBeforeTax() 및 tax()는 WorkOrder 객체가 해당 연산을 수행해야 하기 때문에 모두가 WorkOrder 객체의 중요 책임이다. 그리고 amount()는 태스크의 sqft에 의존하므로 Work Order-Task 객체가 그 메소드에 대한 책임을 갖게 된다.



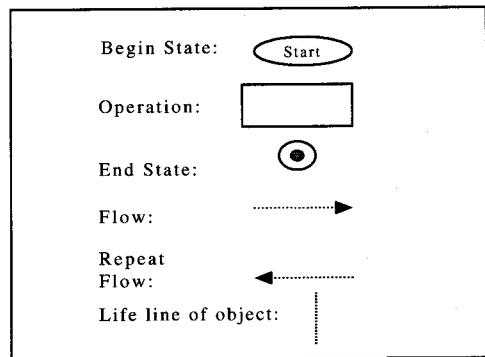
<그림 8> WorkOrder 양식에 대한 최종 객체 구조 모형

단계5에서는 객체들 간의 관계를 연결함으로써 객체 구조 모형을 완성한다. 구조적 관계가 먼저 추가되고, 다음으로 협업 관계가 형성된다. 서버 객체와 클라이언트 객체 간의 역할이 정해 질 때, 객체 메소드도 또한 명확하게 결정된다. 예를 들어, WorkOrder-Task 객체는 Work Order 객체로부터 유래되었다. 따라서 이 둘 사이에는 구조적 관계가 존재하게 된다. 같은 방식으로 WorkOrder-Task객체와 WorkOrder- Task-TaskTab 객체, WorkOrder와 WorkOrder -BillToTab, WorkOrder와 WorkOrder-TaxTab, 그리고 Work Order와 WorkOrder-SalesPerson 간에 각각 구조적 관계가 존재한다. 한편, Work Order 객체의 totalBeforeTax의 값을 얻기 위해서는 Work Order-Task 객체에 의해 제공되는 amount 필드의 값이 요구된다. 따라서, Work Order 객체와 WorkOrder-Task 객체 간에 메소드 amount()를 통해 협업 관계가 설정된다. 다음 <그림 8>은 ECRC 카드를 이용하여 작성된 객체 구조 모형을 보여주고 있다.

3.2.5 시나리오 설계

객체 행위를 모형화하기 위해 시나리오가 도입되었다. 시나리오는 하나의 유즈 케이스(use

case)의 실 예로서 스크립트(scripts)와 유사하지만 [Jacobson, 1995], 시스템 실행동안 발생하는 사건들을 순차적으로 기술한다는 면에서 중요한 차이점이 있다. FORE 방법론은 이전 단계에서 파악된 객체들을 이용하여 실행 시나리오를 구성한다. 시나리오를 그래픽하게 표현하기 위해 SOHDM(Scenario-based Hypermedia Development Methodology) [Lee et al., 1997 ; Lee et al., 1998] 의 SAC(scenario activities chart) 를 부분적으로 수정하여 객체 프로세스 액션 다이어그램으로 구성했다. <그림 9>에 시나리오를 묘사하기 위한 그래픽 표현 기호가 제시되어 있다. <표 6>은 시나리오 설계를 위한 절차를 보여준다.



<그림 9> 객체 프로세스 액션 시나리오 표현 기호

<표 6> 시나리오 설계 과정

과 정	단계별 절차	산 출 물
시나리오 설계	1. 목표 시나리오 선정 1.1 단일 양식을 이용하는 업무 프로세스 파악 2. 업무 프로세스 안에 포함되는 객체 선택 2.1 이벤트와 액터 파악 3. 1차 시나리오 작성 3.1 객체 연산을 순차화 함 3.2 연산 수행 주체가 되는 액터 객체 정의 3.3 변수 필드를 보유하고 있는 컨테이너 객체 파악 4. 설계자 또는 역공학 엔지니어의 관여를 통해 상세 시나리오 기술 4.1 이벤트가 발생하는 위치를 출발점으로 표시 4.2 실행 순서에 따라 각 객체에 대한 연산을 배열 4.3 객체 간에 연산의 협업 관계를 설정 4.4 객체가 최초 생성되는 시점에 객체의 생명선 표시 4.5 업무 프로세스의 마지막에 종료 표시	객체 프로세스 액션 시나리오

