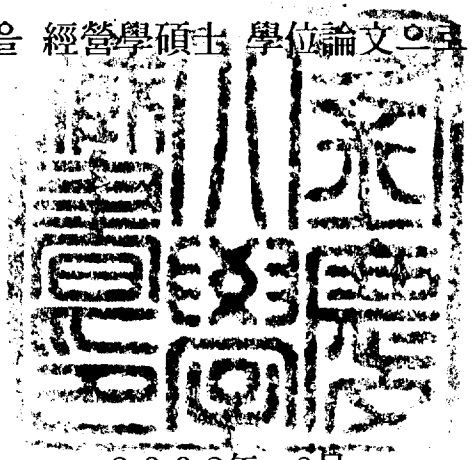
經營學碩士 學位論文

프로세스 리파지토리를 위한 스키마 설계에 관한 연구

指導敎授 李 在 楨

이 論文을 經營學碩士 學位論文으로 提出함



2002年 2月

釜慶大學校 大學院

經 營 學 科

鄭 湖 錫

鄭湖錫의 經營學碩士 學位論文을 認准함

2001年 12月 26日

主　審　經營學博士　魚 允 洋　

委　員　經營學博士　金 河 均　

委　員　經營學博士　李 在 楨　

차 례

제 1 장 서 론 1

1.1 연구 배경 1

1.2 연구 목적 2

1.3 연구의 방법 및 범위 3

1.4 논문의 구성 4

제 2 장 이론적 배경 5

2.1 비즈니스 리엔지니어링 5

2.1.1 프로세스의 개념 5

2.1.2 비즈니스 리엔지니어링의 개념 7

2.1.3 비즈니스 리엔지니어링의 추진절차 9

2.1.4 리엔지니어링과 벤치마킹 11

2.2 프로세스 모델 13

2.2.1 프로세스 모델의 개념 13

2.2.2 IDEF0 모델링 방법론 15

2.3 프로세스 리파지토리 20

2.3.1 프로세스 리파지토리의 개념 20

2.3.2 프로세스 리파지토리 관련 선행연구 22

2.3.3 선행 연구의 한계 24

제 3 장 스키마 설계 26

3.1 본 연구의 프로세스 모델 27

3.2 개념적 스키마(Conceptual Schema) 28

3.3 내부 스키마(Internal Schema) 34

제 4 장 사례연구 39
4.1 샘플 프로세스 39
4.2 프로세스 데이터입력 40
4.3 프로세스 분석 45

제 5 장 결론 51
5.1 연구의 요약 51
5.2 연구의 의의 51
5.3 연구의 한계 및 향후 연구 과제 52

참 고 문 헌 53

표 차 례

〈표. 1〉 프로세스의 정의 5
〈표. 2〉 리엔지니어링에 대한 정의 7
〈표. 3〉 프로세스 개선과 프로세스 혁신과의 차이 9
〈표. 4〉 벤치마킹의 정의 11
〈표. 5〉 프로세스 구성요소 분류 14
〈표. 6〉 프로세스 모델링의 4가지 관점 14
〈표. 7〉 IDEF0 프로세스 다이어그램으로부터의 개체 도출 29
〈표. 8〉 개체와 그 속성 29
〈표. 9〉 데이터 정의어(DDL) 34
〈표. 10〉 activity 테이블 36
〈표. 11〉 IO 테이블 36
〈표. 12〉 IO_List 테이블 37
〈표. 13〉 control 테이블 37
〈표. 14〉 control_list 테이블 37
〈표. 15〉 mechanism 테이블 37
〈표. 16〉 mechanism_list 테이블 38
〈표. 17〉 구현환경 40

그림차례

〈그림. 1〉 조직에서의 프로세스 흐름 6
〈그림. 2〉 리엔지니어링 추진절차 10
〈그림. 3〉 IDEF0의 ICOM 16
〈그림. 4〉 IDEF0의 탑다운(top-down) 분해 17
〈그림. 5〉 IDEF0 모델링의 예. 1 18
〈그림. 6〉 IDEF0 모델링의 예. 2 19
〈그림. 7〉 IDEF0 모델링의 예. 3 19
〈그림. 8〉 IDEF0 모델링의 예. 4 20
〈그림. 9〉 데이터베이스 스키마 설계 단계 26
〈그림. 10〉 개체관계도 (1) 31
〈그림. 11〉 개체관계도(2) 33
〈그림. 12〉 주문획득 활동의 IDEF0 모델 40
〈그림. 13〉 활동(activity) 모듈 41
〈그림. 14〉 IO 모듈 42
〈그림. 15〉 IO_List 모듈 42
〈그림. 16〉 컨트럴 모듈 43
〈그림. 17〉 컨트럴 리스트 모듈 43
〈그림. 18〉 매커니즘 모듈 44
〈그림. 19〉 매커니즘 리스트 모듈 44
〈그림. 20〉 질의 예. 1 45
〈그림. 21〉 질의 예. 2 46
〈그림. 22〉 질의 예. 3 47
〈그림. 23〉 질의 예. 4 48
〈그림. 24〉 질의 예. 5 48
〈그림. 25〉 질의 예. 6 49
〈그림. 26〉 질의 예. 7 50

A Study of Schema Design for Process Repository

Chung, Hoe-Suk

Department of Business Administration, Graduate School,
Pukyong National University

Abstract

Process repository is consistent, easy-to-use collections of process knowledge, used for multiple purposes. In most existing process repository approaches, only unstructured data, like documents and images, can be managed. That means, process model which is represented as unstructured data is not easy to comapre and analyze.

The purpose of this study is to propose the schema for process repository which can overcome the limitation of existing process repositories.

This research project describes the background of process repository, which are concept of BPR , Benchmarking, process repository and limitation of existing process repository and it designed the schema for process repository and created the tables for relational database. In order to validate the schema of this study, it examined the process model which was stored in relational database using SQL.

The contributions of this study are as follows;

First, it designed the schema for process repository to store process models.

Second, the process model which stored in relational database in the type of structured data can improve the performance of process analysis.

Finally, using IDEF0 for process modeling technique, the hierarchical structure

of process can be represented. And it makes it possible to store large and complex process in relational database.

제 1 장 서 론

1.1 연구 배경

오늘날의 기업환경은 과거의 어느 때보다도 급격하게 변화하고 있다. 이에 따라 기업의 경영방식도 많은 변화를 요구하고 있다. 지금까지의 경영방식은 날마다 접하게 되는 일상적인 문제해결에는 도움을 주었지만 기업의 근본적인 문제들을 다루는 데는 어쩔 수 없는 한계를 가지고 있다. 최근의 경영 여건은 이러한 문제들에 대한 근본적인 해결방안을 요구하고 있다.

과거에는 크게 중시하지 않았던 시간이 새로운 경쟁요인으로 떠오르고, 고객만족의 중요성은 단순한 생산성의 증대보다 더 중요한 위치를 차지하게 되어, 소극적인 경영개선 정도로는 대처하기가 힘들게 되었다. 이제는 완전히 다시 시작해야 하고 현재의 업무 수행방식을 고객의 관점에서 전면적으로 뜯어고치지 않으면 안 된다. 기업들의 경영 목표는 고객과 그들의 만족에 있으며 기업들은 고객의 관점에서 비즈니스 프로세스를 다시 보게된 것이다[Hammer, 1997]. 비즈니스 리엔지니어링(Business Reengineering)이나 벤치마킹, ABC 등의 기법들은 기업에서의 이러한 혁신을 수행하려는 노력이라고 할 수 있다[Nyamekye, 2000].

비즈니스 리엔지니어링 추진절차에서 벤치마킹은 유효한 수단으로 인정받고 있다[Harrison & Pratt, 1992; Chang, 1994; Furey, 1993, 신홍철, 1994]. 벤치마킹 기법은 업계에서 가장 우수한 성과를 내는 모범기업의 선진 프로세스(Best Practice)를 찾아내어 이들의 성과를 기준으로 목표를 설정하도록 해 주는 역할을 한다[Camp, 1989].

최근 들어 이러한 벤치마킹 활동을 시스템적으로 지원하려는 노력이 이루어지고 있다. 즉, 여러 조직의 선진 프로세스(Best Practice)를 수집·분류하여 이를 시스템에 저장함으로서 리엔지니어링을 위해 필요한 프로세스 관련 정보를 효과적으로 활용하고자 하는 프로세스 리파지토리 연구가 여러 기관에서 활발히 진행

되고 있는 것이다[Carr, 1999]. 그 대표적인 예가 MIT의 온라인 프로세스 핸드북 리파지토리(Online Process Handbook Repository)이다. 프로세스 핸드북 리파지토리는 동일한 목적의 비즈니스 프로세스를 업무 수행방식에 따라 다양하게 정의해 놓은 전자적인 형태의 참고서이다[Malone, 1993].

MIT에서 프로세스 리파지토리 연구가 처음 시작된 이후 주로 연구소나 공공단체에서 공익을 목적으로 프로세스 리파지토리 연구가 진행되고 있다. 그러나 이러한 프로세스 리파지토리들은 대부분 참조 모델을 수집하고 체계적으로 분류하는데 급급할 뿐, 어떠한 방식으로 저장하여 효율적으로 활용할 것인가에 관한 연구는 부진하였다.

기존의 프로세스 리파지토리들은 문서 또는 다이어그램 같은 비구조적인 정보로 표현된 프로세스 모델들을 보유하고 있는데, 이러한 비구조적인 정보는 그 활용에 큰 어려움이 있고, 필요에 따라 변경하기가 어렵다는 단점이 있다. 이미지나 문서와 같은 비구조적 데이터로 존재하는 프로세스 모델의 내용을 체계적으로 분석하고 활용하려면, 구조적 데이터로 데이터베이스에 저장할 필요가 있는 것이다. 최근의 국내 연구[김훈태, 2001]에서는 이러한 한계를 극복하기 위한 시도가 있었으나 프로세스 모델의 계층적 분해가 네단계로 한정되어 있다는 한계를 가지고 있다.

1.2 연구 목적

프로세스 리파지토리는 선진 프로세스(Best Practice)를 체계적으로 저장하여 이를 필요에 따라서 활용하려는 목적을 가지고 있으며 프로세스 리엔지니어링 및 벤치마킹을 지원하기 위한 종합적 지원도구의 성격을 가진다.

프로세스 리파지토리 관련 선행연구들은 대부분 다이어그램으로 표현된 비즈니스 프로세스를 이미지 파일 또는 문서 파일과 같은 비구조적 데이터의 형태로 프로세스 모델들을 보유하고 있다. 그러나 이렇게 표현된 프로세스는 필요에 따라 변경하기가 용이하지 않다.

그리고 이미지 또는 문서와 같은 비구조적 데이터로 저장된 프로세스 모델들은 프로세스들을 비교·분석하기가 용이하지 않다는 단점이 있다. 기존의 연구들은 프로세스를 표현하기 위한 나름대로의 모델링 기법은 가지고 있지만, 정작 이를 비교·분석하여 벤치마킹 수행시 활용할 수 있는 방안제시는 못하고 있다.

본 연구의 목적은 이러한 선행연구들의 한계를 극복할 수 있는 프로세스 리파지토리를 위한 스키마를 제시하는 것이다.

기존 프로세스 리파지토리 연구들은 문서 또는 이미지와 같은 비구조적 데이터로 프로세스 모델을 저장하기 때문에 프로세스를 비교·분석할 수 있는 기능이 부족하다. 따라서 본 연구에서는 기존 연구의 이러한 한계를 극복하기 위하여 프로세스 모델을 구조적 데이터로 관계형 데이터베이스에 저장하기 위한 스키마를 제시한다. 이를 통해 자사의 비즈니스 프로세스를 분석하고, 타 기업의 그것과 비교할 수 있는 향상된 기능이 제공되어 진다.

그리고 프로세스 리파지토리를 위한 프로세스 모델은 프로세스의 성격에 따라서 매우 커질 수 있으므로 계층적 분해를 지원해야만 한다[Jarzabek, Ling, 1995]. 본 연구에서는 이러한 목적을 달성하기 위하여 프로세스를 계층적으로 분해하여 표현하는 기능을 가진 IDEF0 모델에 근거하여 스키마[1]를 설계하였다.

1.3 연구의 방법 및 범위

본 연구에서는 다음과 같은 방법으로 본 연구의 목적을 달성하였다. 먼저 본 연구의 시발점이 된 비즈니스 리엔지니어링과 프로세스 리파지토리의의 개념, 그리고 기존 프로세스 리파지토리의 성격 및 한계에 대하여 고찰하였다. 그리고 본 연구의 가장 핵심인 프로세스 리파지토리를 위한 스키마를 설계하여 관계형 데이터베이스에 테이블을 생성하였다. 마지막으로 프로세스 모델의 사례를 통하여 관계형 데이터베이스에 구조적 데이터의 형태로 저장된, 프로세스 모델이 어떻게 활용

---

1) 스키마란 데이터베이스의 논리적 기술, 즉 전체적인 데이터베이스의 구조를 기술한 것이다. 그리고 요구사항 조사로부터 최종적으로 DBMS에 구현할 수 있는 스키마 정의까지의 모든 활동을 스키마 설계라고 한다[Shamkant, 1992].

될 수 있는지 SQL을 이용하여 실험하였다.

연구의 범위는 프로세스 모델의 저장 및 활용의 핵심이 되는 스키마 설계와 테이블 생성으로 한정하였으며, GUI 환경에서 프로세스 모델을 다이어그램 형태로 편집·수정·조회하기 위한 자동화 도구의 개발은 연구의 범위에서 제외하였다.

1.4 논문의 구성

본 논문은 서론, 이론적 배경, 스키마 설계, 사례연구 그리고 결론의 다섯 개 장으로 구성되어 있다.

제 2장에서는 본 연구의 이론적 배경인 비즈니스 리엔지니어링, 벤치마킹, 프로세스 모델, 그리고 프로세스 리파지토리의 개념 및 그 특징을 살펴보았다.

제 3장에서는 IDEF0 프로세스 모델링 방법론을 바탕으로 프로세스 리파지토리를 위한 스키마를 제시하였다.

제 4장에서는 본 연구에서 제시된 스키마와 테이블들이 어떻게 활용될 수 있는지 프로세스 모델의 사례를 가지고 스키마의 타당성을 실험하였다.

마지막으로 제 5장에서는 본 논문에서 논의된 사항들을 요약하고 연구의 기여와 한계 및 향후 연구과제를 제시하였다.

제 2 장 이론적 배경

본 장에서는 프로세스 리파지토리와 관련된 비즈니스 리엔지니어링, 벤치마킹, 프로세스 모델에 대한 개념 및 그 특징을 살펴보고, 프로세스 리파지토리의 선행 연구들을 고찰한다.

2.1 비즈니스 리엔지니어링

2.1.1 프로세스의 개념

비즈니스 리엔지니어링의 근본대상은 프로세스이다[Hammer, 1990; Davenport, 1990; Vantrappen, 1992; Chang, 1994]. 기업에서의 주요 업무는 한 부서 독단으로 이루어지는 경우는 거의 없으며, 여러 부서의 협력을 통하여 이루어진다. 따라서 한 부서의 부문별 최적화보다는 업무의 수행단위인 프로세스를 대상으로 혁신을 이룩하여야 전사적인 최적화를 이룩할 수 있다[Maelao, Pidd, 2000].

프로세스적 접근방식을 파악하기 위해서는 먼저 프로세스의 정의를 살펴볼 필요가 있다. 〈표. 1〉은 프로세스에 대한 대표적인 정의들이다.

〈표. 1〉 프로세스의 정의

정의자	정의
Davenport [1990]	비즈니스의 결과물(outcome)을 획득하기 위해 수행되는 논리적으로 연관된 과업들의 집합[2]
Hammer & Champy [1993]	하나 이상의 입력을 받아들여 고객에게 가치 있는 결과를 산출하기 위한 활동들의 집합[3]
Johansson [1993]	입력을 받아들여 변형시키고 결과를 산출하는 연결된 활동들의 집합[4]

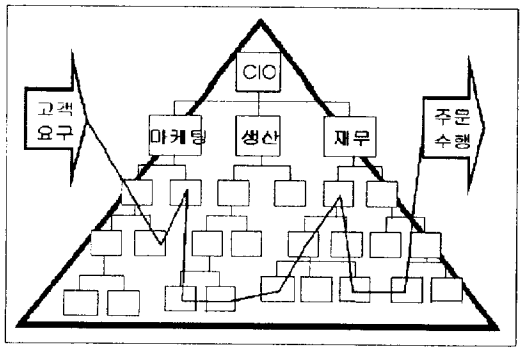
2) a set of logically related tasks performed to achieve a defined business outcome
3) a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer
4) a set of linked activities that take an input and transform it to create an output

즉, 프로세스란 기업 내부 또는 외부의 고객에게 가치를 전달하기 위한 시작에서 끝까지의 과정을 의미한다. 예를 들어 신제품을 개발할 때에는 단순히 디자인 부서에만 국한된 업무가 아닌 제조와 판매 부서까지의 연계를 필요로 한다. 그것은 신제품을 개발하기 위해서는 고객의 요구 사항을 제일 먼저 접하는 판매 부서의 피드백을 받아야 하며 실제 제조될 수 있는 디자인이 되기 위해서는 제조시의 요구 사항이 검토되어야 함을 의미한다[Hunt, 1996].

프로세스는 과업과 다르다. 과업(task)이란 일의 한 부분으로 한 사람에 의해 수행되는 비즈니스 활동이며, 프로세스는 관련된 과업의 그룹 전체로 고객에 대한 가치를 창조하는 것이다. 예를 들어, 주문처리는 재화의 형태로 고객에게 전달되는 가치를 창조하는 프로세스이다. 이는 다음과 같은 매우 많은 과업으로 구성되어 있다. 즉, 고객에게서 주문을 받고, 이를 컴퓨터에 입력하여 고객의 신용을 점검하고, 생산계획을 수립하며, 재고를 할당하고, 수송방법을 선택하며, 제품을 포장하고, 발송하는 것 등이다. 누구도 제품이 적재되기 전까지는 수송할 수 없고, 물건을 가져오기 전까지는 포장할 수 없다. 오직 이런 과업들이 전체로 통합될 때 개별적 활동들이 가치를 창조한다. 다시 말해서 고객에게 가치를 전달하기 위해 전체 프로세스의 흐름을 대상으로 하는 경영혁신이 필요한 것이다[Hammer, 97].

Davenport는 프로세스의 두 가지 특성을 지적하였다. 첫째는 프로세스는 내부 또는 외부의 고객을 가진다는 것이고, 둘째는 프로세스는 조직내 또는 조직간의 하부조직을 가로지른다는 것이다[Davenport, 1993]. 즉 〈그림. 1〉 에서 보이는 바와 같이 프로세스는 조직을 기능 교차적(cross functional)으로 가로지르는 특징을 가지며, 프로세스의 시작과 끝은 고객이다[Grover, Malhotra 1997].

〈그림. 1〉 조직에서의 프로세스 흐름 [Grover, Malhotra 1997]



2.1.2 비즈니스 리엔지니어링의 개념

비즈니스 리엔지니어링은 business process reengineering [Hammer&Champy, 1993], Core Process redesign [Kaplan & Murdoch, 1991], Process Innovation [Davenport & Short, 1990], Breakpoint business process redesign [Johansson et al., 1993] 등의 다양한 이름으로 불리며 그 명칭에 따라 의미하는 바도 약간씩의 차이를 보인다. 그러나 이런 명칭들은 모두 고객 만족도가 최대가 되도록 정보기술을 구사하여 업무과정을 맞추고 타사와의 경쟁에서 이겨 시대의 변화에 대응할 수 있도록 기업의 비즈니스 운영을 종래와는 전혀 다른 방법으로 재구축함을 의미한다. 〈표. 2〉는 리엔지니어링에 대한 대표적인 정의들이다.

〈표. 2〉 리엔지니어링에 대한 정의

Davenport & Short (1990)	조직내 및 조직간의 작업 흐름과 프로세스의 분석 및 설계[5]
Hammer & Champy (1993)	비즈니스 프로세스를 기본적으로 다시 생각하고 근본적으로 재설계하여 비용·품질·서비스·속도와 같은 핵심적 성과를 극적으로 향상시키는 것[6]
Petrozzo & Stpper (1994)	기업의 제품과 서비스에 대한 시간, 비용, 품질, 고객의 평가에 있어서의 급진적인 향상을 성취하기 위하여 프로세스, 조직, 그리고 그것들을 지원하는 정보시스템의 동시적인 재설계[7]

리엔지니어링(Reengineering)이란 용어는 Michael Hammer가 처음 사용하였는데, 그는 리엔지니어링을 "현대의 정보기술을 사용하여 업무 프로세스를 급진적으로 재설계 함으로써 프로세스의 성과를 극적으로 향상시키는 것"이라고 정의하였

5) the analysis and design of work flows and processes within and between organizations

6) the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service and speed

7) BPR involves the concurrent redesign of processes, organizations, and their supporting information systems to achieve radical improvement in time, cost, quality, and customers' regard for the company's products and services.

다[Hammer. 1990]. 이후 마이클 해머는 이를 좀 더 보충하여, "비즈니스 프로세스를 기본적으로 다시 생각하고 근본적으로 재설계하여 비용·품질·서비스·속도와 같은 핵심적인 성과를 극적으로 향상시키는 것"이라고 재정의 하였다. 이러한 정의 속에는 다음과 같은 개념들이 포함되어 있다[Hammer & Champy 1993].

첫째, 극적(dramatic)인 향상을 이룬다는 개념은 BPR이 점진적인 변화를 추구하는 것이 아니라, 업무성과의 극적인 향상을 추구한다는 것이다. 또한 점진적인 개선에서처럼 미세한 변화를 이루는 것이 아니라, 낡은 것을 버리고 새로운 것으로 과감하게 대체해야 한다는 것이다.

둘째, 근본적(fundamental)이라는 개념은 BPR을 실행함에 있어서 우리가 현재 하고 있는 일을 왜 해야 하는지, 혹은 왜 지금과 같은 방법으로 실행해야 하는지와 같은 가장 근본적인 질문을 해야 한다는 의미이다.

셋째, 완전히 새롭게(radical) 설계한다는 개념은 현존하는 모든 구조와 절차를 버리고 완전히 새로운 업무처리방법을 만들어 내는 것을 의미한다. 즉, 업무를 개선시키거나, 향상 혹은 변경시키는 것이 아니라 새롭게 만들어 내는 것이다.

마지막으로, 분석의 대상은 바로 비즈니스 프로세스이다. 프로세스란 하나 이상의 입력을 받아들여 고객에게 가치 있는 결과를 산출하는 행동들의 집합을 의미한다. 그러므로 이러한 프로세스에서 불필요하거나 중복되거나 낭비적인 요소를 제거하지 않고서는 경영혁신이 이루어질 수 없다는 것이다.

이러한 개념들을 통해 볼 때, 비즈니스 리엔지니어링이란 비즈니스 프로세스의 정확한 파악으로부터 출발하여 기업의 조직·문화에 이르는 기업의 경영 시스템 전체를 재구축 함으로서, 성취도를 대폭적으로 증가시키려는 경영혁신기법이라고 할 수 있다.

비즈니스 리엔지니어링은 〈표. 3〉에 보이는 바와 같이 종래에 시도되었던 여러 가지 경영개선활동들과 비교할 때 큰 차이점을 보인다. 과거의 경영개선이나 전사적인 품질관리 등은 현 업무의 점진적(incremental)인 효율성 증대를 목표로 하였으나, 비즈니스 리엔지니어링은 급격한 업무방식의 재구축을 통하여 극적인 효과를 목표로 하고 있기 때문이다[Davenport, 1993].

〈표. 3〉 프로세스 개선과 프로세스 혁신과의 차이 [Davenport, 1993]

구 분	프로세스 개선	프로세스 혁신
변화 수준	점진적	급진적
출발점	기존 프로세스	완전히 새로 시작(영점출발)
변화의 빈도	일회/지속적	일회
소요 기간	짧다	길다
참여	상향식 (종업원부터)	하향식(최고경영진부터)
전형적인 적용 범위	좁음, 기능내	넓음, 여러 기능에 걸침
위험 부담	보통	높음
주요 지원수단	통계/계량분석	정보기술
변화의 유형	문화적	문화적/구조적

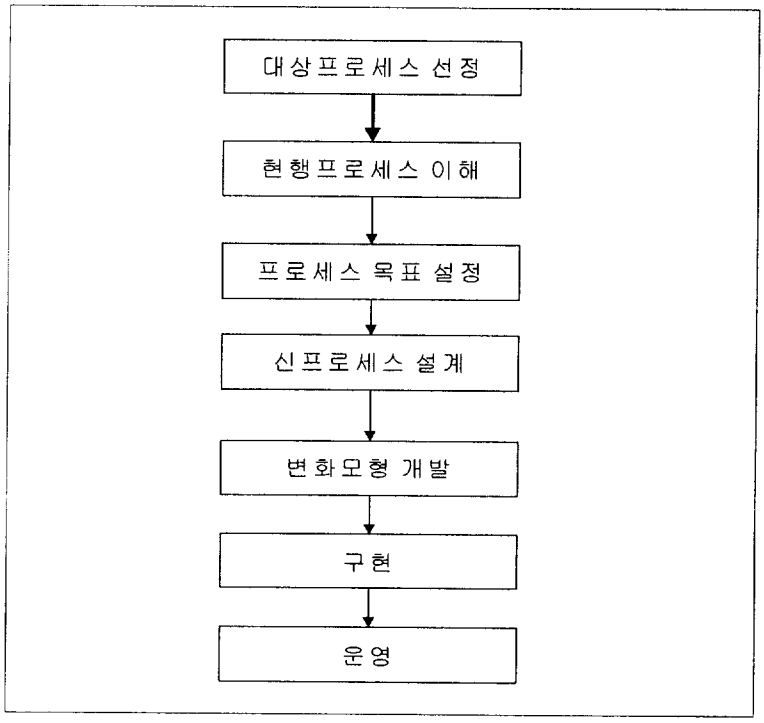
2.1.3 비즈니스 리엔지니어링의 추진절차

리엔지니어링을 추진하는 절차는 학자에 따라 여러 가지 의견이 있다. 그러나 여러 학자들의 의견을 종합하면 〈그림. 2〉 와 같다.

첫 번째 단계는 리엔지니어링이 필요한 대상 프로세스를 선정하는 것이다. 이 단계에서 대상 프로세스가 선정 되면 현행 프로세스에 대한 이해가 요구되며, 이를 토대로 프로세스 모델이 작성된다. 이 단계에서의 필수 도구는 프로세스 모델이다 [Cheung, Bal, 1998]. 프로세스 모델을 이용하여 현행 프로세스(As-Is Model)를 이해한다. 그리고 현행 프로세스를 이해하는 단계가 완료되면 프로세스의 목표를 설정한다.

프로세스 목표설정은 장차 이 프로세스가 어떤 모습으로 수행되어야 하는가를 기술하는 것이다. 프로세스 목표는 "이 프로세스가 어떠한 경영목표를 달성하여야 하는가"라는 질문으로 시작하여 경영전략, 고객의견 그리고 벤치마킹 기법을 통하여 변화하고자 하는 내용을 정의한다. 특히 이 단계에서 벤치마킹 기법은 업계에서 가장 우수한 성과를 내는 모범기업의 선진 프로세스(Best Practice)를 찾아내 이들의 성과를 기준으로 목표를 설정하도록 해 주는 역할을 한다[Harrison & Pratt, 1992; Richman & Konntz, 1993; Chang, 1994; Furey, 1993; 신홍철 외, 1994].

〈그림. 2〉 리엔지니어링 추진절차 [김효석외, 2000]



네 번째 단계는 리엔지니어링 과정에서 제일 중요하고 창의적인 부분인 신프로세스 설계 단계이다. 앞 단계에서 설정된 프로세스 목표를 달성하기 위해 프로세스가 어떻게 달라져야 할 것인가를 고안해내는 단계이다. 리엔지니어링 팀에 의한 브레인스토밍을 통하여 창의적인 아이디어를 발굴하고 벤치마킹으로 세계 초일류 기업들의 선진 프로세스들을 조사한다. 이렇게 발굴된 아이디어를 백지 위에 그리면서 설계내용이 구체화되면 새 업무방식에 대한 프로세스 모델(To-Be Model)을 다이어그램으로 표현한다.

다섯 번째 단계는 변화모형개발이다. 새로운 프로세스를 구현하고 운영하기 위해서는 기업의 조직, 정보시스템구조 및 성과측정/평가시스템을 변화시키는 것이 필요하다.

마지막으로 운영단계에서는 조직의 기술적인 측면을 고려하여 새로운 프로세스를 시범적으로 운영해본다.

2.1.4 리엔지니어링과 벤치마킹

벤치마킹의 정의를 살펴보면, 〈표. 4〉와 같다.

〈표. 4〉 벤치마킹의 정의

정의자	정의
미국생산성본부 (APQC)	벤치마킹은 체계적 그리고 지속적인 측정 프로세스이다. 자사의 성과개선에 유용한 정보를 얻기 위해 자사의 업무수행방식을 측정하고 동 방식을 전세계 어느 곳, 어느 조직이든 선도자적 위치에 있는 조직의 프로세스와 비교하는 과정
Camp (1989)	최고의 성취도를 획득하기 위하여 최고의 실제사례를 찾는 과정
Watson (1992)	자사에 유용한 타조직의 우수실무를 파악하는 방법론

위의 정의에서 발견할 수 있는 중요한 시사점은 '찾는 과정'이다. 즉, 벤치마킹을 실행하는 과정에서 가장 중요한 것은 지속적인 학습과 개선을 유지해 나가는 것이다[Camp, 1989].

앞 절의 리엔지니어링 추진절차에서 보인 바와 같이, 리엔지니어링을 실행하는 과정에서 벤치마킹은 유효한 수단으로 활용되고 있다[Harrison & Pratt, 1992; Richman & Konntz, 1993; Chang, 1994; Furey, 1993, 신홍철, 1994]. 특히 정보기술의 급격한 발전으로 정보기술을 사용하여 업무 프로세스를 재설계할 수 있는 기회가 늘어남에 따라, 벤치마킹을 위해 타사의 사례를 참조해야 할 필요성이 커지고, 벤치마킹의 중요성이 더욱 부각되고 있다.

Richman과 Konntz는 리엔지니어링에 있어 벤치마킹이 다음과 같은 측면에서 도움이 된다고 주장하였다[Richman & Konntz, 1993].

벤치마킹은 회사의 비전과 프로세스의 목적을 설정하는데 도움이 된다. 비즈니스 프로세스 리엔지니어링의 목적은 어떤 프로세스의 비용, 품질, 서비스, 그리고 시간과 같은 성과지표를 획기적으로 혁신하는 것이다. 이러한 비즈니스 리엔지니어링의 목적은 결과적으로 고객의 요구를 만족시키고, 기업의 경쟁력을 강화하고자 하는 벤치마킹의 목적과 기본적으로 일맥상통하는 측면이 있다. 그리고 비즈니스 리엔지니어링을 수행하기 위해서는 먼저 경영의 비전을 명확하게 정의해야 한다. 이는 비즈니스 리엔지니어링의 방향과 기업의 전략적 방향간에 일관성이 유지되도록 하기 위함이다. 이 과정에서 벤치마킹을 통해 시장에서의 자사의 경쟁적 위치를 우수한 기업의 그것과 비교해 봄으로써 전략적 방향을 올바로 설정하는데 도움이 된다.

또한 벤치마킹은 현행 프로세스를 이해하고 측정하는데도 도움을 준다. 벤치마킹에서도 마찬가지이지만 비즈니스 리엔지니어링 과정에서도 프로세스를 재설계 하기에 앞서, 먼저 조직의 현행 프로세스에 대해 충분히 이해하고 분석하는 과정을 거친다. 현행 프로세스를 분석하는 과정에서 도출되는 성과 측정치는 나중에 비즈니스 리엔지니어링 또는 벤치마킹의 목표를 정의할 때 기준으로 사용된다.

마지막으로 벤치마킹은 프로세스의 원형을 설계하고 구축하는데 도움이 된다. 실제로 프로세스를 재설계하고 재설계한 모습대로 프로세스를 구현하기 위해서는 많은 시행착오와 노력이 필요하다. 그리고 재설계 작업이 완료되었다 하더라도 외부환경이 변화함에 따라 영향을 받는 프로세스에 대해서는 변화를 반영할 수 있도록 지속적으로 수정해야 한다. 따라서 프로세스를 재설계 하기에 앞서 먼저 벤치마킹을 통해 다양한 자료를 수집·분석한 후 이를 토대로 해당 프로세스에 대한 재설계 작업을 수행하는 것이 바람직하다. 그리고 재설계가 끝난 다음에도 지속적인 벤치마킹을 통해 자사의 기준을 외부의 그것과 끊임없이 비교해 봄으로써 개선의 필요성이 있을 경우 즉시 그 결과를 반영한다.

이처럼 벤치마킹은 리엔지니어링을 수행하는 과정에서 많은 기여를 한다. 그러나 리엔지니어링을 위해 벤치마킹을 활용하는데는 벤치마킹의 대상으로 삼을만한 참조모형을 선정하기가 어렵다는 문제가 있다. 이에 프로세스 리파지토리는 좋은 대안이 될 수 있으며, 선진 프로세스(Best Practice)를 프로세스 리파지토리에 저장하고 이를 활용하자는 것이 프로세스 리파지토리의 기본 취지이다[Malone, 1999].

2.2 프로세스 모델

본 절에서는 프로세스 모델의 개념을 살펴보고, 본 연구에서 사용된 IDEF0 프로세스 모델링 방법론에 고찰하여 본다.

2.2.1 프로세스 모델의 개념

BPR과 벤치마킹의 성공에는 수많은 요인들이 있으나, 그 중 중요한 것은 프로세스를 표현하고 모델화 하는 것이다. 앞서 리엔지니어링 추진절차에서도 살펴보았듯이 프로세스 모델은 리엔지니어링을 위한 필수 도구이다[Cheung & Bal, 1998].

비즈니스 프로세스 리엔지니어링을 통하여 최대의 성과를 달성하려는 기업은 핵심 프로세스를 혁신하여야 하며, 이를 위해서는 먼저 핵심 프로세스와 이를 지원하는 프로세스, 그리고 이를 구성하는 활동(Activity)들을 상세히 파악할 필요가 있다. 즉 각각의 활동들에 대한 목적, 유발점, 입력 및 출력, 그리고 프로세스에 대한 이러한 이해와 분석은 프로세스 모델링과 평가 기법을 활용함으로써 효과적으로 달성할 수 있다[Johansson et al., 1993].

Hunt는 프로세스 모델을 "비즈니스 프로세스를 리엔지니어링 하기 위하여 현행 프로세스를 향상시키거나 새로운 프로세스 지향적인 구조의 구축을 지원하는 분석 및 커뮤니케이션 도구"라고 정의하였다[Hunt, 1996]. 프로세스 모델링 기법은 특정 프로세스의 흐름을 시각적으로 묘사하기 위하여 사용하는 도구이며, 프로세스 모델링의 목적은 기존의 비즈니스 프로세스를 상세하게 분석함으로써 프로세스 재설계를 지원하기 위한 것이다[Wang, 1994].

Curtis는 프로세스의 구성요소를 〈표. 5〉 과 같이 5W1H(What, Why, Who, Where, When, How)로 정의하고 이것을 표현하기 위하여 〈표. 6〉 과 같이 프로세스 모델의 네 가지 관점을 제시하였다[Curtis et al., 1992].

〈표. 5〉 프로세스 구성요소 분류

What(프로세스 및 관련개체)	수행되는 프로세스 입력자원 요구사항 프로세스의 결과물
Why(프로세스의 수행 목적)	프로세스의 수행목적 프로세스의 가치
Who(프로세스와 관련된 수행자)	프로세스의 수행자 프로세스 수행자의 역할 프로세스의 고객 수행자-고객 관계
Where(프로세스와 관련된 장소)	프로세스 수행 장소(물리적) 프로세스 수행 부서(개념적) 프로세스 결과물의 이동
When(프로세스와 관련된 시간)	프로세스의 수행시기
How(프로세스의 수행방식)	수행절차 자원 사용 기준

〈표. 6〉 프로세스 모델링의 4가지 관점

정보적	작업활동의 목적 작업활동의 유형 작업활동의 가치
기능적	수행작업 입력자료 및 산출물의 흐름
행위적	작업순서 피드백 루프 의사결정조건
조직적	프로세스의 수행 부서 수행자의 역할 의사소통 매체

5W1H에서 "What"은 프로세스 또는 프로세스와 관련 있는 개체, "Why"는 프로세스 수행이유 즉, 수행목적을 나타내며, "Who"는 프로세스와 관련된 사람으로서 수행자 혹은 고객을 뜻하며, "Where"는 프로세스와 관련된 장소, "When"은 프로세스와 관련된 시간, "How"는 프로세스의 수행방식과 같은 내용을 나타낸다.