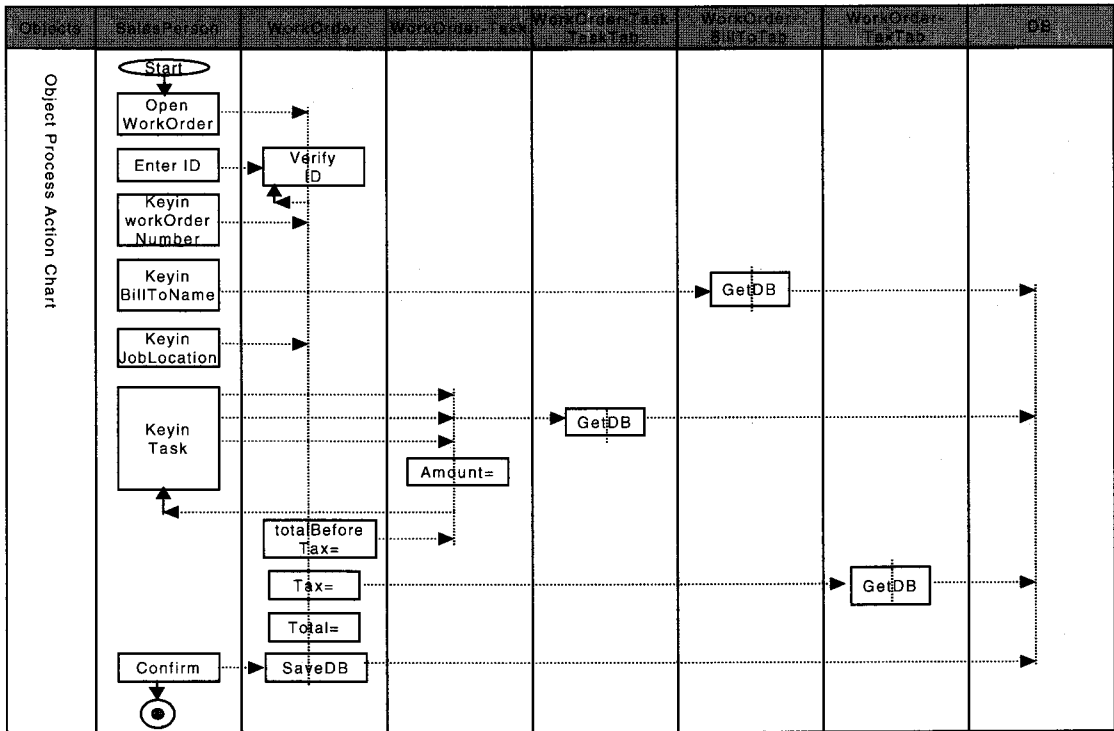
사용자는 작업 기술서에 의거 양식을 이용한 업무 프로세스를 수행한다. 이 예에서 세일즈맨이 사용자로서 WorkOrder 양식을 이용한다. 사용자와 양식을 기반으로 한 응용 시스템 간에 일련의 처리들이 시나리오처럼 묘사된다. 단계1에서 시나리오를 단순화하기 위해 단일 양식을 이용한 프로세스를 단위 시나리오로 가정하고 대상 시나리오를 정했다. 예를 들어, Work Order 프로세스가 WorkOrder 양식을 이용하는 업무 프로세스가 된다. JobAssignment 양식의 경우에는 JobAssignment 프로세스가 있다. 액터 객체(actor object)인 세일즈맨이 시스템에 Work Order 를 의뢰했을 때 이벤트가 발생하고 프로세스가 시작된다. 각자의 역할과 책임을 가지고 모든 객체는 시나리오에 참여한다. 부가적으로 각 객체의 고유한 책임은 역공학 엔지니어나 설계자에 의해 부여된다. 또한, 데이터베이스 객체처럼 물리적인 객체를 이 단계에서 고려하여 포함시킬 수도 있다.

<표 7> 은 양식의 연산을 이용하여 유도된 1차 시나리오이다. 이 테이블은 보다 풍부한 의미 있는 정보를 갖는 시나리오를 설계하기 위

한 출발점이다. 하나의 업무 프로세스가 완료될 때까지 모든 연산과 상호작용 정보들이 이 과정의 단계3에서 기술된다. 먼저, 시나리오에 참여하는 객체들이 최상단 좌측에서부터 우측으로 배열된다. 연산 순서에 의거하여 사용자인 액터 객체가 먼저 WorkOrder 객체를 사용하기 위해 호출한다. 액터 객체가 WorkOrder 객체를 대상으로 workOrderNumber를 입력하면 WorkOrder 객체는 WorkOrder-Task 객체의 값을 입력할 수 있도록 WorkOrder-Task 객체를 제공한다. 각 객체별 연산이 사각형 안에 표기된다. 객체 간에는 연산 실행에 따른 클라이언트/서버 객체 서비스 관계가 화살표로 표시된다. 객체가 담당하는 연산이 처음 요청될 때 객체가 생성되며, 자신의 역할이 종료되면 객체의 수명도 끝나게 된다. 객체의 생성에서 종료까지의 객체 수명을 생명선(lifeline)으로 표시한다. <그림 10>은 WorkOrder 프로세스의 실행 시나리오를 보여준다. 이 것은 <표 7>내의 1차 시나리오를 역공학 엔지니어가 개선한 시나리오로서 이를 객체 프로세스 액션 시나리오라고 부른다.