프로세스 모델링을 통하여 얻고자 하는 정보들은 앞에서 언급되었던 5W1H의 프로세스 구성요소에 관한 것들이다. 그러나 비즈니스 프로세스는 그 안에 포함된 정보가 광범위하고 복잡하기 때문에 이를 모델링 하기는 쉽지 않으며 아직까지 공인된 도구는 없다[조윤호 외, 1998].

대표적인 프로세스 모델링 방법론에는 IDEF0, Flow Chart, DFD (Data Flow Diagrams), Petrinet [Alast, Hee, 1996], OOA (Object Oriented Analysis) [Yu, Wright, 1997], Prince2(Process based Project management)[8] 등 여러 가지 방법론이 있다. 그러나 위의 프로세스 모델링 기법들은 각기 나름대로의 관점을 가지고 정보를 표현하고 있으며 어떠한 기법도 모든 관점에서의 프로세스 정보를 다 표현하지는 못한다[Wang, 1994].

2.2.2 IDEF0 모델링 방법론

IDEF(Integrated computer aided manufacturing DEFinition Language)는 1981년 미 공군의 ICAM(Integrated Computer Aided Manufacturing) 프로젝트의 생산 시스템 분석 및 설계 목적으로 개발되었다. 생산 시스템을 세가지 관점 즉, 기능, 정보, 그리고 동적 관점으로 나누고 각각의 관점을 모형화할 수 있는 방법론을 개발하였다. 가장 많이 사용되고 있는 것은 기능 모형을 위한 IDEF0(IDEF Function Modeling)와 정보 모형화를 위한 IDEF1(IDEF Information Modeling), 시스템 내부의 데이터를 모델링 하는 기법인 IDEF1X, 동적모형을 위한 IDEF2(IDEF Dynamics Modeling), 시스템의 작동 프로세스를 모델링하는 IDEF3(IDEF Process Modeling) 방법론이 있다. 또한 객체지향 설계 기법인 IDEF4(IDEF Object-Oriented Design)와 온톨로지 표현 기법인 IDEF5(IDEF Ontology Description Capture)도 개발 중이다[9].

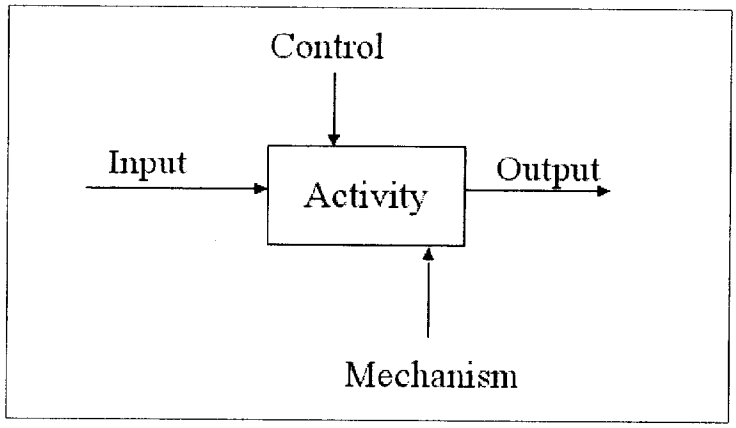
8) Prince2 홈페이지, http://ww.prince2.com/managementoverview.html/
9) IDEF 홈페이지, http://www.idef.com

IDEF는 외형적으로는 각각 독립된 기법으로 구성되어 있지만 모델 간에 호환성이 있어 특정 기법으로 표현된 모델을 언제든지 다른 기법의 모델로 변환할 수 있다. 이러한 특성 때문에 IDEF 방법론은 PROSIM, BPWin 등과 같은 수많은 CASE 도구의 모델링 방법론으로 활용되고 있다. 그리고 상용화된 일부 제품들은 IDEF0에 ABC 기능을 부가하는 방식을 채택하고 있다[Cheung, Bal, 1998].

IDEF0는 기업에서 수행 되어지고 있거나 수행되어야 할 활동, 또는 활동과 활동간의 관계를 박스와 화살표로 표현하며 각각의 박스는 계층적으로 분해 가능한 셀(Cell)로 가정한다[Hunt, 1996].

IDEF0 다이어그램의 기본구성은 〈그림. 3〉과 같이 활동(activity)과 ICOM (Inputs, Outputs, Controls, Mechanisms)으로 되어 있다. 활동은 시간에 걸쳐 일어나고 목표와 측정 가능한 결과를 가지는 비즈니스 프로세스, 기능(function), 또는 과업(task)이며[Hunt, 1996], 이것은 박스 형태의 다이어그램으로 표현된다. 그리고 활동박스에는 활동의 고유번호를 명시한다. ICOM은 활동이 수행되는데 필요한 개체들, 혹은 활동에 관여된 개체들로서, 활동과 연결된 화살표로 나타내며 입력(input), 출력(output), 제어(control), 매커니즘(mechanism)의 네 가지 구성요소로 이루어져 있다.

〈그림. 3〉 IDEF0의 ICOM



입력(Input)은 활동 박스의 왼쪽으로부터 들어가는 화살표로서 활동을 수행하는데 필요한 개체 혹은 데이터이다. 활동이 수행됨에 의하여 소모되거나 변형된다는 특징을 가지고 있다. 예를 들면 재료, 반제품, 기초데이터 등이 있다.

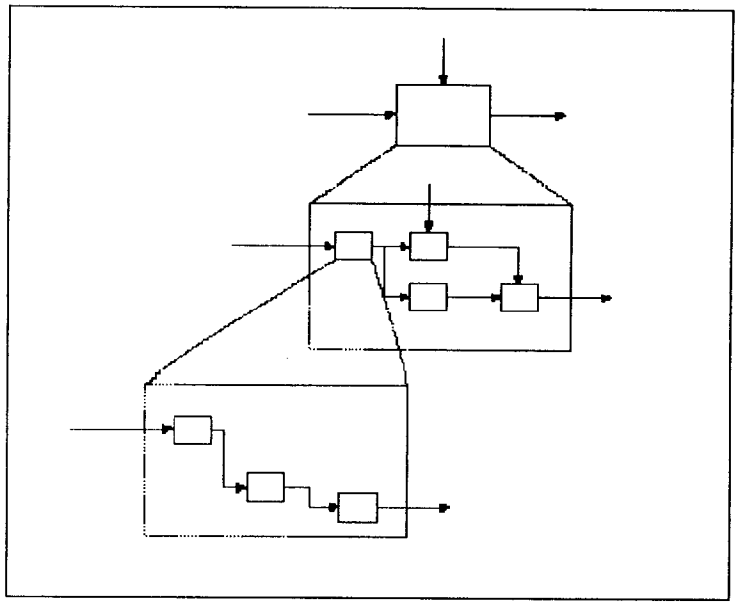
출력(Output)은 기능 박스의 오른쪽으로 나오는 화살표로 표현되며 활동의 결과로 산출되는 산출물을 말한다. 반제품, 제품, 가공된 자료 등이 이에 해당한다.

제어(Control)는 활동 박스의 위쪽으로부터 들어가는 화살표로 표현되며 활동을 통제·제어하는 제약조건, 가이드 혹은 출력을 결정하는데 필요한 제약조건 등이다. 이를테면 작업 표준, 회사 규정, 품질표준 등은 전형적인 제어이다.

매커니즘(Mechanism)은 활동박스의 밑에서부터 들어가는 화살표로 표현되며 활동을 수행하는 사람 또는 개체 등이다. 이것은 무엇에 의해 그 기능이 수행되는가 혹은 기능의 수행에 어떠한 자원이 소용되는가를 나타낸다. 사람, 기계, 로봇, 정보시스템 등이 매커니즘에 해당한다.

그리고 입력은 활동에 의하여 출력으로 변환되며, 제어는 활동에 의하여 변환되지 않는다는 차이점을 가지고 있다.

〈그림. 4〉 IDEF0의 탑다운(top-down) 분해[Biazzo, 200]



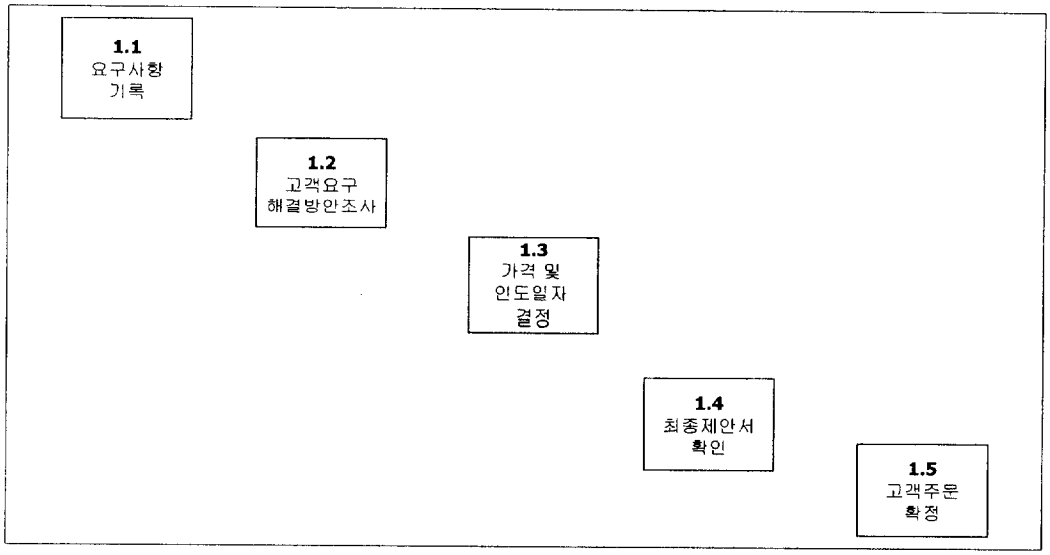
IDEF0 모델링에 있어서 기본이 되는 또 하나의 표현 방법은 활동의 수직적 해체라는 개념이다 〈그림. 4〉. 이것은 셀(Cell) 모델링 표현방법 또는 탑다운 분해 방법이라고도 하는데 기업에서 분석되어야 하는 활동을 계속적으로 분해하여, 상위단계의 활동을 구성하는 하위단계의 활동들을 분석하는 것이다. 즉, 탑다운 분해는 하나의 활동을 이를 구성하는 하위 활동으로 나눔으로서 활동을 상세화 한다. 그리고 이러한 기능은 활동이 무엇을 수행하는지에 대한 이해와 의사소통에 요구되는 자세한 사항에 대한 단계적인 설명을 지원한다.

IDEF0의 모델링의 절차는 네단계로 이루어져 있다.

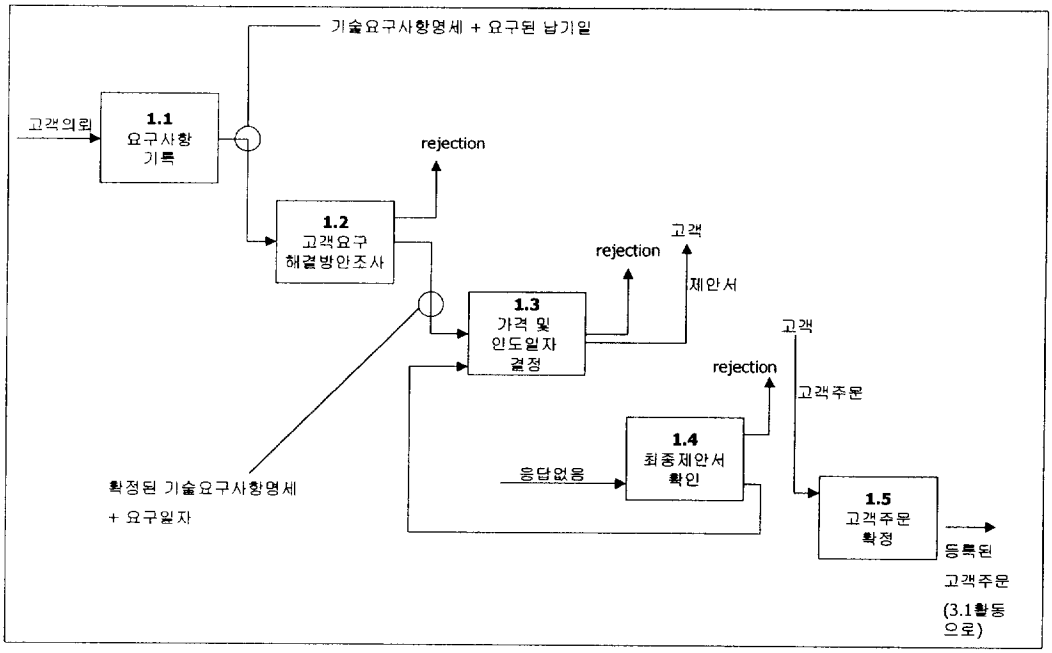
첫번째로, 활동들을 박스로 표현하고, 각 다이어그램마다 활동명과 고유의 활동번호를 붙인다 〈그림. 5〉. 두 번째로, 각 활동과 활동간의 입출력 흐름을 표현한다 〈그림. 6〉. 세번째로, 각 활동의 수행을 제어하는 사항, 즉 컨트럴을 표현한다 〈그림. 7〉. 마지막으로, 이들 활동의 수행이 무엇에 의해 수행되는가, 즉 매커니즘을 표현한다 〈그림. 8〉. 완료된 활동모델은 현실세계의 활동 및 입출력 관계뿐 아니라 이들 활동에서 발생되는 정보, 이를 통제하는 제약 및 수행하는 메커니즘을 직관적으로 파악할 수 있다[Hunt, 1996].

IDEF0의 모델링 절차를 예시하기 위해 사용된 프로세스 사례는, Tatsiopolous의 연구에서 제시된 제조업체의 주문처리 프로세스 중 주문획득 활동이다 [Tatsiopolous, Panayiotou, 2000].

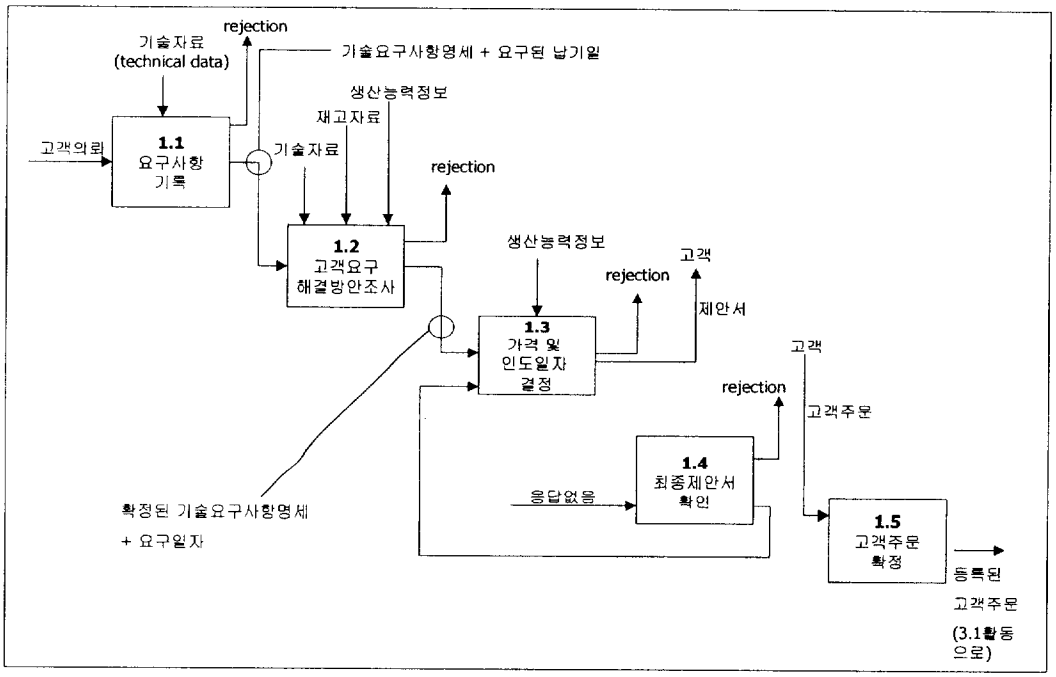
〈그림. 5〉 IDEF0 모델링의 예. 1



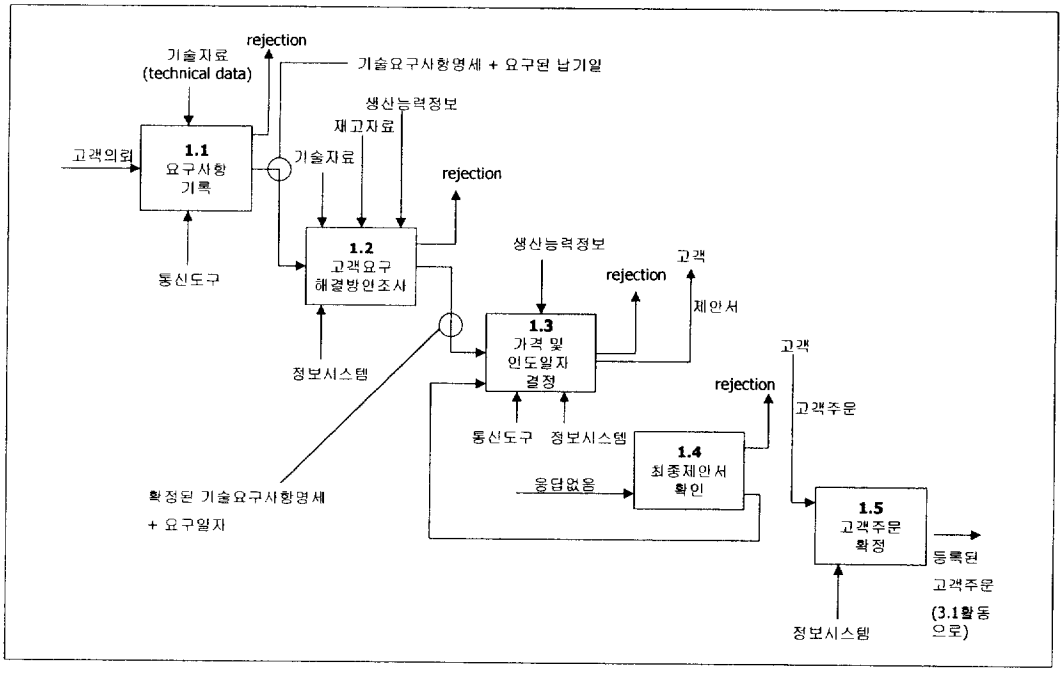
〈그림. 6〉 IDEF0 모델링의 예. 2



〈그림. 7〉 IDEF0 모델링의 예. 3



〈그림. 8〉 IDEF0 모델링의 예. 4



2.3 프로세스 리파지토리

이 절에서는 프로세스 리파지토리와 관련한 선행 연구를 살펴본다. 앞 절에서 살펴본 바와 같이 벤치마킹은 리엔지니어링을 수행하는 과정에서 많은 기여를 한다. 왜냐하면 벤치마킹을 통해 프로세스에 대한 아이디어를 얻음으로서 리엔지니어링을 효과적으로 추진할 수 있기 때문이다. 그리고 본 연구에서 제시하는 주요한 개념중의 하나인 프로세스 리파지토리는 바로 이러한 벤치마킹을 지원하기 위한 도구이다. 선진 프로세스(Best Practice)를 프로세스 리파지토리에 저장하고 이를 활용하자는 것이 프로세스 리파지토리의 기본 취지이다[Malone, 1999].