< 표 7> WorkOrder프로세스에 대한 개략 시나리오

연산 순서	대응 양식 필드	액터 객체	컨테이너 객체
System provided	Date_v	System	WorkOrder
User Keyin	WorkOrderNumber_v	User	WorkOrder
User Keyin	BillToName_v	User	WorkOrder
User Keyin	BillToAddress_v	User	WorkOrder
User Keyin	DataRequired_v	User	WorkOrder
User Keyin	Task_v1, task_v2 ...	User	WorkOrder-Task
Database Access	Cost/sqft_v1, cost/sqft_v2 ...	Database	WorkOrder-Task-Task
User Keyin	Sqft_v1, sqft_v2 ...	User	WorkOrder-Task
Computation	Amount_v1m amount_v2 ...	Unknown	WorkOrder
Computation	TotalBeforeTax_v	Unknown	WorkOrder
Database Access	TaxRate_v	Database	WorkOrder-Tax
Computation	Tax_v	Unknown	WorkOrder
Computation	Total_v	Computation	WorkOrder
User Keyin	SalesPerson_v	Unknown	WorkOrder



<그림 10> WorkOrder 양식에 대한 객체 프로세스 액션 시나리오

3.2.6 모형 통합

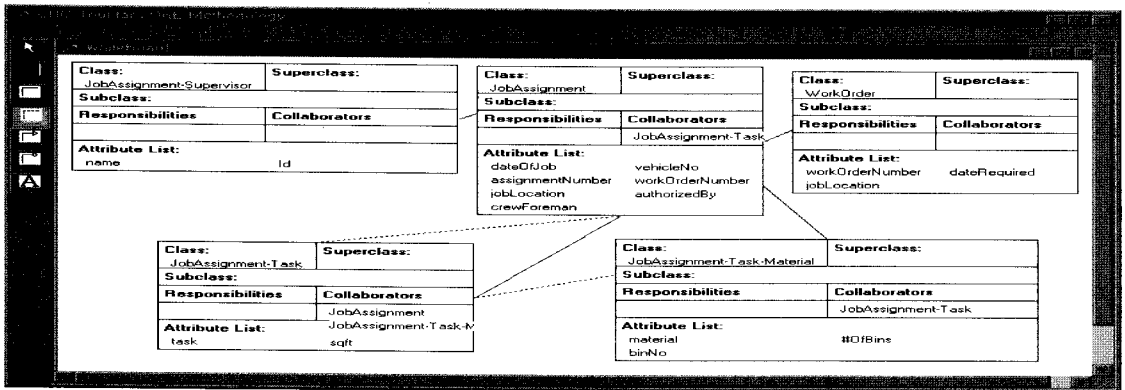
여러 양식으로부터 추출된 각 객체 구조 모형이 <표 8>의 단계에 따라 하나의 모형으로 통합되고 결과로 나온 모형을 더욱 개선한다. 보다 추상화 된 상위의 모형을 얻고자 일반화/상세화 및 컴포지션 계층(composition hierarchy)의 원리가 적용되기도 한다. 새로운 구조적 관계와 협업 관계도 추가된다. 이 과정의 목적은 여러 양식을 이용하여 새로이 생성된 모형을 기존의 객체 모형과 통합하는 것이다. 본 예에서는 두 종류의 양식을 사용했다. WorkOrder 양식은 이미 분석되어 결과가 유도되었다. JobAssignment 양식에 대한 객체 구조 모형의 결과는 <그림 11>에 제시된 것과 같다. 예에서 WorkOrder의 객체는 두 가지 양식 모두에서 유도된 공통 객체이다. 이 공통의 객체를 통해 두 모형을 서로 통합할 수 있다. 또한, Work

Order-SalesPerson과 JobAssignment-Supervisor 객체는 Employee 라는 보다 일반적인 객체의 서브 타입 클래스로 구성될 수 있다. 이와 같은 방법으로 통합된 최종 모형이 <그림 12>에 도시되어 있다.

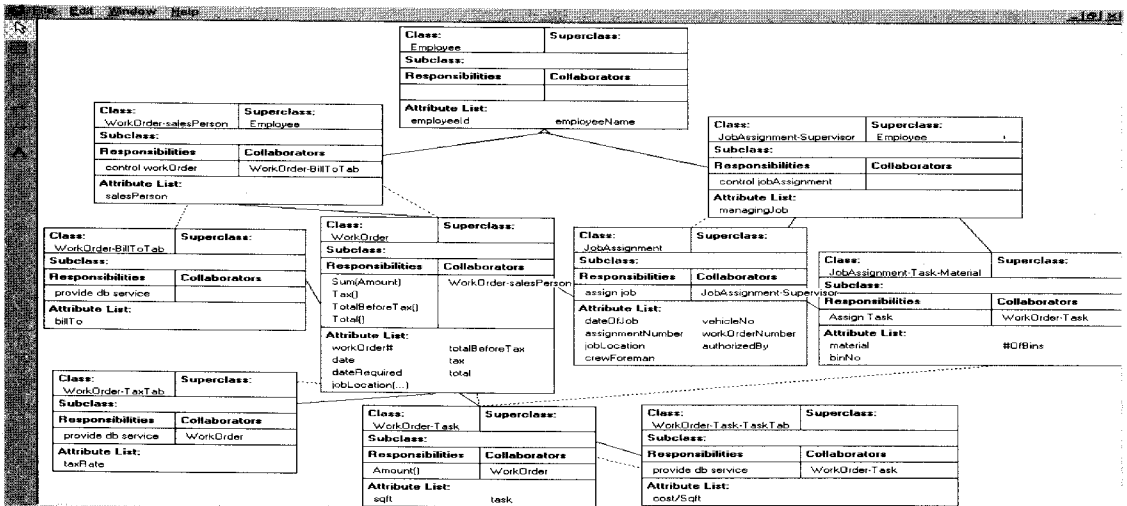
통합 모형이 완료된 후 일반화/상세화와 같은 모형화의 원리를 반복적으로 적용한다면 그 결과 모형은 보다 완전해질 것이다. 더불어 통합 과정에서는 이름 및 구조의 충돌(conflict)과 같은 현상이 발생할 수 있는데 이를 위한 대책이 강구되어야 한다. 본 방법론에서는 새로 유도되는 객체 이름을 본래의 객체 이름과 추출된 객체 또는 필드 이름을 결합하여 명명하는 특별한 명명 규칙을 사용하여 이름 충돌 현상을 방지한다. 구조 충돌의 경우에는 기존 Batini et al.의 방식을 적용하여 해결토록 한다 [Batini et al., 1992].