2.3.1 프로세스 리파지토리의 개념

리파지토리의 정의는 학자들마다 매우 다양하다. MacHaughey는 리파지토리를

"정보저장을 위한 빈 그릇으로 비유할 수 있으며, 만일 기업모형의 컴포넌트들이 채워진다면 정보백과사전이라고 할수 있다"고 정의하였으며[MacGauhey, 1993], Martin은 "리파지토리는 단순한 자료정보 뿐만 아니라 계획, 모델, 설계사양의 함축적 표현도 포함한다" 고 정의하였다[Martin, 1989]. 리파지토리는 CASE 리파지토리[MacGauhey, 1993], 디자인 리파지토리[Szykman, 2000], 소프트웨어 리파지토리[Clayton, 2000], 프로세스 리파지토리[Carr, 1999] 등 다양한 분야가 있다. 리파지토리는 어떻게 사용되어 지느냐에 따라 데이터베이스, 파일, 또는 문서들의 저장장소일 수 있다. 리파지토리에 저장되는 정보들은 주로 단순한 운영데이터가 아닌 소프트웨어 구현코드, 기업 모형, 프로젝트 자료와 규칙, 디자인 등이다[10]. 즉 리파지토리는 지식관리 기능을 수행하는 구성요소라고 할 수 있다[이희석 외, 1999].

MIT에서 수행된 프로세스 리파지토리 연구성과를 상업화 한 피오스(Phios)사[11]에서는 프로세스 리파지토리를 "다양한 목적으로 사용되는 일관성 있고 사용하기 쉬운 프로세스 지식의 저장소[12]"로 정의하고 있다. 즉, 프로세스 리파지토리는 프로세스 모델로 표현된 선진 프로세스(Best Practice)를 저장한 지식베이스라고 할 수 있다. 본 연구에서는 피오스사의 정의에 따라 연구를 수행하였다.

앞서 살펴본 바와 같이 리엔지니어링 수행시 벤치마킹은 중요한 역할을 한다. 다시 말해서 서로 유사한 프로세스 모델을 공유하고 이를 충분히 비교·분석하였는가의 여부가 리엔지니어링 및 벤치마킹에서 매우 중요하다. 이 때문에 일부 컨설팅 회사와 기업들은 나름대로 물류, 마케팅, 제조 등과 같은 범용 프로세스를 정의하여 활용하고 있다. 그러나 이렇게 확보된 프로세스 사례들은 범위가 특정 업무에 국한되었거나, 서류와 같은 비전자적인 매체에 저장되어 있기 때문에 이를 보관, 검색, 참조하기가 상당히 어려운 것이 현실이다. 또한 각 기업은 자사의 프로세스에 관한 각종 표준, 매뉴얼 등의 자료를 보유하고 있지만 이를 사용자의 목적에 맞게 재구성하거나, 내용을 검색하기가 힘들다. 아울러 문서형태로 보관되어 있는 이러한 자료들은 최신의 내용으로 유지하는데도 많은 어려움이 따른다[Malone, 1999].

---

10) http://www.helloec.net/network/Repository.htm

11) http://www.phios.com/hbr

12) consistent, easy-to-use collections of process knowledge, used for multiple purposes

이러한 문제점을 개선하기 위해서는 정형화된 모델링 도구로 묘사된 업무사례를 전자적인 방식으로 통합된 장소에 보관하는 것이 필요하다. 이런 목적으로 구축된 시스템이 프로세스 리파지토리이다. 이러한 프로세스 리파지토리는 리엔지니어링 및 벤치마킹에 필요한 선진 프로세스(Best practice)를 제공하는데 많은 장점을 제공할 수 있다.

2.3.2 프로세스 리파지토리 관련 선행연구

프로세스 리파지토리 연구 중에서 가장 대표적인 것이 MIT의 프로세스 핸드북 리파지토리(Process Handbook Repository)이다[13]. 이 연구에서는 기존의 프로세스 재설계나 전사적 품질관리 등의 경영혁신 기법들이 지속적인 경영혁신을 주장하고 있기는 하지만 실제로 이를 실천하는데 필요한 지침을 제공하는데는 미흡했다는 인식에서 출발하고 있다. 그리고 경영혁신의 필요성과 이를 위한 수단이 부재한 현실간의 괴리를 극복하기 위하여 프로세스 핸드북 리파지토리를 제안하고 있다[Malone, 1999].

현재, 프로세스 핸드북 리파지토리는 5000개 이상의 프로세스들과 활동(Activity)들을 저장하고있다. 이를 위해 MIT 프로세스 핸드북 프로젝트 팀은 데이터를 탐색하고 관리하기 위해 소프트웨어 패키지를 개발하였으며, 프로세스 핸드북은 상용화되어, 계약을 맺은 기업들에 한해서 월드와이드웹을 통해 프로세스 리파지토리를 제공하고 있다[14].

조직은 자사의 프로세스를 향상시키기 위해, 프로세스 핸드북을 이용하여 최고의 운영방법을 배울 수 있고, 프로세스를 창조적으로 수정 또는 확장 할 수도 있다. 이 연구에서는 과거 조직의 경험을 활용하고, 정보기술 분야에서 쏟아지는 신기술을 조직의 프로세스 재설계에 접목하기 위해 다음과 같은 방법을 제시하고 있다.

프로세스 핸드북 프로젝트는 조직의 비즈니스 프로세스를 분류하기 위한 독특한 체계를 개발하였다. 프로세스 핸드북의 2차원 구조는 프로세스 파트(Process

13) http://dime.mit.edu/html/project.html
14) http://ccs.mit.edu/ph/

Parts)와 프로세스 타입(Process Types)에 의해서 정보를 조직한다. 예를 들어, 일반적인 제품판매 프로세스를 탐색하는 사용자는 프로세스의 컴포넌트 파트(Components Part) 또는 서브액티버티(subactivity)에 관한 상세한 정보를 획득하기 위해 데이터베이스를 통해 수직적으로 이동하여 검색한다. 만약 인터넷을 통한 판매 또는 재무서비스 판매와 같은 전문화된 프로세스를 조사하기 위해서는 수평적으로 이동하여 검색한다. 이러한 구조의 장점은 탐색하는 과정에서 자신이 의도한 내용과는 직접적인 관련성은 없지만 프로세스를 재설계 하는데 참조할 수 있는 다양한 아이디어를 수집할 수 있다는 것이다.

그리고 특정한 상황에서 유용하고 바람직한 대안을 선정할 수 있도록 여러 가지 대안을 제시한다. 이를 프로세스 핸드북 리파지토리에서는 번들(bunldle)이라고 한다. 번들이란 동일한 목적을 가진, 그러나 각기 다른 방법이나 다른 대상을 가진 프로세스들의 집합을 의미하며, 이를 통해 프로세스를 비교·분석할 수 있다.

조직은 프로세스 핸드북을 다음과 같은 두가지 방법으로 이용할 수 있다. 첫째는, 프로세스 리파지토리를 이용하여 자사의 프로세스들을 명백하게 정의하고 관리한다. 둘째는, 타 조직이 유사한 프로세스를 어떻게 수행하는지의 첨단의 지식을 획득할 수 있다. 예를 들어, Dow Corning사는 SAP 시스템을 도입하는 과정에서 자사의 모든 프로세스들을 기록하고 전 조직에 걸쳐 정보를 공유하기 위한 일관성 있는 방법이 부재함을 깨닫고 프로세스 리파지토리를 도입하였으며, 현재 Dow Corning사는 프로세스 리파지토리를 자사의 인트라넷으로 흡수시킬 계획을 가지고 있다. 또 다른 활용 사례로, Supply Chain Council에서는 400개 이상의 기업들에게 프로세스 리파지토리를 통해 공급망 프로세스의 산업표준 모델들을 제공하고 있다[Carr, 1999].

국내의 연구로는 기업정보화지원센터가 각 업종별 표준업무 프로세스 리파지토리를 개발 중이며, 현재 그 중 일부 프로세스를 사이버 IT 기술지원센터 또는 기업정보화지원센터 홈페이지에 설치하여 일반 중소기업에게 보급하고 있다[홍정완, 2001][15]. 이 시스템은 비즈니스 프로세스, 검색, 사전, 도움말로 구성되며 HTML 페이지로 제공되고 있다.

업무와 관련된 자료를 저장하고 재활용하기 위한 노력의 일환으로 전사적자원

15) http://www.itr.re.kr/process

관리(Enterprise Resource Planning: ERP)를 위한 리파지토리를 들 수 있다. ERP 시스템에서는 기업내에 존재하는 다양한 정보자산을 통합적으로 관리하고 활용하기 위해 리파지토리를 사용한다. 그러나 ERP 시스템의 리파지토리는 본 연구에서 제시하는 프로세스 리파지토리와는 구분되는 개념이다. 그러나 기업의 정보를 체계적으로 저장하고 관리한다는 측면에서는 유사한 면이 있다. 이희석 등에 의해 수행된 이 연구는 전사적 정보자산을 통합적으로 관리하기 위한 연구이다[이희석 외, 1999]. 이 연구에서 제시하는 리파지토리는 정보시스템을 분석, 설계, 구현하는데 필요한 여러 가지 도구들을 통합적으로 관리하는 역할을 한다. 그리고 리파지토리에 접근하기 위한 수단으로 데이터베이스 관리시스템과의 인터페이스를 제공한다. 이 연구의 핵심주제는 리파지토리와 관련한 표준중 하나인 IRDS(Information Resource Dictionary System) 기반의 리파지토리를 구현하는 것이다. 그러나 이 시스템은 정보시스템의 구축 및 사용에 관련된 정보를 리파지토리에 저장하려는 연구이며, 본 연구의 주제인 프로세스 리파지토리와는 성격이 많이 다르다고 할 수 있다.

2.3.3 선행 연구의 한계

프로세스 리파지토리에 대한 선행 연구들은 선진 프로세스(Best Practice)를 체계적으로 저장하여 이를 필요에 따라서 활용하려는 목적을 가지고 있으며 프로세스 리엔지니어링 및 벤치마킹을 지원하기 위한 종합적 지원도구의 성격을 가지고 있다. 그러나 이러한 선행연구의 한계점을 살펴보면 다음과 같다.

첫째, 앞서 살펴본 프로세스 리파지토리들은 대부분 다이어그램으로 표현된 비즈니스 프로세스를 이미지 파일 또는 문서 파일과 같은 비구조적 데이터의 형태로 프로세스 모델들을 보유하고 있다. 그러나 이렇게 표현된 프로세스는 필요에 따라 변경하기가 용이하지 않다. 예를 들어서 MIT의 프로세스 핸드북은 고유의 모델링 기법을 사용하여 프로세스를 모델링하여 저장하고 있다. 그러나 이러한 모델은 설혹 전자적인 형식의 파일로 저장되어 있다 하더라도, 여기에 포함된 몇 가지 활동을 수정하고자 할 때는 전체 프로세스를 재작성해야 한다.

둘째, 이미지 또는 문서와 같은 비구조적 데이터로 저장된 프로세스 모델들은 프로세스들을 비교·분석하기가 용이하지 않다는 단점을 지적할 수 있다. 기존의 연

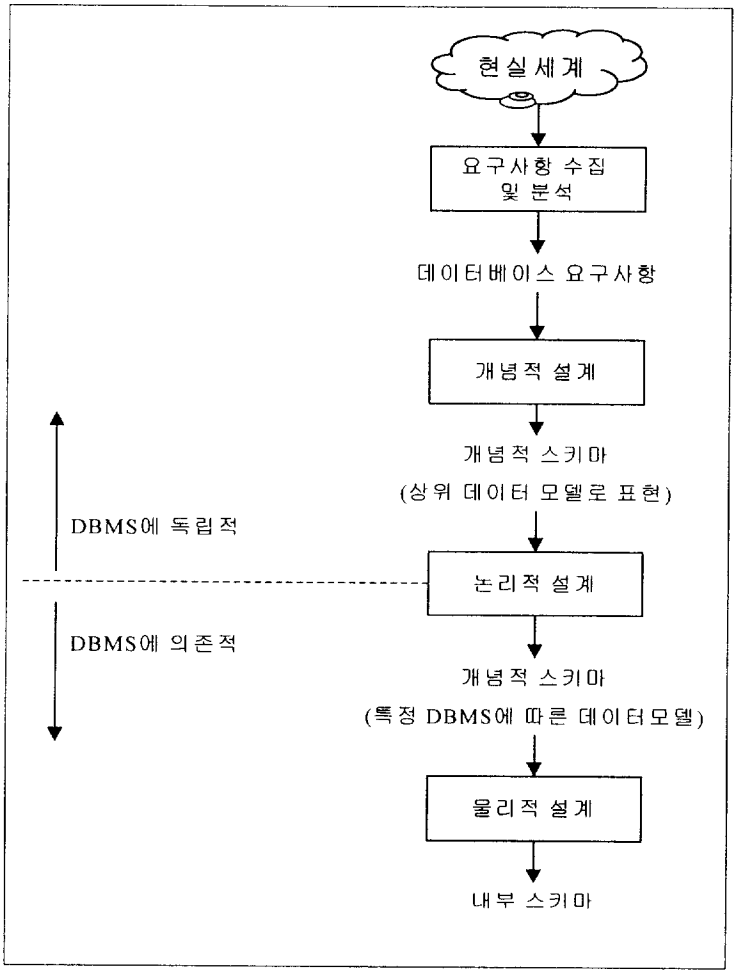
구들은 프로세스를 표현하기 위한 나름대로의 모델링 기법은 가지고 있지만, 정작 이를 비교·분석하여 벤치마킹 수행시 활용할 수 있는 방안제시는 못하고 있다.

셋째, 기업정보화지원센터에서 수행된 연구에서는 프로세스 모델을 구조적 데이터로 관계형 데이터베이스에 저장하기 위한 스키마를 제시하였다[그림. 9]. 그러나 이것은 프로세스의 탑다운(top-down) 분해가 네단계까지 밖에 않된다는 단점을 가지고 있기 때문에 규모가 크고 복잡한 프로세스를 표현하는데 한계를 가진다. 프로세스 모델은 큰 프로세스도 표현할 수 있도록 탑다운 분해를 충분히 지원해야 한다.

이러한 단점을 극복하기 위하여 본 연구에서는 프로세스 모델을 구조적 데이터의 형태로 관계형 데이터베이스에 저장함으로서 프로세스의 비교·분석 기능을 강화하였으며, 프로세스 모델로 IDEF0를 선정함으로서 하향식 탑다운 분해를 충분히 지원하도록 하였다.

제 3 장 스키마 설계

〈그림. 9〉 데이터베이스 스키마 설계 단계 [Shamkant, 1992]



스키마란 데이터베이스의 논리적 기술, 즉 전체적인 데이터베이스의 구조를 기술한 것이다. 개념적 스키마는 논리적인 데이터베이스의 구조를 표현한 것으로 개체-관계도 (Entity Relationship Diagram: ERD) 에 의해 기술되며 물리적인 저장구

조의 표현은 감추고 개체와 이들간의 관계를 표현하는데 집중한다. 이러한 개념 스키마는 곧바로 데이터 정의어(Data definition Language: DDL)와의 일대일 사상(mapping)에 의한 물리적인 데이터베이스 구조로 변환된다. 내부 스키마(internal schema)는 데이터베이스의 물리적 저장구조를 기술한다. 그리고 요구사항 조사로부터 최종적으로 DBMS에 구현할 수 있는 스키마 정의 까지의 모든 활동을 스키마 설계라고 한다[Shamkant, 1992].

3장에서는 본 연구에 적합한 프로세스 모델을 선정하여, 프로세스 리파지토리를 위한 개념적 스키마와 내부 스키마를 제시한다.

3.1 본 연구의 프로세스 모델

프로세스 리파지토리를 구축하기 위해서는 적합한 프로세스 모델링 기법을 선정하는 것이 중요하다. 프로세스 모델링 기법들은 각기 나름대로의 관점을 가지고 정보를 표현하며 어떠한 기법도 모든 관점에서의 정보를 표현하지는 못한다. 그리고 너무 많은 정보를 하나의 다이어그램에 표현하려고 하는 경우 지나치게 복잡해진다[Wang, 1994].

본 연구는 프로세스 모델로 표현된 선진 프로세스(Best Practice)를 저장하는 프로세스 리파지토리의 스키마 설계라는 목적을 가지고 있다. 그리고 단순한 프로세스 모델의 제공을 넘어서 구조적 데이터의 형태로 관계형 데이터베이스에 저장된 프로세스 모델의 분석기능을 제공하고자 한다. 따라서 프로세스의 묘사뿐만 아니라 분석기능이 뛰어난 프로세스 모델링 도구의 선정이 필요하다. 아울러 단순히 기능적인 관점이 아닌 범기능적인(cross-functional) 프로세스의 표현이 가능한 도구를 선정하여야 한다.

본 연구에서는 프로세스를 표현하기 위한 프로세스 모델링 도구로서 IDEF0를 선택하였으며, IDEF0 방법론의 장점은 다음과 같다.

첫째로, IDEF0는 탑다운(top-down) 방식으로 프로세스를 분해하여 분석해 나가므로, 최하위 엑티버티의 의미까지 상세히 표현할 수 있다[Jarzabek, Ling, 1995; Shimizu, Sahara, 2000].

둘째로, 간단한 사용상의 규칙과 표기법으로 인해 사용자들이 쉽게 배우고, 사용할 수 있다. [Shapiro, 1994].