<표 8> 모형 통합 과정

과 정	단계별 절차	산 출 물
모형 통합	<ol style="list-style-type: none"> 업무 프로세스의 실행 순서를 기초로 여러 양식으로부터 생성된 객체 구조 모형을 선택 2진 관계에 따라 두 개의 모형씩 각 모형을 서로 통합 <ol style="list-style-type: none"> 모형간 공통 객체를 기초로 단일 모형으로 통합 객체 연산의 협업 관계를 기초로 모형을 단일 모형으로 통합 통합에 따른 갈등 해소 <ol style="list-style-type: none"> 이름 갈등 해소를 위해 동일 객체의 다른 이름들을 단일화 객체 구조 갈등 해소 최종 모형이 완성되기까지 단계1에서 단계3 반복 통합 모형 개선 	통합 객체 모형



<그림 11> Job Assignment 양식에 대한 객체 구조 모형



<그림 12> 통합 객체 구조 모형

IV. 방법론 비교

FORE 방법론과 대표적인 기존 역공학 방법론을 비교 요약하면 <표 9>와 같다. 먼저, FORE 방법론은 시스템에 대한 객체 지향 관점을 가진다. 기존 응용 시스템의 의미 있는 정보가 객체 모형에 의해 훨씬 잘 표현될 것으로 기대되는데 그 이유는 객체 지향 기술이 풍부한 표현력을 가졌을 뿐 아니라 가장 발전된 모형화 수단이기 때문이다. 반면, 기존 역공학 방법론 대부분이 데이터 또는 프로세스 어느 한 관점을 채택하고 있으며 객체 지향의 관점을 거의 적용하고 있지 않다.

기존 방법론들은 다양한 입력 원천자료를 활용하여 역공학을 수행하고 있다. 즉, 데이터베이스 스키마, 원시 프로그램 및 실행 프로그램 또는 기존 시스템의 모든 요소들을 입력 원천자료로 활용하는 등의 다양한 방법론들이 존재한다. FORE 방법론에서는 사용이 간편한 입력 원천자료를 찾고자 시도했다. 그 결과 현재 운영되는 기존 응용 시스템의 화면 양식과 시스템 동작 중에 사용자/시스템 상호작용에 의해 발생하는 정보가 주요한 입력 원천자료로 채택되었다. 이는 양식과 상호작용 정보를 활용한 최초의 역공학 연구라 할 수 있다. 이러한 정보들은 기존 응용 시스템의 동작시 항시 획득 가능한 정보들로서 이들을 얻기 위해 부

<표 9> 역공학 방법론 비교

특성 \ 방법론	Chiang, 1994	RE ² , 1993	RECAST, 1993	Saleh, 1996	REDO, 1993	FORE Methodology
시스템 관점	데이터	프로세스	데이터 및 프로세스	프로세스	객체	객체
목적	물리적 데이터베이스 시스템으로부터 데이터에 관한 의미 있는 정보 복구	기존 시스템으로부터 재사용 모듈 작성	CASE 환경에서 사용하기 적합한 형태로 산출물을 문서화	통신 소프트웨어를 유지보수하는 개발 그룹의 생산성 제고	순공학 CASE틀을 위한 입력 데이터 생성	기존 응용 시스템으로부터 데이터와 프로세스 모두를 캡슐화 한 객체를 유도
입력 원천자료	물리적 데이터베이스 스키마 및 데이터 레코드	기존 시스템의 모든 원천자료	COBOL 원시프로그램	통신 소프트웨어	COBOL 원시 코드	· 실행 중인 기존 응용 시스템의 화면 양식 · 응용 시스템과 사용자와의 상호작용 정보
목표 모형	EER 모형	공식적인 모형이 사용되지 않음	SSADM	ISO 표준 Estelle · Communicating finite state machines 모형에 기반	객체에 대한 SSADM 표현 및 기능적 기술 명세	ECRC 카드에 기반한 객체 지향 모형 및 시나리오
반	불필요	원시 프로그램의 재입력 필요	시스템 리파지토리의 활성화 필요		형식언어 Z를 이용한 표현 필요	불필요
가정 및 제한	· 3단계 정규화 · 필드 명을 기반으로 내포 종속성 생성(synonms/homonyms 파악)	· 대상 기준이 추상 데이터 타입을 구현한 재사용 모듈을 자동적으로 생성해 주지 못함 · 사람의 판여 요구	특정한 원시 프로그램과 환경에 의존	역공학 접근과 설계자 입력 및 검토회의 방식의 결합	유지 보수하는 사람, 분석가 및 사용자들을 위한 상위 수준의 이해 용이한 명세를 얻는데 구조적 표현이 중요한 수단	사용자의 추론에 의해 객체의 내부 처리규칙을 파악
사람의 관여	필요함	필요함	필요함	프로토콜 설계자에 의한 많은 지원이 필요	유지보수 엔지니어에 의한 많은 지원이 필요	역공학 엔지니어의 관여가 필요
공식적 모형과의 매핑을 위한 매카니즘	제공되지 않음	제공되지 않음	변환 규칙 제공	제공되지 않음	공식언어 Z와의 매핑	UML 매핑
자동화된 지원도구	지원 안됨	Prolog 프로그램 터서너리	PSL/PSA 리파지토리	반자동 Estelle 기반틀	REDO 툴킷	부분적으로 지원

가적으로 각종 원천자료들을 재입력하거나 선행 처리하는 작업이 필요하지 않다. 많은 응용 시스템들이 매우 오래 전에 개발되어 운영되고 있기 때문에 실행 가능한 프로그램 파일 외에는 어떠한 정보도 존재하지 않는 경우가 많음을 고려할 때 이는 가용성 측면에서 유리하다 하겠다.

기존 역공학 방법론에서는 복구된 의미 있는 정보나 설계 정보들을 표현하기 위한 목표 모형으로서 ERD, DFD 또는 프로그램 구조도와 같은 전통적인 모형화 표현들을 일반적으로 이용하고 있다. 그러나 전통적인 모형에서 데이터 관점과 프로세스 관점은 각기 독립적으로 추상화를 처리하기 때문에 시스템 전체의 관점에서 보면 불완전하다. 객체 지향 기술은 기존의 어떠한 다른 모형이나 기술에 비해 우수하다고 알려져 있다. 더욱이 시스템에 대해 객체 관점을 취하고 있으므로 객체 모형을 목표 개념적 모형으로 채택하는 것이 자연스럽다. FORE 방법론은 CRC 카드 기반의 객체 지향 모형을 활용한다. CRC 카드는 사용하기 편리한 객체 지향 모형화 방법이므로 복구된 의미 있는 정보를 객체 지향 모형으로 용이하게 표현할 수 있을 것이다. 또한, 객체의 행위를 표현하기 위해 시나리오가 도입되었다.

Chiang의 방법론 외에는 원시 프로그램의 재입력, 리파지토리 정보의 활성화(population) 또는 정규화(normalization)와 같은 입력 원천자료의 선행 처리를 요구한다. 이에 비해 FORE 방법론은 기존 시스템의 실제 실행 중에 발생하는 정보를 수집하여 활용하므로 선행 처리 작업이 필요하지 않다. 많은 기존 응용 시스템들이 실행 프로그램만을 가지고 있는 상황을 고려하면 FORE 방법론이 보다 자연스럽다고 할 수 있다.

역공학의 현 기술 수준을 고려한다면 아직 완벽한 방법론은 존재하지 않는다고 할 수 있다. 따라서 각 방법론은 그 나름의 제한점이나 가정을 내포하고 있다. FORE 방법론에서는 객

체 메소드의 내부 처리규칙을 역공학 엔지니어가 양식 필드의 이름을 이용하여 추론해야 한다는 제한점을 가진다. 왜냐하면 FORE 방법론에서는 원시 프로그램 자체를 입력 원천자료로 고려하지 않기 때문이며 또한 FORE의 첫번째 과정에서 처리규칙과 관련된 정보를 수집할 수 없기 때문이다.

역공학 방법론에서는 예외 없이 사람의 관여를 요구한다. FORE 방법론에서도 객체 메소드의 내부 처리규칙을 추론함에 있어 역공학 엔지니어가 주요한 역할을 수행한다. 역공학의 고유한 속성상 사람의 관여는 불가피하다고 한다. 그러나 역공학 기술 수준이 향상될 수록 사람의 관여는 점차 줄어들 것으로 기대된다.