셋째로, IDEF0는 업계에서 광범위하게 보급되어, 프로세스 모델링 툴로서의 유용성을 인정받고 있으며. 구미의 대부분의 시스템 분석가들은 프로세스 모델링 툴로서 IDEF를 교육받고 있다[Shapiro, 1994].

넷째로, IDEF0는 빠르게 모델링할 수 있기 때문에 비즈니스 리엔지니어링과 같은 급속한 변화를 추구할 때 매우 적합하다[조윤호 외, 1998].

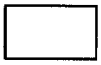
마지막으로, IDEF0는 ABC와의 연계가 용이하다[Tatsiopolous, Panayiotou, 2000; Shimizu, Sahara, 2000]. 그리고 ABC에서 제공하는 여러 정보는 리엔지니어링과 같은 최근 기업조직의 혁신기법들과 적절히 조화되어 사용될 수 있다[유관희, 1998]. ABC와의 연계가 용이하다는 장점은 프로세스 모델이 구조적 데이터로 관계형 데이터베이스에 저장될 경우 더욱 그 유용성을 발휘할 수 있으며, 프로세스를 비교·분석할 수 있는 기준을 제시한다.

3.2 개념적 스키마(Conceptual Schema)

개념적 스키마를 설계하는 첫 번째 단계는 대상으로부터 개체(entity)와 그들간의 관계(relationship)을 파악하는 것이다. 우선 [표. 7]과 같이 IDEF0의 구성요소로부터 개체(entity)를 도출하였다.

〈표. 7〉 에서 좌측은 IDEF0 다이어그램이고, 우측은 IDEF0 다이어그램으로부터 도출된 개체(Entity)들이다. IDEF0 다이어그램의 기본구성은 활동(activity)과 ICOM (Inputs, Outputs, Controls, Mechanisms)이다. IDEF0 다이어그램의 구성요소 중 활동, 컨트럴, 매커니즘은 그대로 동일한 명칭의 개체로 도출하였고, 입력과 출력은 IO라는 하나의 개체로 도출하였다. 선행 활동에서 나온 출력이 곧 후행 활동의 입력에 해당하므로 출력과 입력은 IO라는 하나의 개체로 파악될 수 있다. 그러므로 입력과 출력을 하나로 묶어서 IO라는 하나의 개체를 도출하였다.

〈표. 7〉 IDEF0 프로세스 다이어그램으로부터의 개체 도출

IDEF0 다이어그램		개체 (Entity)
활동(Activity)		Activity
입력(Input)	→	IO
출력(Output)	→	
컨트럴(Control)	↓	Control
매커니즘(Mechanism)	↑	Mechanism

〈표. 8〉 개체와 그 속성

활동 (활동번호, 활동레벨, 활동명, 활동설명, 직접비, 간접비, 원가동인, 상위활동)
activity act_id act_level act_name act_desc direct_cost indirect_cost cost_driver parent_act

IO (IO번호, IO명, IO설명, 선행활동, 후행활동)
io io_id io_name io_desc from_act to_act

컨트럴 (컨트럴번호, 컨트럴명, 컨트럴설명)
control control_id, control_name, control_desc

매커니즘 (매커니즘번호, 매커니즘명, 매커니즘설명)
mechanism mech_id mech_name mech_desc

〈표. 8〉 에 각 개체들의 개체명 및 속성명이 서술되어 있다.

활동(activity) 개체는 활동번호(act_id), 활동레벨(act_level), 활동명(act_level), 활동설명(act_desc), 직접비(direct_cost), 간접비(indirect_cost), 원가동인(cost_driver), 그리고 상위활동(parent_act)이라는 속성들을 가진다. 활동번호(act_id) 속성은 활동 개체의 주키(primary key)이고, 활동레벨(act_level)은 IDEF0 모델에서 활동마다 주어지는 고유번호이다. 특기할 것은 활동 개체의 속성에 직접원가(direct_cost), 간접원가(indirect_cost), 원가동인(cost_driver)이 있다는 것이다.

직접원가는 특정제품과 관련하여 개별적으로 추적 가능한 원가로서 예컨데, 특정 제품을 생산하기 위하여 투입된 특정 원재료와 특정 노동자의 노동시간, 특정 기계설비의 가동원가 등이 예가 될 수 있다. 간접비는 특정 제품과 관련하여 그 제품의 생산을 위해 소멸된 경제적 희생과 창출된 부가가치를 연결시키기 어려운 경우의 원가를 말한다. 그리고 원가동인은 왜 그런 수준의 원가가 발생되었는가를 설명해 주는 것으로 원가를 발생시키거나 또는 발생 정도에 영향을 미치는 요소라고 정의될 수 있다[유관희, 1998]. 직접비, 간접비, 원가동인은 간단하게나마 프로세스 리파지토리의 계량적 비교·분석 기능을 제공하기 위하여 추가한 것이다. 그리고 상위활동(parent_act) 속성은 IDEF0의 계층적 분해 기능을 지원하기 위하여 추가한 속성이다. 상위활동 속성의 값을 통하여 활동들간의 계층구조를 파악할 수 있다.

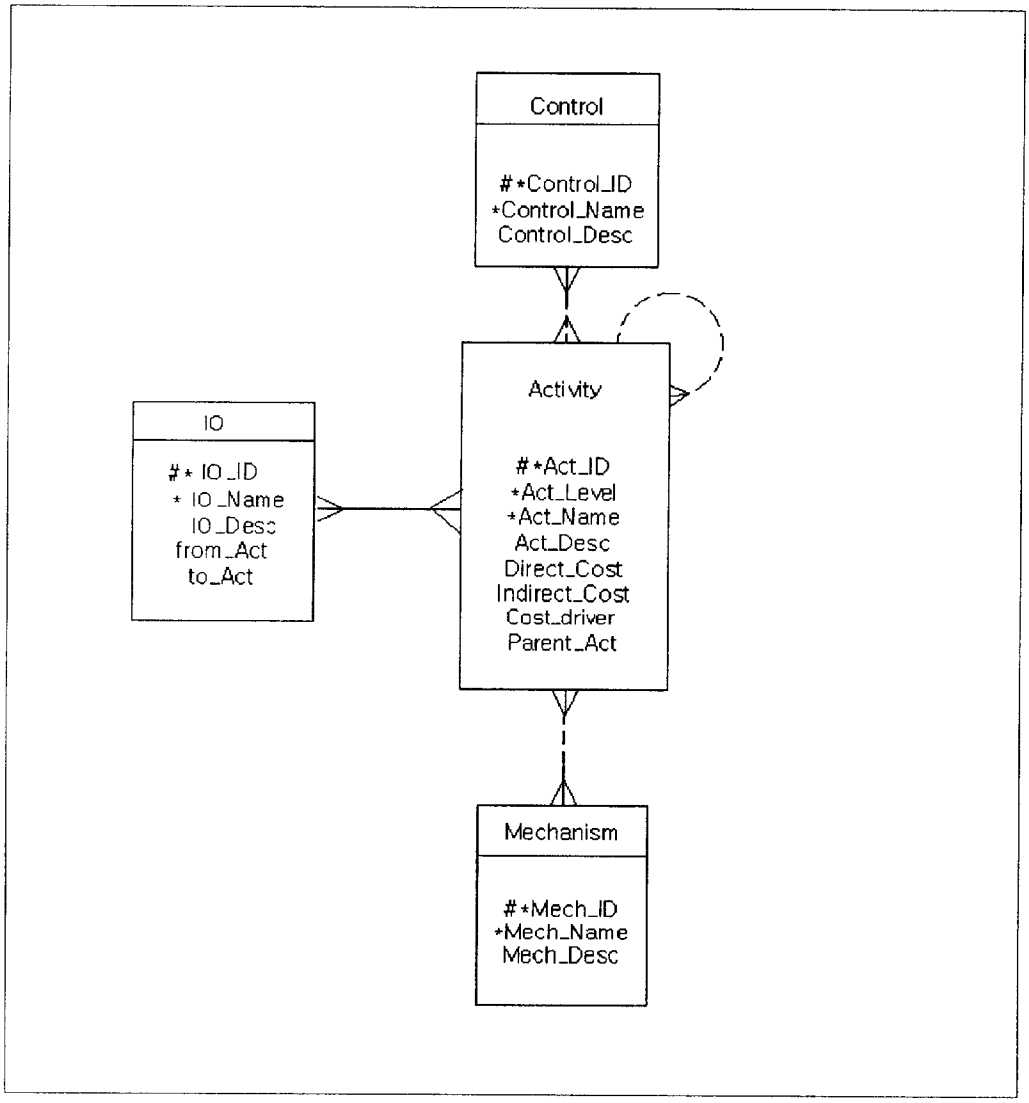
IO 개체는 IDEF0에서의 입력(input)과 출력(output)을 하나로 합친 개체이다. 선행 활동에서 나온 출력이 곧 후행 활동의 입력에 해당하므로 출력과 입력은 IO라는 하나의 개체로 파악될 수 있다. 그러므로 입력과 출력을 하나로 묶어서 IO라는 하나의 개체를 도출하였다. IO 개체에는 IO번호(io_id), IO명(io_name), IO설명(io_desc), 선행활동(from_act), 후행활동(to_act)으로 구성되어 있다. IO번호(io_id)는 IO개체의 주키(primary key)이다. 선행활동(from_act) 속성은 IO가 출력으로 나오게 되는 활동개체의 활동번호(act_id)를 참조하며, 후행활동(to_act) 속성은 IO가 입력으로 들어가게 되는 활동개체의 활동번호(act_id)를 참조한다.

컨트럴(control) 개체는 주키로 컨트럴번호(control_id)를 가지며 컨트럴명(control_name), 컨트럴설명(control_desc)이라는 속성들을 가지고 있다.

마찬가지로 매커니즘(mechanism) 개체는 주키로 매커니즘번호(mech_id)를 가지고 매커니즘명(mech_name)과 매커니즘설명(mech_desc)이라는 속성들을 가진다.

본 연구의 개체관계도(Entity Relationship Diagram: ERD)는 총 4개의 개체(entity)와 그들 간의 관계(relationship)로 구성되어 있다[그림.10]. 이들 중 가장 중요한 역할을 하는 개체는 활동(activity) 개체이다. 활동 개체는 자기 자신을 포함하여 다른 모든 개체들과 관계(relationship)를 맺고 있다.

〈그림. 10〉 개체관계도 (1)


Control
#*Control_ID
*Control_Name
Control_Desc
Activity
#*Act_ID
*Act_Level
*Act_Name
Act_Desc
Direct_Cost
Indirect_Cost
Cost_driver
Parent_Act
IO
#* IO_ID
* IO_Name
IO_Desc
from_Act
to_Act
Mechanism
#*Mech_ID
*Mech_Name
Mech_Desc

활동개체는 자기자신과 관계(Recursive Relationship)를 맺고 있으며 1대다(One-to-Many)의 대응비를 가진다. 이것은 IDEF0의 탑다운(top-down) 분해기능을 반영한 것이다. 이와 같이 자기 자신과 관계(Recursive Relationship)를 맺음으로서 계층구조를 표현할 수 있다[16]. 즉 활동개체의 상위활동 속성을 참조함으로서 계층구조의 파악이 가능하다. 그리고 활동개체는 io 개체와 다대다(Many-to-Many)의 대응비를 가진다. 이것은 하나의 활동에 여러 개의 io가 존재할 수 있기 때문이며, 선행활동에서의 출력된 하나의 개체가 여러 개의 후행활동에 각각 입력될 수 있기 때문이다.

컨트럴과 매커니즘 개체는 활동 개체에 대하여 각기 다대다(Many-to-Many)의 대응비를 가진다. 왜냐하면 하나의 활동 개체는 여러개의 컨트럴과 매커니즘을 가질 수 있고, 컨트럴과 매커니즘도 각기 여러개의 활동 개체와 관계될 수 있기 때문이다. 예를 들어 [그림 8]의 주문획득 활동의 프로세스 모델을 보면, '고객요구해결방안 조사(1.2)'라는 활동에는 '기술자료', '재고자료', '생산능력정보'라는 세 개의 컨트럴이 들어간다. 한편 위의 세가지 컨트럴은 다른 활동에도 컨트럴로 작용할 수 있다. 그러므로 다대다의 대응비가 성립한다. 마찬가지로 '가격 및 인도일자결정(1.3)'이라는 활동에는 '통신도구'와 '정보시스템'이라는 두 개의 매커니즘이 작용하고 있으며, 이 두개의 매커니즘은 각기 다른 활동에도 매커니즘으로 사용될 수 있으므로 각기 다대다의 대응비를 갖는다.

〈그림. 10〉 의 개체관계도에서 활동 개체와 관계를 맺고 있는 컨트롤 개체와 매커니즘 개체가 각기 다대다의 대응비를 가지고 있다. 이와 같이 다대다(Many-to-Many)의 관계는 새로운 개체를 만들어서 다대다(Many-to-Many)의 관계를 각기 일대다(One-to-Many)의 관계로 풀어주어야 한다[17].

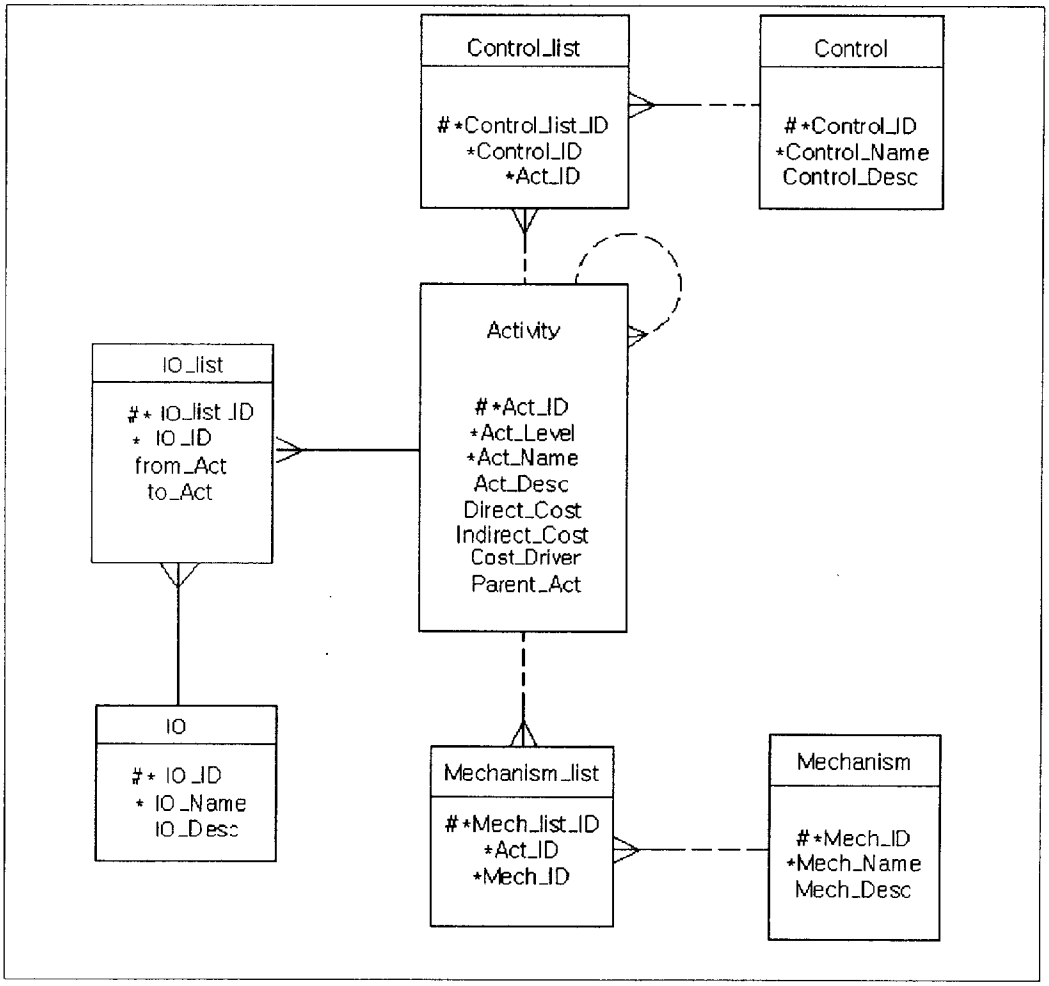
〈그림.11〉 에서 보이는 바와 같이 다대다 (Many-to-Many)의 대응비를 풀기위해 io_list, Control_list 그리고 Mechanism_list 개체를 만들어 각기 일대다(One-to-Many)의 대응비로 변환하였다.

---

16) Oracle Education Center, *Data Modeling and Relational Database Design*, Chapter8 Modeling Hierarchies

17) Oracle Education Center, *Data Modeling and Relational Database Design*, Chapter7 Resolving Many-to-Many Relationship

〈그림. 11〉 개체관계도(2)


Control_list
#*Control_list_ID
*Control_ID
*Act_ID
Control
#*Control_ID
*Control_Name
Control_Desc
Activity
#*Act_ID
*Act_Level
*Act_Name
Act_Desc
Direct_Cost
Indirect_Cost
Cost_Driver
Parent_Act
IO_list
#* IO_list_ID
* IO_ID
from_Act
to_Act
IO
#* IO_ID
* IO_Name
IO_Desc
Mechanism_list
#*Mech_list_ID
*Act_ID
*Mech_ID
Mechanism
#*Mech_ID
*Mech_Name
Mech_Desc

〈그림. 11〉 의 개체관계도를 구체적으로 살펴보면, control_list 개체는 control_list_id를 주키로 가지며 control_id와 act_id라는 속성을 가진다. control_id는 컨트럴 개체의 control_id를 참조하며, act_id는 activity 개체의 act_id를 참조한다. 그리고 mech_list 개체도 mech_list_id를 주키로 가지며 mech_id와 act_id라는 속성을 가지고 있다. mech_id는 매커니즘 개체의 mech_id를 참조하며, act_id는 activity 개체의 act_id를 참조하고 있다. 마찬가지로 io_list 개체를 설정하여 io 개체와 activity개체의 다대다의 관계를 일대다의 관계로 변환하였다.

3.3 내부 스키마(Internal Schema)

개념적 설계 단계에서 개체관계도(Entity Relationship Diagram: ERD)로 표현된 시스템의 요구사항을 물리적인 데이터베이스로 구현하기 위해서는 관계형 데이터베이스에 맞게 적절히 사상(mapping)되어야 한다. 〈표. 9〉 는 ERD로부터 사상된 관계형 데이터 정의어(Data Definition Language: DDL)이다. 데이터 정의어는 데이터베이스에 저장할 데이터의 구조를 지정하는 언어를 일컫는다[Date, 2000].