Chiang의 방법론 외에는 모든 방법론들이 지원 툴을 제공한다. 더욱이 RE2 [Canfora, 1993]과 RECAST [Edwards et al., 1993]는 복구된 의미 있는 정보를 자신의 리파지토리에 저장 활용한다. FORE 방법론을 지원하는 프로토타입 시스템은 개발 중에 있다. 이 프로토타입 시스템은 두 개의 서브 시스템으로 구성된다. 하나는 양식 및 상호작용 정보들을 수집하는 서브 시스템(AFCS, Automatic Form Capturing Subsystem)이며, 다른 하나는 역공학의 모형화를 지원하는 서브 시스템(RESS, Reverse Engineering Support Subsystem)이다. 현재는 RECAST와 REDO의 지원 툴이 FORE의 프로토타입 시스템에 비해 보다 강력한 기능을 갖는 것으로 판단된다. 향후 FORE의 지원 툴이 보다 향상된 기능을 포함하도록 개발하여야 할 것이다.

V. 결 론

본 논문에서는 기존 응용 시스템의 환경으로부터 개념적 수준에서의 의미 있는 정보를 복구하기 위한 FORE 방법론을 제안했다. FORE 방법론은 양식의 구조 및 시스템과 사용자 간의 상호작용 정보들을 활용하여 객체 지향 개

념적 모형을 복구한다. 기존 응용 시스템에 관한 어떠한 정보도 존재하고 있지 않을 지라도 사용자에게 의해 응용 시스템이 동작된다면 FORE 방법론은 분석에 필요한 정보를 수집할 수 있으며 실제 시스템을 역공학할 수 있다. 양식에 관한 지식이라고 불리는 기존 응용 시스템의 동작 상태 정보를 자동적으로 수집하기 위하여 지원 서브시스템을 개발하여 활용하고자 한다. 간단한 예에 적용한 결과 FORE 방법론이 성공적임을 확인할 수 있었다. 적용 시 예제 양식에 관한 지식은 개발 예정인 지원 도구를 통해 수집 가능한 정보로 가정하고 자료로 만들어 활용했다. 현재 역공학 전체 과정을 지원하기 위한 일종의 역공학 CASE 도구를 개발 중에 있다.

본 연구의 기여 사항을 정리하면 다음과 같다. 우선, 입력 원천자료로서 양식과 사용자/시스템 상호작용에 의해 발생하는 정보를 처음으로 활용하여 역공학 연구를 시도했다는 점을 들 수 있다. 다음은 양식에 관한 지식을 자동적으로 수집할 수 있는 방안을 알고리즘화 된 절차로서 제시했으며 현재 그 지원 도구를 개발

중에 있다. 다음으로는 데이터와 프로세스를 모두 복구하려고 시도했으며 이를 캡슐화 한 객체 모형을 적용했다. 마지막으로 기존에 잘 알려진 예에 본 방법론을 시험적으로 적용해 보았다. 조직의 기존 응용 시스템에 대한 유지보수뿐 아니라 다양한 영역의 기존 응용 시스템들을 이전하거나 리엔지니어링 할 경우에 FORE 방법론에 의한 접근 방법이 두루 적용될 수 있을 것으로 기대된다.

본 연구를 좀 더 보완하기 위해서는 다음의 연구가 필요하다. 첫째, 기존 응용 시스템으로부터 보다 다양한 종류의 지식을 발견하여 보다 완전하게 기존 시스템의 의미 있는 지식 정보를 복구할 수 있는 방안을 마련하는 것이다. 둘째, FORE 방법론에서 제시된 과정을 보다 자동화하여 역공학 엔지니어나 설계자를 편리하게 지원할 수 있는 도구의 개발이 필요하다. 셋째, 본 방법론을 보다 큰 규모의 실제 사례에 적용하여 방법론을 검증해 보는 것이 요구된다. 넷째, 메소드의 내부 처리규칙을 보다 완벽하게 복구할 수 있는 연구가 필요하다.

〈참 고 문 헌〉

- [1] Aiken, P., A. Muntz and R. Richards, A framework for reverse engineering DoD legacy information systems, *Proceedings Working Conference on Reverse Engineering*, Baltimore, Maryland, May 2123, 1993, pp. 180-191.
- [2] Azar, N. and E. Pichat, Translation of an extended Entity-Relationship model into the universal relation with inclusions formalism in: S. Spaccapietra, ed., *Proc. 5th Int. Conf. On the Entity-Relationship Approach*, IEEE Computer Society Press, Silver Spring, 1986, pp. 253-270.
- [3] Bailin, S. C., An object-oriented requirements specification method, *Communications of the ACM*, Vol. 32, No. 5, May 1989, pp. 608-623.
- [4] Batini, C., B. Demo and A. Di Leva, A Methodology for Conceptual Design of Office Databases, *Information Systems*, Vo. 9, No. 3/4, 1984, pp. 251-263.
- [5] Batini, C., S. Ceri and S. B. Navathe, *Conceptual Database Design An Entity Relationship Approach*, Benjamin/Cummings, Red-