〈표. 9〉 데이터 정의어(DDL)

```
CREATE TABLE activity (
        act_id                  VARCHAR2(20)   NOT NULL,
        act_level               VARCHAR2(20),
        act_name                VARCHAR2(20)   NOT NULL,
        act_desc                VARCHAR2(20),
        direct_cost             NUMBER(10),
        indirect_cost           NUMBER(10),
        cost_driver             VARCHAR2(10),
        parent_act_id           VARCHAR2(20) );
CREATE TABLE io (
        io_id                   VARCHAR2(20)   NOT NULL,
        io_name                 VARCHAR2(20)   NOT NULL,
        io_desc                 VARCHAR2(20),
        CONSTRAINT io_id_pk PRIMARY KEY(io_id) );
```

```
CREATE TABLE io_list (
        io_id_list                  VARCHAR2(20)   NOT NULL,
        io_id                       VARCHAR2(20),  NOT NULL
        from_act                    VARCHAR2(20),
        to_act                      VARCHAR2(20),
        CONSTRAINT io_id_list PRIMARY KEY(io_id_list),
        CONSTRAINT io_id_fk FOREIGN KEY (io_id)
                                    REFERENCES io(io_id),
        CONSTRAINT from_act_fk FOREIGN KEY (from_act)
                                    REFERENCES activity(act_id),
        CONSTRAINT to_act_fk FOREIGN KEY (to_act)
                                    REFERENCES activity(act_id) );
CREATE TABLE control (
        control_id                  VARCHAR2(20)   NOT NULL,
        control_name                VARCHAR2(20)   NOT NULL,
        control_desc                VARCHAR2(20),
        CONSTRAINT control_id_pk PRIMARY KEY(control_id),
        );

CREATE TABLE control_list (
        control_list_id             VARCHAR2(20)   NOT NULL,
        control_id                  VARCHAR2(20)   NOT NULL,
        act_id                      VARCHAR2(20)   NOT NULL,
        CONSTRAINT control_list_id_pk PRIMARY KEY(control_list_id),
        CONSTRAINT control_id_fk FOREIGN KEY (control_id)
                                    REFERENCES control(control_id),
        CONSTRAINT act1_id_fk FOREIGN KEY (act_id)
                                    REFERENCES activity(act_id));

CREATE TABLE mechanism (
        mech_id                     VARCHAR2(20)   NOT NULL,
        mech_name                   VARCHAR2(20)   NOT NULL,
        mech_desc                   VARCHAR2(20),
        CONSTRAINT mech_id_pk PRIMARY KEY(mech_id));
```

```
CREATE TABLE mech_list (
        mech_list_id                VARCHAR2(20)    NOT NULL,
        act_id                      VARCHAR2(20)    NOT NULL,
        mech_id                     VARCHAR2(20)    NOT NULL,
        CONSTRAINT mechanism_list_id_pk PRIMARY KEY(mech_list_id)
        CONSTRAINT mech_id_fk FOREIGN KEY (mech_id)
                                    REFERENCES mechanism(mech_id),
        CONSTRAINT act2_id_fk FOREIGN KEY (act_id)
                                    REFERENCES activity(act_id));
```

아래의 표는 데이터베이스에 생성된 테이블과 그 속성에 대한 상세이다.

〈표.10〉 activity 테이블

필드명	데이터타입	비고
act_id	Varchar2	활동에 부여된 ID, Primary Key
act_level	Varchar2	IDEF0 모델에서의 활동 번호
act_name	Varchar2	활동명
act_desc	Varchar2	활동에 대한 설명
direct_cost	Number	직접비용
indirect_cost	Number	간접비용
cost_driver	varchar2	원가동인
parent_act	Varchar2	IDEF 계층구조상 상위 활동의 act_id를 참조

〈표. 11〉 IO 테이블

필드명	데이터타입	설명
io_id	Number	IO에 부여된 ID, Primary Key
io_name	Varchar2	IO명
io_desc	Varchar2	IO에 대한 설명

〈표. 12〉 IO_List 테이블

필드명	데이터타입	설명
IO_List_ID	Number	IO에 부여된 ID Primary Key
IO_ID	Varchar2	IO명 IO 테이블의 IO_ID 참조
from_Act	Varchar2	출력으로 나온 활동 Activity 테이블의 Act_ID 참조
to_Act	Varchar2	입력으로 들어가는 활동 Activity 테이블의 Act_ID 참조

〈표. 13〉 control 테이블

필드명	데이터타입	설명
control_id	Number	컨트럴에 부여된 ID, Primary Key
control_name	Varchar2	컨트럴명
control_desc	Varchar2	컨트럴에 대한 상세

〈표. 14〉 control_list 테이블

필드명	데이터타입	설명
control_list_id	Number	컨트럴 리스트에 부여된 ID, Primary Key
control_id	Number	Control 테이블의 control_ID 참조
act_id	Varchar2	Activity 테이블의 Act_ID 참조

〈표. 15〉 mechanism 테이블

필드명	데이터타입	설명
mech_id	Number	Primary Key
mech_name	Varchar2	매커니즘명
mech_desc	Varchar2	매커니즘에 대한 설명

〈표. 16〉 mechanism_list 테이블

필드명	데이터타입	설명
mech_list_id	Number	Primary Key
act_id	Varchar2	Activity 테이블의 Act_ID 참조
mech_id	Number	Mechanism 테이블의 Mech_ID 참조

제 4 장 사례연구

4.1 샘플 프로세스

본 연구의 프로세스 사례는 제조업체의 주문처리 프로세스 중 주문획득 활동[Tatsiopolous, Panayiotou, 2000]이다. 주문처리 프로세스는 고객의 요구사항에 따라 제품을 공급하며 세일즈, 원가계산, 제품개발, 생산계획 등의 활동들을 포함한다.

주문획득 활동은 주문처리 프로세스의 첫 번째 활동으로, 고객의 요청에 따라 생산능력, 원자재 가격 등을 고려하여 가격 및 배달일자를 계산해서 제안서를 발송하고 주문을 받는 활동이다. 아래의 [그림. 12]는 주문획득 활동을 IDEF0로 표현한 것이다. 본 사례 및 데이터의 출처는 Tatsiopoulos의 활동기반원가관리(activity based costing: ABC)와 프로세스 모델의 통합에 관한 연구이며, Tatsiopoulos의 연구는 IDEF0 모델 뿐만 아니라 직접비, 간접비, 원가동인 등 활동기반원가관리와 관련된 정보를 제공하고 있다.

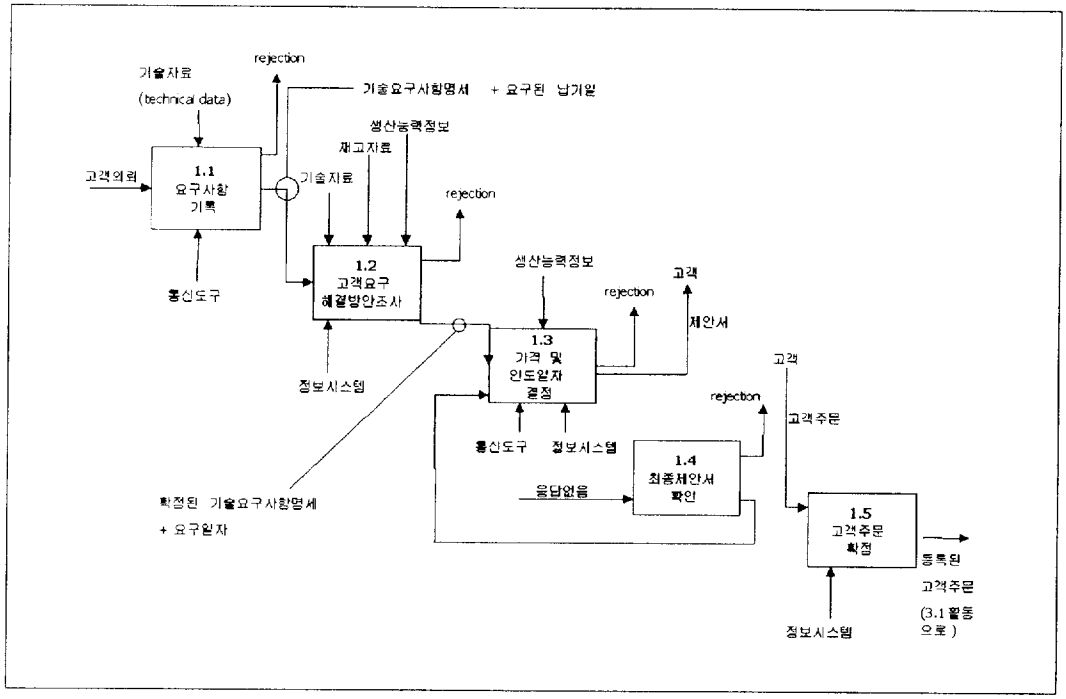
〈그림.12〉 의 내용을 설명하면 다음과 같다. 고객의뢰라는 입력이 발생하면, '요구사항 기록(1.1)' 활동은 '기술자료(technical data)'라는 컨트럴과 '통신도구'라는 매커니즘을 이용하여 '요구사항명세 + 요구된 납기일'과 'rejection'이라는 두 개의 출력을 산출한다.

'요구사항명세 + 요구된 납기일'은 '고객요구 해결방안 조사(1.2)' 활동에 입력되는데, '기술자료', '재고자료', '생산능력정보'라는 컨트럴과 '정보시스템'이라는 매커니즘을 이용하여 'rejection'과 '확정된 기술요구사항 명세 + 요구일자'라는 출력을 산출한다.

'가격 및 인도일자 결정(1.3)' 활동은 입력을 받아들여 rejection하던가 고객에게 '제안서'를 발송한다. 고객으로부터 응답이 없으면 '최종제안서확인(1.4)' 활동은 rejection하거나 '가격 및 인도일자 결정(1.3)' 활동으로 피드백한다.

고객으로부터 고객주문이 '고객주문확정(1.5)' 활동으로 입력되면 '고객주문확정' 활동은 '정보시스템' 매커니즘을 이용하여 '등록된 고객주문'을 산출하고, 이것은 3.1 활동으로 입력된다.

〈그림. 12〉 주문획득 활동의 IDEF0 모델 [Tatsiopolous, Panayiotou, 2000]



4.2 프로세스 데이터입력

본 연구에 사용된 데이터베이스는 Oracle7이다. 그리고 프로세스 모델과 관련된 데이터를 데이터베이스에 입력하는 것을 쉽게 하기 위하여 오라클 디벨로퍼를 사용하여 프로그램을 개발하였다. 이 프로그램을 통하여 데이터의 입력·수정·삭제를 용이하게 할 수 있다.

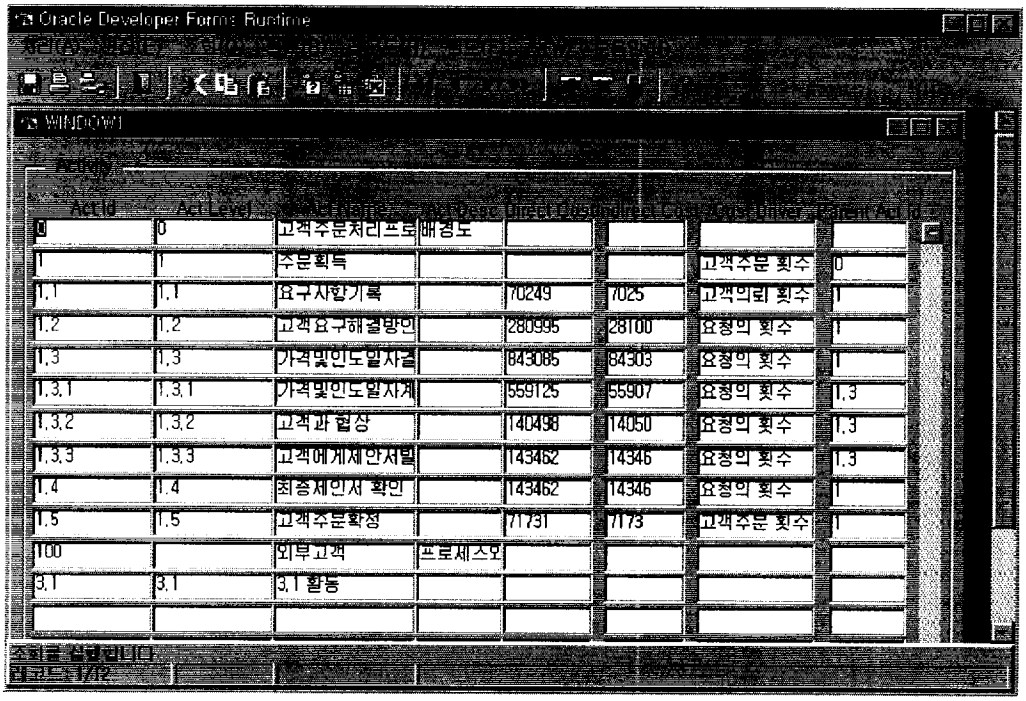
〈표. 17〉 구현환경

구 분	사 양
OS	Windows98
Database	Oracle 7.3
개발도구	Oracle Developer 6.0

본 연구에서 개발한 소프트웨어는 활동 모듈, IO 모듈, IO_list 모듈, 컨트럴 모듈, 매커니즘 모듈, control_list 모듈, mech_list 모듈로 구성되어 있으며 각각의 모듈들은 테이블별로 데이터를 입력, 조회, 수정, 삭제하는데 사용된다.

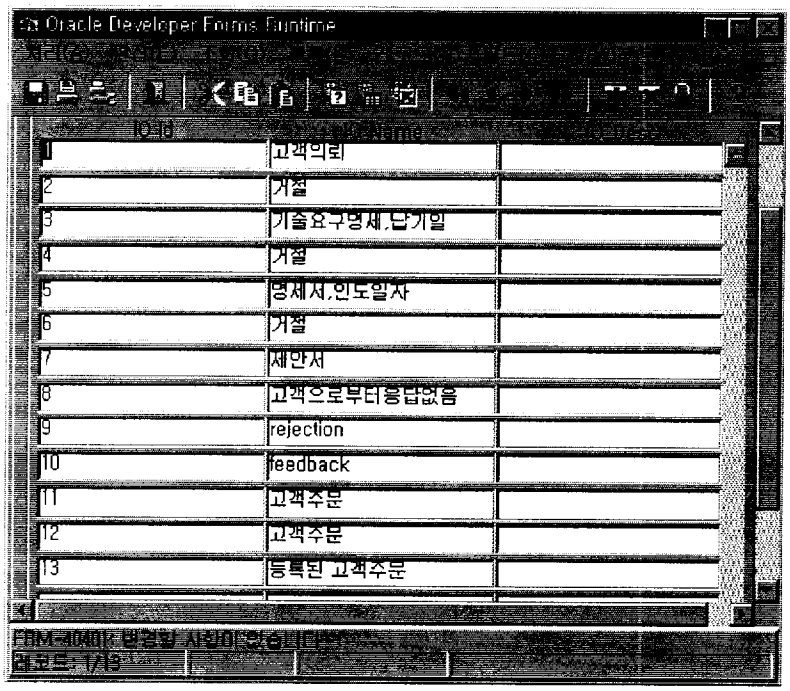
리파지토리를 사용하기 위해서는 우선 각 모듈을 이용해서 해당 데이터들을 입력해야 한다. 〈그림. 13〉 는 활동(activity) 데이터 입력의 예로서, 활동(activity) 모듈이다. 〈그림. 13〉 에서 보는 바와 같이 레코드별로 활동의 속성에 대하여 입력하고, 저장 버튼을 눌러 커밋(commit)[18]한다.

〈그림. 13〉 활동(activity) 모듈

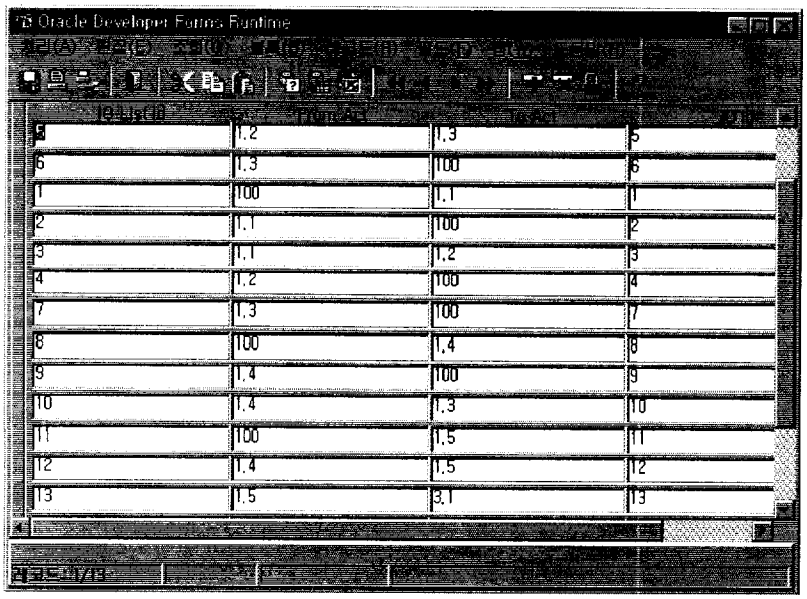


18) 데이터베이스에서 트랜잭션(transaction)이 성공적으로 이루어진 경우, 트랜잭션이 실행완료되었다고 한다. SQL에서는 성공적으로 이루어진 트랜잭션의 내용을 데이터베이스에 영구반영하기 위하여 COMMIT이란 명령을 사용한다.