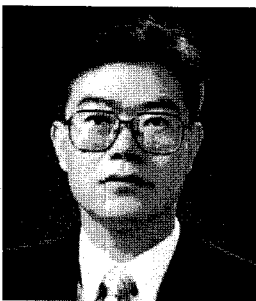
- wood City, 1992.
- [6] Bertiss, A. T. and SYSLAB, Reverse Engineering, Reengineering, And Concurrent Engineering of Software, *International Journal of Software Engineering and Knowledge Engineering*, Vo. 5, No. 2, 1995, pp. 299-324.
- [7] Blaha, M. R., W. J. Premerlani and J. E. Rumbaugh, Relational database design using an object-oriented methodology, *Communications of the ACM*, Vo. 31, No. 4, April 1988.
- [8] Booch, G. and J. Rumbaugh, *Unified Method for Object-oriented Development Document Set Version 0.8*, Rational Software Corporation, 1995.
- [9] Booch, G., *Object Oriented Design with Applications*. Benjamin/Cummings Publishing Company, Inc., 1991.
- [10] Boulanger, D. and S. March, An approach to analyzing the information content of existing databases, *Data Base*, Vol.20, No. 2, 1989, pp. 1-8.
- [11] Canfora, G. A. Cimitile and M. Munro, A Reverse Engineering Method for Identifying Reusable Abstract Data Types, *Proceedings Working Conference on Reverse Engineering*, Baltimore, Maryland, May 2123, 1993, pp. 73-82.
- [12] Chen, P. P., The Entity-Relationship model: towards a unified view of data, *ACM Trans. Database Syst.*, 1, 1976, pp. 9-36.
- [13] Chen, P. P., *Database design based on entity and relationship*, in: S.Yao, ed., Principles of Database Design, Prentice-Hall, Englewood Cliffs, 1985, pp. 174-210.
- [14] Chiang, R. H. L., T. M. Barron and V. C. Storey, Reverse Engineering of Relational Database : Extraction of an EER Model from a Relational Database, *Data & Knowledge Engineering*, 12, 1994, pp. 107-142.
- [15] Chiang, R. H. L., T. M. Barron and V. C. Storey, Performance evaluation of reverse engineering relational database into extended Entity-Relationship Models, in: R. Elmasri et al., eds., *Proc. 12th Int. Conf. On the Entity-Relationship Approach ER93*, Springer LNCS 823, Berlin, 1994, pp. 352-363.
- [16] Chikofsky, E. J. and J. H. Cross II, Reverse Engineering and Design Recovery : A Taxonomy, *IEEE Software*, January 1990, pp. 13-17.
- [17] Choobineh, J., M. Mannino, J. F. Nunamaker and B. R. Konsynski, An Expert Database Design System Based on Analysis of Forms, *IEEE transactions on Software Engineering*, Vol. 14, No. 2, February 1988.
- [18] Choobineh, J., M. Mannino and V. Tseng, A Form-Based Approach for Database Analysis and Design, *Communications of the ACM*, Vol. 35, No. 2, February 1992, pp. 108-120.
- [19] Cutillo, F., P. Fiore and G. Visaggio, Identification and Extraction of Domain Independent Components in Large Programs, *Proceedings Working Conference on Reverse Engineering*, Baltimore, Maryland, May 2123, 1993, pp. 83-92.
- [20] Edwards, H. M. and M. Munro, Reverse Engineering from COBOL to SSADM Specification, *Proceedings Working Conference on Reverse Engineering*, Baltimore, Maryland, May 2123, 1993, pp. 44-53.
- [21] Jacobson, I., *Object-Oriented Software Engineering-A use Case Driven Approach*, Addison-Wesley, 1995.
- [22] Kantola, M., H. Mannila, K-J. Raiha and

- H. Siirtola, Discovering Functional and Inclusion Dependencies in Relational Databases, *International Journal of Intelligent Systems*, 7, 1992, pp. 591-607.
- [23] Kim, I-K. and J-K. Lee, *Expert System for Conceptual Database Design*, MS Thesis, Department of Management Information Systems, KAIST, Korea, 1994.
- [24] Kim, K-H. and Y-G. Kim, Process reverse engineering for BPR: A form-based approach, *Information & Management*, 33, 1998, pp. 187-200.
- [25] Kim, Y-G., Improving Legacy Systems Maintainability, *Information Systems Management*, Winter 1997, pp. 7-11.
- [26] Lano, K. and H. Haughton, Integrating Formal and Structured Methods in Reverse Engineering, *Proceedings Working Conference on Reverse Engineering*, Baltimore, Maryland, May 21-23, 1993, pp. 17-26.
- [27] Lee, H., H. Bae and C. Yoo, A Methodology for Deriving an Object Model by Using Structured Analysis Results, *Journal of the Korean Operations Research and Management Science Society*, Vol. 21, No. 3, December 1996, pp. 175-195.
- [28] Lee, H., C. Yoo, C. Lee, Y. Kim, J. Kim and S. Cho, Developing Intranet Hypermedia System Using Scenario-Based Object-Oriented Technique, *Korean Management Science Review*, The Korean OR/MS Society, Vo. 14, No. 2, November 1997, pp. 113-137.
- [29] Lee, H., C. Lee and C. Yoo, A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Applications, *31st Proceedings of Hawaii International Conference on System Sciences*, January 1998.
- [30] Lee, B-Y., *A Maintenance Support Expert System for Legacy Systems: UNIK-METASOFT*, Ph D Thesis, Graduate School of Management, KAIST, November 1996.
- [31] Markowitz, V. M. and J. A. Makowsky, Identifying extended Entity-Relationship object structures in relational schemes, *IEEE Transactions on Software Engineering*, 16, 1990, pp. 777-790.
- [32] Martin, J., *Information Engineering I, II, III*, Addison-Wesley, 1990.
- [33] Meta Systems Ltd., *PSL/PSA Reference manuals*, Meta Systems Ltd., Ann Arbor, Michigan, 1987.
- [34] Navathe, S. B. and A. M. Awong, Abstracting relational and hierarchical data with a semantic data model, in: S. T. March, ed., *Proc. 6th Int. Conf. on the Entity-Relationship Approach*, North-Holland, Amsterdam, 1987, pp. 305-336.
- [35] Ning J. Q., A. Engberts and W. Kozaczynski, Automated Support for Legacy Code Understanding, *Communications of the ACM*, Vol. 37, No. 5, May 1994, pp. 50-57.
- [36] Premerlani, W. J. and M. R. Blaha, An Approach for Reverse Engineering of Relational Databases, *Communications of the ACM*, Vol. 37, No. 5, May 1994, pp. 42-49.
- [37] Pyle D. R., Joseph M., *Entity-Relationship Models Expressed in Z: A Synthesis of Structured and Formal Methods*, Oxford University Programming Research Group, 1991.
- [38] Quatrani, T., *Visual Modeling with Rational Rose and UML*, Addison-Wesley, 1998.
- [39] Rational Software Corp., *Introduction to Rational Rose/C++ using UML professional services*, Rational Software Corporation,