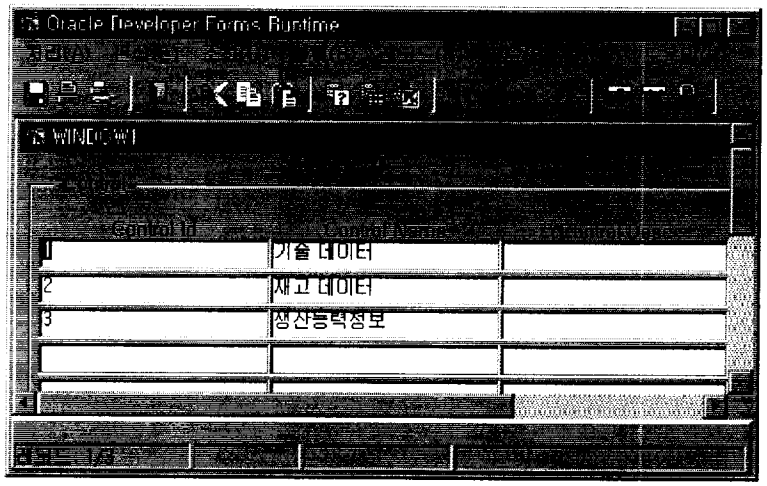
〈그림. 14〉 IO 모듈



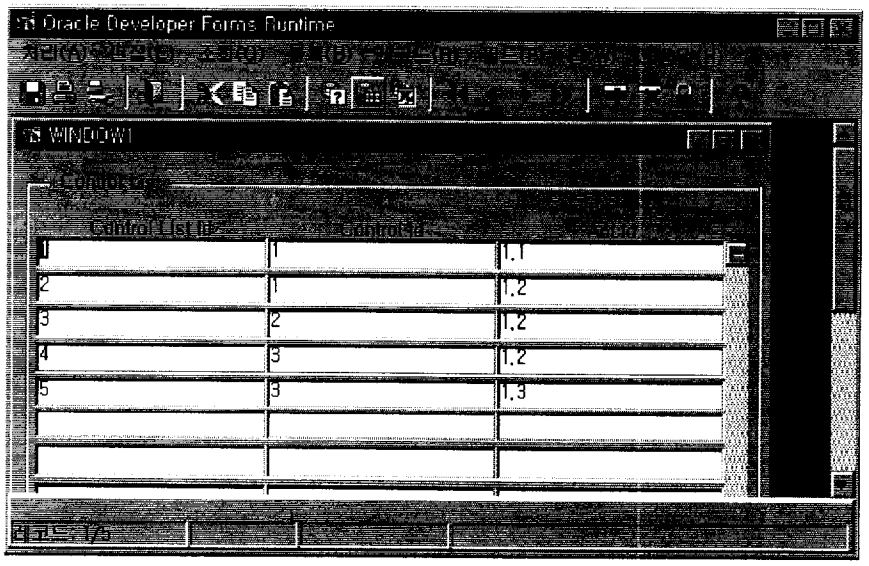
〈그림. 15〉 io_list 모듈



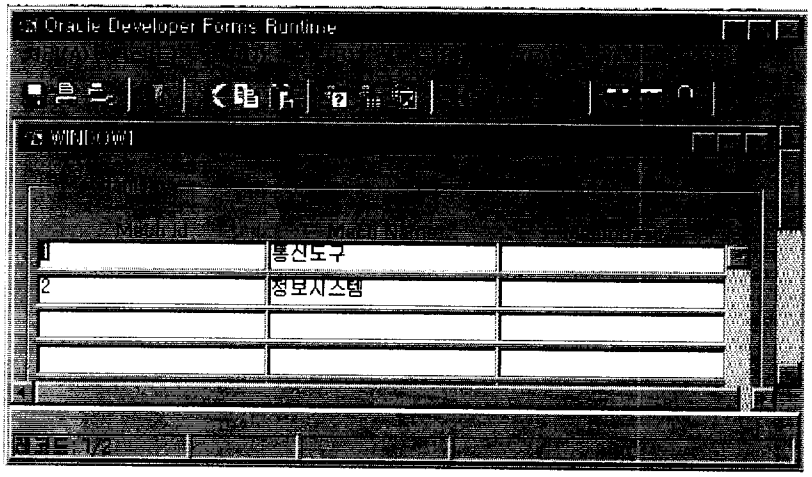
〈그림. 16〉 컨트럴 모듈



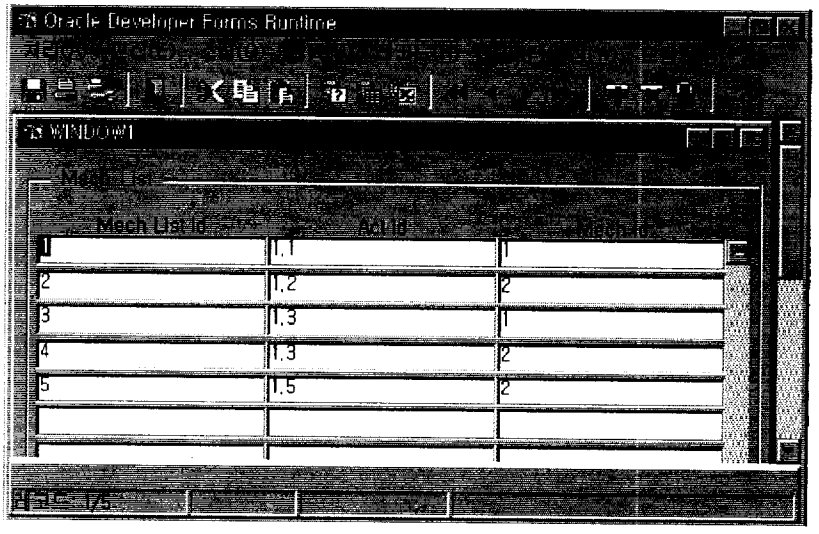
〈그림. 17〉 컨트럴 리스트 모듈



〈그림. 18〉 매커니즘 모듈



〈그림. 19〉 매커니즘 리스트 모듈

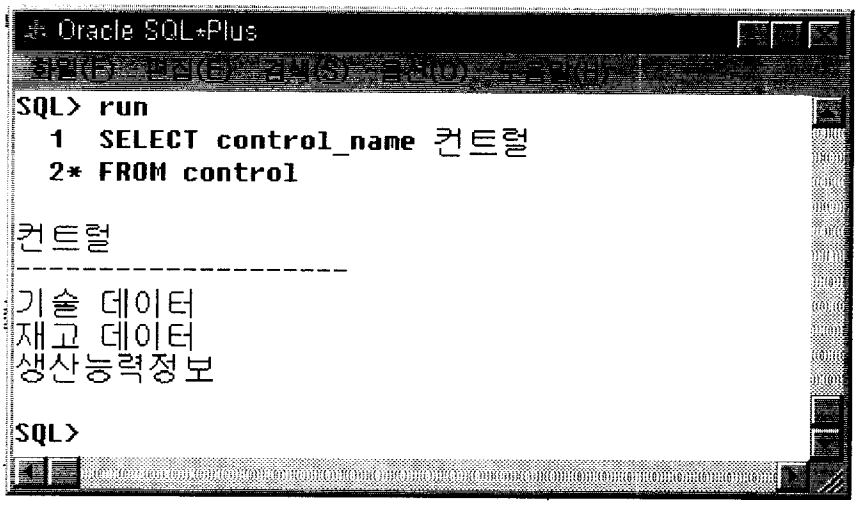


4.3 프로세스 분석

본 연구의 프로세스 리파지토리는 SQL을 이용하여 분석이 가능하다. 그러나 프로세스를 분석할 수 있는 근거로는 프로세스 모델, 직접비, 간접비 그리고 원가동인 밖에 없으므로 초보적인 질의만을 제시한다.

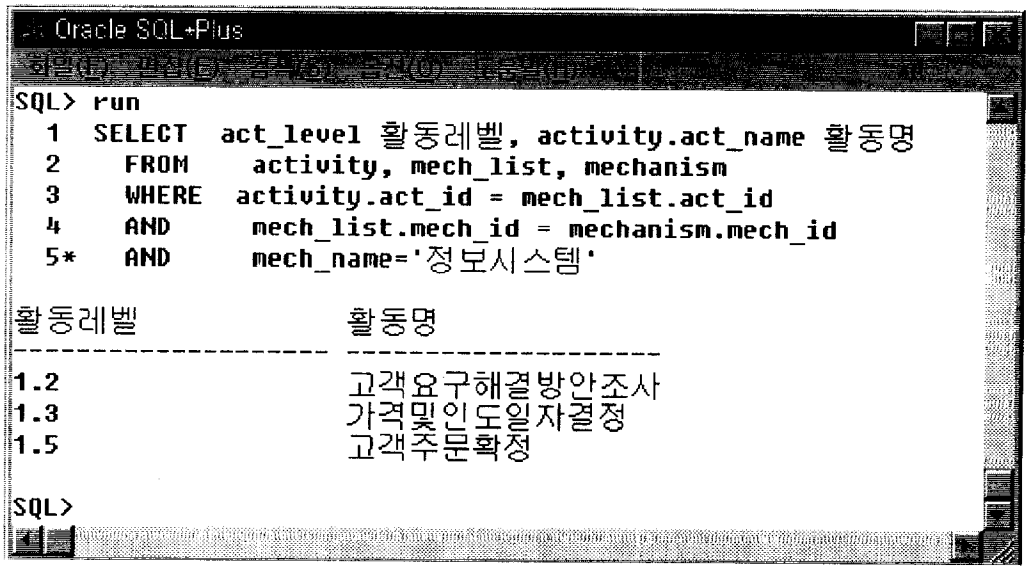
〈그림. 20〉 에 있는 질의문는 프로세스에 사용되고 있는 컨트럴의 목록을 출력하는 것이다. 문서나 이미지와 같은 비구조적 정보로 표현된 프로세스 모델은 이와 같은 분석에 대해서 일일이 프로세스 모델을 읽어나가는 방법밖에 없다. 그러나 본 연구의 리파지토리는 간단한 질의문으로 조회가 가능하다.

〈그림. 20〉 질의 예. 1



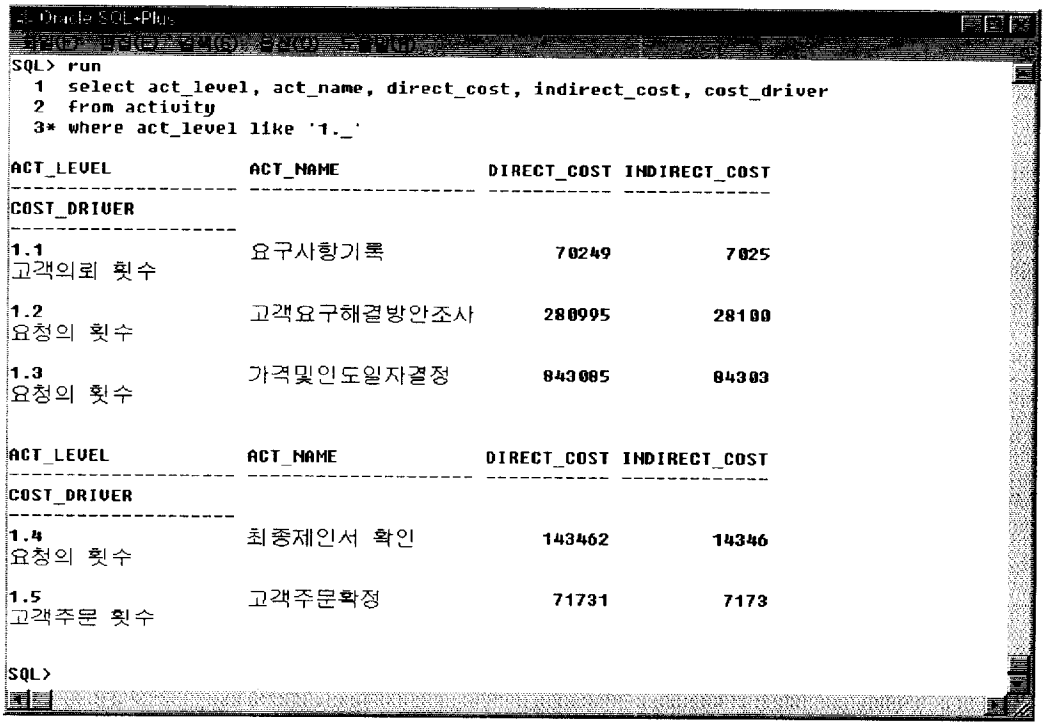
〈그림. 21〉 은 매커니즘으로 '정보시스템'을 사용하는 활동들의 목록을 출력하는 질의문이다. 프로세스 리엔지니어링의 노력에서 정보시스템은 중요한 역할을 한다 [Hammer, 1990; Davenport & Short, 1990; Malone, 1992; Malhotra, 1998]. 그러므로 프로세스 리파지토리를 이용한 벤치마킹을 수행할 경우 타 조직에서 정보시스템을 어떻게 사용하고 있는지 분석하는 것은 매우 의미가 있다.

〈그림. 21〉 질의 예. 2



〈그림. 22〉 는 레벨1에 있는 활동들의 직접비, 간접비, 원가동인을 출력하는 질의문이다. 활동기반원가관리(Activity Based Costing: ABC)에서는 전통적인 회계 기준과는 다르게 활동을 단위로 하여 비용을 산출한다[유관희, 1998]. 이와같이 ABC를 적용하여 비용을 측정하게 되면 어떠한 활동에 필요한 비용과 비용을 발생시키는 원가동인에 관한 정보를 얻을 수 있다. 이러한 정보들은 비즈니스 리엔지니어링 수행시 어디에 집중을 하여야 하는지 결정하는데 도움을 줄 수 있다 [Nyamekye, 2000].

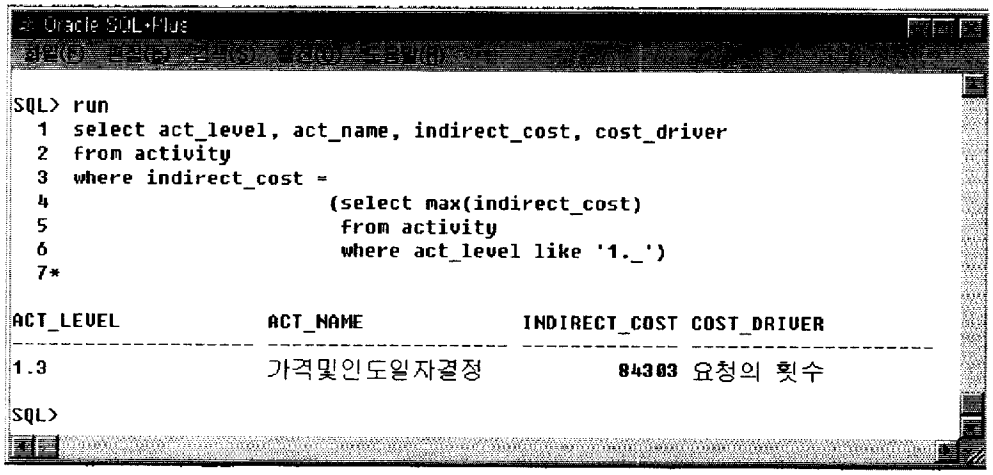
〈그림. 22〉 질의 예. 3



```
SQL> run
  1  select act_level, act_name, direct_cost, indirect_cost, cost_driver
  2  from activity
  3* where act_level like '1._'

ACT_LEVEL            ACT_NAME             DIRECT_COST INDIRECT_COST
-------------------- -------------------- ----------- -------------
COST_DRIVER
--------------------
1.1                  요구사항기록               70249          7025
고객의뢰 횟수

1.2                  고객요구해결방안조사       280995         28100
요청의 횟수

1.3                  가격및인도일자결정         843085         84303
요청의 횟수

ACT_LEVEL            ACT_NAME             DIRECT_COST INDIRECT_COST
-------------------- -------------------- ----------- -------------
COST_DRIVER
--------------------
1.4                  최종제안서 확인            143462         14346
요청의 횟수

1.5                  고객주문확정               71731          7173
고객주문 횟수

SQL>
```

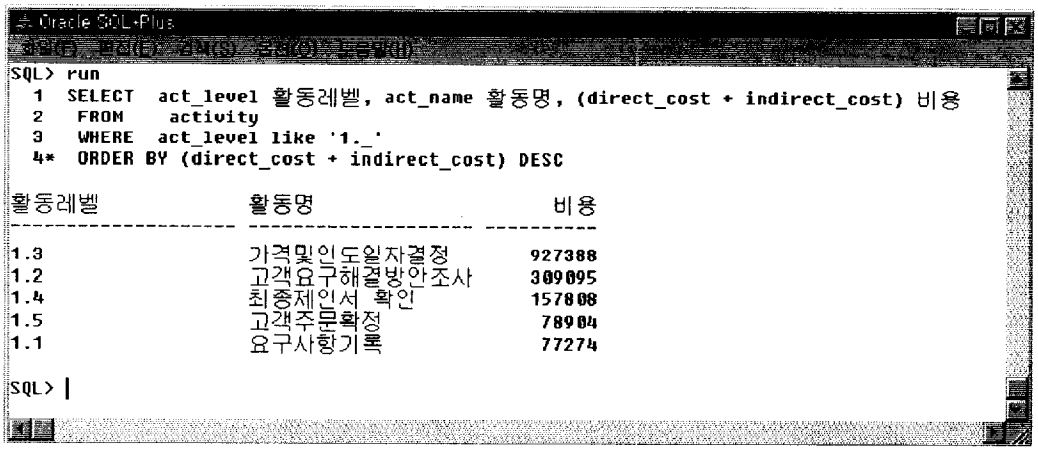
〈그림. 23〉 은 레벨1에 있는 활동 중에서 간접비가 가장 큰 활동의 활동레벨, 활동명, 간접비, 원가동인을 출력하는 질의문이다.

〈그림. 23〉 질의 예. 4



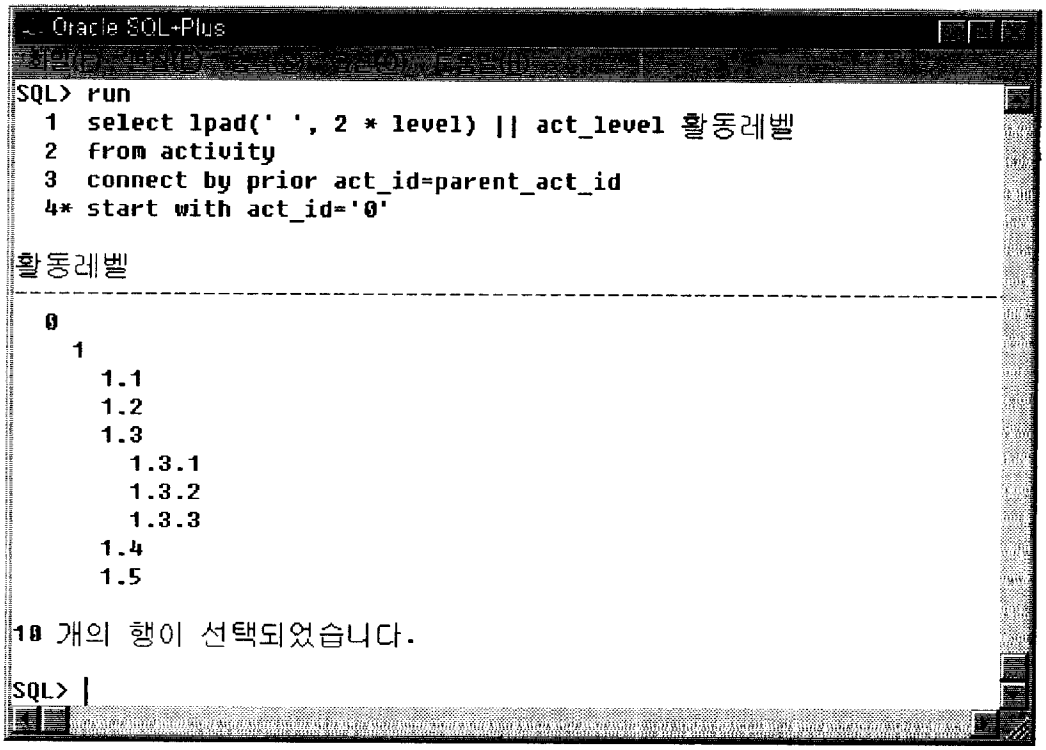
〈그림. 24〉 는 레벨1에서 활동레벨, 활동명, 그리고 직접비와 간접비의 합을 비용이 큰 순서로 출력하는 질의문이다. 활동 테이블의 속성 중 직접비(direct_cost)와 간접비(indirect_cost) 속성을 이용하여 결과를 구할 수 있다.