- Corporate Headquarters, 2800 SanTomas Expressway, Santa Clara, CA 95-051-0951, 1996.
- [40] Rumbaugh, J., M. R. Blaha, W. Premerlani, F. Eddy and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall, New York, 1991.
- [41] Saleh, K. and A. Boujarwah, Communications software reverse engineering: a semi-automatic approach, *Information and Software Technology*, 38, 1996, pp. 379-390.
- [42] Shoval, P. and N. Shreiber, Database Reverse Engineering: From the Relational to the Binary Relationship Model, *Data & Knowledge Engineering*, 10, 1993, pp. 293-315.
- [43] Shu, N. C., V. Y. Jum and et al., Specification of Forms Processing and Business Procedures for Office Automation, *IEEE Transactions on Software Engineering*, Vol. 8, No. 5, September 1982, pp. 499-512.
- [44] Silva, M. J. V. and C. R. Carlson, MOODD, a method for object-oriented database design, *Data & Knowledge Engineering*, 17, 1995, pp. 159-181.
- [45] Tschritzis, D., Form Management, *Communications of the ACM*, Vol .25, No. 7, July 1982, pp. 453-478.
- [46] Umar, A., *Application(RE) Engineering Building Web-Based Applications and Dealing with Legacies*, Prentice Hall PTR, 1997.
- [47] Waters, R. C. and E. Chikofsky, Reverse Engineering: Progress Along Many Dimensions, *Communications of the ACM*, Vol. 37, No. 5, May 1994, pp. 23-24.
- [48] Wilkinson, N. M., *Using CRC Cards An Informal Approach to Object-Oriented Development*, SIGs Books, 71 W. 23rd Street, Third Floor, New York, New York 10010, 1995.
- [49] Yourdon, E. and L. Constantine, *Structured Design*, Yourdon Press, New York, 1978.
- [50] Yourdon, E., *Modern Structured Analysis*, Prentice-Hall International Editions (1989), A Division of Simon & Schuster, Englewood Cliffs, NJ 07632.

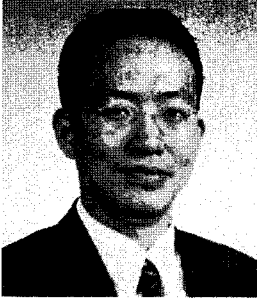
◆ 이 논문은 1998년 10월 13일 접수하여 1차 수정을 거쳐 1999년 3월 10일 게재 확정되었습니다.

◆ 저자소개 ◆



유천수 (Yoo, Cheon Soo)

국방과학연구소 선임연구원으로 근무하고 있으며 경북대학교 전자공학사 (1982), 홍익대학교 전자계산학 석사 (1984), KAIST 테크노경영대학원 경영공학 전공 박사 과정 재학 중 (1994 - 현재) 이다. 주요 관심 분야는 기업 데이터베이스 설계, 역공학 및 객체지향 방법론 등이다.



이희석 (Lee, Hee Seok)

서울대학교 산업공학 학사, KAIST산업공학 석사, 그리고University of Arizona에서 경영정보 시스템을 전공하여 경영학 박사학위를 취득하였으며, 현재 KAIST 테크노경영대학원 부교수로 재직하고 있다. 1992년에서 1994년 까지 네브라스카 주립대에서 경영정보 분야 연구 및 강의를 하였다. 주요 연구 관심 분야는 기업 정보 시스템, 데이터베이스, 클라이언트/서버, 분산 객체, 통신경영, 정보 전략 계획 수립 등이다.