〈그림. 24〉 질의 예. 5



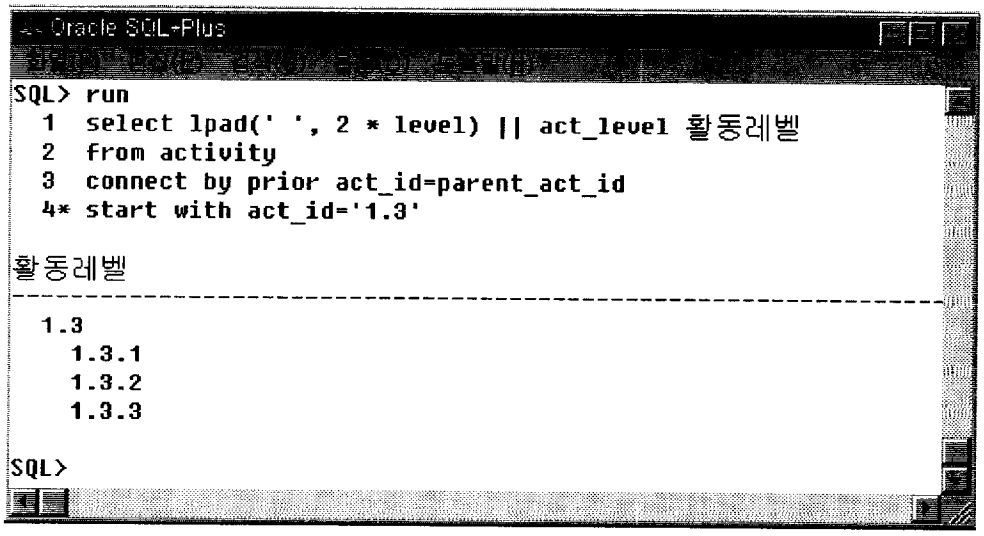
〈그림. 25〉 는 프로세스를 계층적으로 분해하여 표현하는 기능을 가진 IDEF0의 계층적 분해 에 대한 계층적 질의(hierarchical queries) 예제이다. 'CONNECT BY PRIOR'는 부모노드(parent node)로부터 자식노드(child node)로 검색한다는 문장이며, 'START WITH'는 검색시작점을 지정하는 문장이다.

〈그림. 25〉 질의 예. 6



〈그림. 26〉 은 '질의 예.6'과 동일한 질의문이지만 전체 활동이 아닌 '1.3'활동을 검색시작점으로 지정하여 질의하였다.

〈그림. 26〉 질의 예. 7



위에서 사례로 보인 질의문과 같은 방법으로 본 프로세스 리파지토리는 분석이 가능하다. 그러나 초보적인 질의만 가능하였을 뿐 심도 있는 분석은 시도하지 못하였다. 왜냐하면 본 연구의 스키마에서는 프로세스의 측정을 지원하기 위해 단순히 직접비와 간접비 그리고 원가동인만을 활동의 속성에 추가하였기 때문이다. 그리고 본 연구의 범위에 프로세스 분석은 포함되어있지 않다. 체계적인 프로세스 분석을 위하여 IDEF0와 ABC의 통합에 관한 연구들에 기초한 스키마 확장이 요구된다.

제 5 장 결론

5.1 연구의 요약

본 연구는 선진 프로세스(Best Practice)를 저장하기 위한 프로세스 리파지토리의 스키마를 설계하였으며, 프로세스 모델을 구조적 데이터의 형태로 관계형 데이터베이스에 저장함으로서 여러 효과를 확인하였다.

우선 프로세스의 비교·분석 기능이 강화되었다. 리엔지니어링 및 벤치마킹에 있어서 프로세스를 분석하는 것은 매우 중요한 활동이다. 다양한 사례를 평가하고 적합한 프로세스를 선택하기 위해서는 프로세스의 우열을 가리는 것이 필요하기 때문이다. 그러나 기존의 리파지토리 시스템은 단지 참조모형의 제공에 급급했을 뿐 프로세스 분석을 위한 기능은 많지 않았다. 본 연구에서는 프로세스 모델을 구조적 데이터로 관계형 데이터베이스에 저장함으로서 프로세스 모델 관련자료의 계량적 저장을 가능하게 하여 향상된 프로세스 비교·분석의 가능성을 제시하였다. 즉 프로세스의 특성을 계량적으로 나타내고 수치의 비교를 통하여 프로세스간 비교가 가능한 것이다.

그리고 저장된 데이터의 수정이나 업데이트가 용이하다. 도식으로 저장된 프로세스는 저장시의 비효율성뿐만 아니라 관리에도 어려움이 많았다. 이것은 프로세스가 이미지 형식의 파일로 저장되어 있기 때문에 프로세스의 조그마한 변화에도 모든 프로세스를 다시 작성해야 했다. 그러나 관계형 데이터베이스에 구조적 데이터로 저장된 프로세스 모델 데이터는 계량적이고 문자화된 데이터이므로, 변경된 부분만 수정하면 되므로 유지 관리가 용이하다.

5.2 연구의 의의

본 연구는 다음과 같은 의의를 가지고 있다.

첫째, 프로세스 리파지토리에 비구조적 정보로 저장되어 오던 프로세스 모델을

구조적 형태로 관계형 데이터베이스에 저장하기 위한 스키마를 설계하였다.

둘째, 구조적 데이터로 관계형 데이터베이스에 저장된 프로세스 모델은 프로세스 분석의 성능을 향상시킨다.

셋째, 프로세스 모델링 기법으로 IDEF0를 사용함으로서 프로세스의 최하위 엑티버티까지 계층적으로 상세히 분석할 수 있다.

5.3 연구의 한계 및 향후 연구 과제

본 연구의 한계 및 향후 연구과제는 다음과 같다

첫째, 본 연구에서는 프로세스 리파지토리를 위한 스키마만을 제시하였을 뿐, GUI 환경에서의 프로세스 다이어그램의 입력, 조회 및 수정을 위한 자동화 도구는 개발하지 않았다. 제시한 스키마가 효과적으로 활용되기 위해서는 GUI 환경에서의 프로세스 모델 구축이 가능한 손쉬운 사용자 인터페이스의 개발이 요구된다.

둘째, 본 연구의 스키마에서는 프로세스의 측정을 지원하기 위하여 단순히 직접비와 간접비 그리고 원가동인만을 활동의 속성에 추가하였을 뿐, 체계적인 프로세스 분석 방법을 제시하지 못하였다. 이를 극복하기 위하여 IDEF0와 ABC의 통합에 관한 기존 연구들에 기초해 스키마를 확장할 필요가 있다.

참 고 문 헌

[Alast, Hee, 1996] Alast, W., Hee, K., Business process redesign: a Petri-net-based approach, *Computers in industry*, 29, pp.15-26, 1996

[Biazzo, 2000] Biazzo, S., Approaches to business process analysis: a review, *Business Process Management Journal*, Vol.6, No.2, pp.99-112, 2000

[Carr, 1999] Carr, N. G., "A New Way to Manage Process Knowledge," *Harvard Business Review*, September - October, 1999

[Camp, 1989] Camp, R. C., *Benchmarking: The Search for Industry Best Practice That Lead to Superior Performance*, ASQC Quality Press, 1989

[Chang, 1994] Chang, R. Y., Improve process, reengineer them, or both?, *Training and Development* 48 (3), 54-58

[Cheung, Bal, 1998] Cheung, Y., Bal, J., Process analysis techniques and tools for business improvements, *Business Process Management Journal*, Vol. 4, No. 4, pp.274-290, 1998

[Clayton, 2000] Clayton, N., et al., A Study of Usability of Web-Based Software Repositories, *IEEE*, 2000

[Curtis et al., 1992] Curtis, B., Keller, M., Over, J., *Process Modelling, Comunications of the ACM*, Vol.35, No.9, pp.75-90, 1992

[Cheung & Bal, 1998] Cheung, Y., Bal, J., Process analysis techniques and tools for business improvements, *Business Process Management Journal*, Vol. 4, No.4, pp. 274-290, 1998

[Date, 2000] Date, C. J., *An Introduction to Database Systems*, Addison

Wesley, 2000

[Davenport & Short, 1990] Davenport, Th. H., Short, J. E., The new industrial engineering: Information technology and business process redesign, *Sloan Management Review* 31(4), pp. 11-27, 1990

[Davenport, 1993] Davenport, T. H., *Process Innovation: Reengineering Work through Information Technology*, Boston, MA: Harvard Business School, 1993

[Furey, 1993] Furey, T. R., A six-step guide to process reengineering, *Planning Review* 21 (2), pp.20-38, 1993

[Grover, Malhotra 1997] Grover, V., Malhotra, M. K., Business Process Reengineering: A tutorial on the concept, evolution, method, technology and application, *Journal of Operations Management*, Vol 15, 1997, pp 193-213

[Gruhn, 1998] Gruhn, V. and Schneider, M, Workflow Management Based on Process Model Repository, *IEEE*, pp. 379-388, 1998

[Hammer, 1990] Hammer, M., Reengineering work: Don't automate, obliterate, *Harvard Business Review*, July/August, pp. 104-112, 1990

[Hammer & Champy(1993)] *Reengineering the Corporation: A Manifesto for Business Revolution.* Harper Business, New York, 1993

[Hammer, 1997] Hammer, M., *Beyond Reengineering: How the Processed - Centered Organization is Changing Our Work and Our Lives*, Harper Business, 1997

[Harrison & Pratt, 1992] Harrison, D. B., Pratt, M.D., A Methodology for reengineering business, *Planning Review* 21 (2), pp. 6-11, 1992

[Hunt, 1996] Hunt, V. D., *Process Mapping: How to Reengineer your*

Business Process, John Wiley&Son, Inc., 1996

[Jacobson, 1995] Jacobson, I., *The Object Advantage: Business Process Reengineering with Object Technology*, Addison Wesley, Massachusetts, 1995

[Jarzabek, Ling, 1995] Jarzabek, S., Ling, T. W., Model-based Design of Tools for Business Understanding and Re-engineering, *IEEE*, pp.324-333, 1995

[Johansson et al., 1993] Johansson, H. J., McHugh, P., Pendleburv, J., Wheeler Ⅲ, W. A., *Business Process Reengineering: Breakpoint Strategies for Market Dominance*, John Wiley and Sons, New York

[Kaplan & Murdoch, 1991] Kaplan, R. S., Murdoch, L., Core process redesign, *The McKinsey Quarterly* 2, pp.27-43, 1991

[Kellner et al, 1998] Kellner, M. I. Riddle, W. E., and Tomal, T., Process Guides: Effective Guidance for Process Participants, *5th International Conference on the Software Process*, 1998

[Maelao, Pidd, 2000] Maelao, N., Pidd, M., A conceptual framework for understanding business processes and busienss process modelling, *Journal of Information Systems* 10, pp.105-129, 2000

[MacGauhey, 1993] MacGoahey, R. E., Gibson, M., The Repository/ Encyclopedia: Essential to Information Engineering and Fully integrated CASE, *Journal of Systems Mangement*, 44(3), 1993

[Malhotra, 1998] Malhotra, Y., Business Process Redesign: An Overview, *IEEE Engineering Management Review*, vol. 26, no. 3, Fall, 1998

[Malone, 1992] Malone, T. W., Rockart, F., Information Technology and the New Organization, *IEEE*, pp.636-642

[Malone, 1993] Malone, T. W., Crowston, K., Lee, J., Tools for Inventing Organizations: Toward a Handbook of Organizational Process, *IEEE*, 1993

[Malone, 1999] Malone, T. W., Crowston, K., Lee, J., Tools for Inventing Organizations: Toward a Handbook of Organizational Process, *Management Science* 45(3) pp. 425-443, 1999

[Martin, 1989] Martin, J., *Information Engineering: A Trilogy*, Vol.1-3. Englewood Cliffs, NJ: Prentice-Hall, 1989

[Nyamekye, 2000] Nyamekye, Kofi, New tool for business process re-engineering, *IIE Solutions*, Vol. 32, Issue 3, pp.36-42, 2000

[Petrozzo & Stpper, 1994] Petrozzo, D.P., Stepper, J.C., *Successful Reengineering*, Van Nostrand Reinhold, New York

[Richman & Konntz, 1993] Richman, T., Konntz, C., How Benchmarking Can Improve Business Reengineering, *Planning Review*, vol. 21, no. 6, 1993

[Shamkant, 1992] Shamkant, B. N., Evolution of Data Modeling for Databases, *Communications of the ACM*, Vol.35, No.9, 1992

[Shapiro, 1994] Shapiro, R., M., Integrating BPR with Image-Based Work Flow, Proceedings of the 1994 Winter Simulation Conference

[Shimizu, Sahara, 2000] Shimizu, Y., Sahara, Y., A supporting system for evaluation and review of business process through activity-based approach, *Computers and Chemical Engineering*, 24, 2000, 997-1003

[Szykman, 2000] Szykman, S., et. al., Design Repositories: Engineering Design's New Knowledge Base, *IEEE Intelligent Systems*, pp.48-55, 2000

[Tatsiopolous, Panayiotou, 2000] Tatsiopolous, I. P., Panayiotou, N., The

integration of activity based costing and enterprise modeling for reengineering purposes, *International Journal of Production Economics*, 66, 2000, pp.33-44

[Vantrappen, 1992] Ventrappen, H., Creating customer value by streamlining business process, *Long Range Planning* 25 (1), pp.53-62, 1992

[Wang, 1994] Wang, S., "OO Modeling of Business, *Information System Management*, pp.36-43, 1994

[Watson, 1992] Watson, G. H., *The Benchmarking Workbook: Adapting Best Practices for Performance Improvement*, Productivity Press, 1992

[Yu & Wright, 1997] Yu, B., Wright, D. T., Software tools supporting business process analysis and modelling. *Business Process Management Journal* 3(2), pp.133-150, 1997

[김효석 외, 2000] 김효석, 홍일유, *디지털 경제 시대의 경영정보시스템*, 2000

[김훈태, 2001] 김훈태, 제조분야 업무프로세스 표준모형, 기업정보화지원센터 연구보고서

[신홍철 외, 1994] 신홍철, 변지석, 이경주, 윤순봉, 이범일, *한국의 벤치마킹*, 사계절, 1994

[유관희, 1998] 유관희, *전략경영을 위한 원가관리회계*, 박영사, 1998

[이희석 외, 1999] 이희석, 이제, 이충석, 조창래, 손주찬, 백종명, 전사적 자원관리 시스템을 위한 기업 리파지토리 구축, *경영정보학연구*, 1999

[조윤호 외, 1998] 조윤호, 김재경, 김성희, Role 개념에 근거한 비즈니스 프로세스 시뮬레이션 모형 구축 및 분석, *경영정보학연구*, 제8권 제2호, pp. 69-83, 1998

[홍정완, 2001] 홍정완, 산업정보의 지식화를 위한 업무 프로세스 표준화 방안연

구, 기업정보화지원센터 연구보고서

Oracle Education Center, *Data Modeling and Relational Database Design*

Oracle Education Center, *Introduction to Oracle: SQL and PL/SQL*

Oracle Eduction Center, *Oracle Developer: Build Forms*

IDEF 홈페이지, http://www.idef.com

MIT Process Handbook Project 홈페이지, http://ccs.mit.edu/ph/

Phios 홈페이지, http://www.phios.com

Prince2 홈페이지, http://ww.prince2.com/managementoverview.html/

기업정보화지원센터 프로세스 라이브러리 홈페이지, http://www.itr.re.kr/process

감사의 글

이렇게 석사과정 생활을 무사히 마칠 수 있었던 것은 많은 분들의 사랑과 관심이 있었기 때문입니다.

무엇보다도 먼저 석사 과정 내내 끊임없는 관심과 배려로 알찬 결실을 맺을 수 있도록 지도와 격려를 해주신 지도교수이신 이재정 교수님께 깊은 감사의 말씀을 올립니다.

본 논문이 완성되기까지 많은 조언으로 심사해 주신 어윤양 교수님, 김하균 교수님 그리고 학업의 전진에 도움을 주신 한재호 교수님, 염창선 교수님께 진심으로 감사 드립니다.

그리고 바쁘신 와중에도 많은 도움을 주신 국민대학교의 정승렬 교수님께 진심으로 감사 드립니다.

항상 마음의 도움을 주고 격려해주신 박문규 선생님, 박정재 선배님, 그리고 (주)MTT의 김창관 사장님께 감사의 마음을 전합니다.

항상 친절하게 도움을 준 성봉호, 문준철, 송유광, 유동욱, 이기성, 최영기 후배와 경영정보학과의 모든 후배들에게 고마움을 전합니다.

항상 곁에서 사랑과 관심으로 보아준 사랑하는 여자친구 성희에게도 감사의 마음을 전하고 싶습니다.

무엇보다도 어려움 속에서도 학업을 마칠 수 있도록 사랑과 이해로 돌보아주신 아버님과 어머님께 모든 영광을 돌립니다. 부모님의 헌신적인 노력으로 인해 학업을 마무리 지울 수 있었습니다. 그리고 형님과 누님들 우리 가족 모두와 기쁨을 함께 나누고자 합니다.

여러분들의 도움으로 작은 결실을 맺게 되었음을 항상 기억하고 앞으로 더 나은 미래를 위해 노력을 다하여 사회에 도움이 되는 사람이 되겠습니다.

2001년 12월

정 호 석