

기업 리파지토리 시스템 : 아키텍처 및 ERP 리파지토리 사례

이희석* · 서우종* · 김태훈* · 이충석* · 손명호* · 백종명** · 손주찬** · 박성진**

An Enterprise Repository System : Architecture and ERP Repository Case

Hee-Seok Lee* · Woo-Jong Suh* · Tae-Hun Kim* · Choong-Seok Lee* ·
Myung-Ho Sohn* · Jong-Myung Baek** · Joo-Chan Sohn** · Sung-Jin Park**

Abstract

A repository system has been conceived as a critical weapon for managing organizational information resources. The system can help control the heterogeneous data in a variety of CASE (Computer-Aided Software Engineering) tools. However, current repository systems have limitation in creating a synergetic effect by integrating information resources. Therefore, it is important to develop an integrative repository system, called Enterprise Repository System (ERS). This paper (i) defines ERS on the basis of a framework for repository systems, and (ii) suggests an ERS architecture and its detailed components. Finally, a real-life case of developing ERP repository system is illustrated according to the proposed architecture and components. This illustration may demonstrate the usefulness of this research for help developing an advanced repository system.

* 본 연구는 정보통신부 표준화사업과제(98-198) 및 한국과학재단 특정기술연구과제(98-0102-08-01-3)지원으로 수행되었음.

* 한국과학기술원 테크노경영대학원

** 한국전자통신연구원

1. 서론

급변하는 경영환경 하에서 새로운 정보기술의 활용 역량이 경쟁력의 주요한 원천임을 인식하여, 많은 기업들이 정보시스템의 구축을 통한 경영혁신의 노력을 경주해왔다. 그러나 이러한 노력은 전사적인 차원에서 이루어지지 못하고, 조직 기능에 따라 개별적으로 각기 이질적인 모델과 구현도구를 사용하여 이루어져 왔다. 이로 인해 시스템 상호간의 비통합적인 체계가 야기되고, 전사적인 차원에서 시스템 효과성의 한계가 지적되어 왔다. 이러한 문제점에 대한 전략적 대응 방안으로써, 다양한 기업 내 정보자산을 통합적으로 관리하고 재사용성을 증진시키기 위해 CASE(Computer Aided Software Engineering) 도구가 사용되었다. CASE 도구의 정보자산에 대한 통합적 관리 능력의 핵심은 메타데이터 관리에 있는데, 백과사전(Encyclopedia)[Martin, 1989]이라는 관리적 도구를 통하여 이루어졌다. 이 개념은 데이터베이스 시스템에서 데이터 관리를 위한 자료사전(Data Dictionary)[Moriarty, 1993]의 개념이 확장된 것이다. 자료사전이 데이터베이스 자료구조에 관한 정보로써 단순한 데이터 관리용 메타데이터를 저장하는 반면, 백과사전은 데이터베이스를 포함한 응용 시스템 개발 요소들에 관한 정보자원의 메타데이터를 다루고 있다.

그러나, 정보자산의 유형이 다양화됨에 따라, 새로운 기술 개념에 기반한 설계 및 관리 도구들의 도입이 제기되어 왔다. 이를 위한 한 방안으로써, CASE 도구들뿐만 아니라, 이질적(Heterogeneous) 정보시스템에서 다루어지는 다양한 정보자산에 대한 메타데이터를 통합적으로 관리하는 리파지토리의 유용성이 주목 받아 왔다. 통합된 메타데이터를 통해 정보자산의 관리를 통합적으로 하고자 하는 목적에서, 이질적

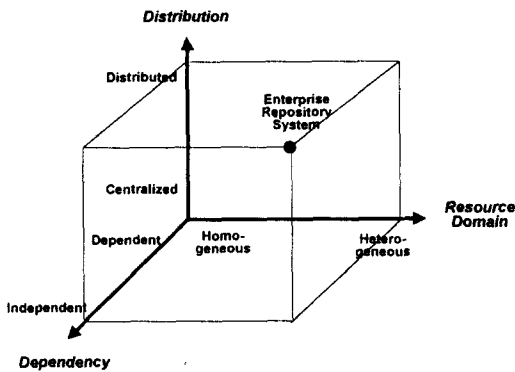
정보시스템 간 또는 CASE 도구들간의 미들웨어로서의 역할은 매우 중요한 것으로 인정되고 있다. 그러나, 현실에서는 리파지토리 시스템이 전사적 차원에서 다양한 정보자산들을 수용하지 못하고, 특정 정보자산을 중심으로 제한적인 기능 위주로 개발되는 경향이 있다. 이러한 점은 리파지토리 시스템에 대한 구축 경험이 아직 부족하고, 개념의 구체적 내용들에 대한 표준적인 정립이 아직 미흡하기 때문이다.

따라서, 본 논문에서는 그 동안 제기되어 온 리파지토리 시스템의 유용성을 총체적으로 수용할 수 있는 기업 리파지토리 시스템(Enterprise Repository System : ERS)에 대한 표준적인 개념을 제시하고자 한다. 기업 리파지토리 시스템은 기업의 모든 형태의 정보 및 지식 자산의 통합적 관리를 지원할 수 있어야 하며, 이를 위해 메타데이터를 기반으로 각종 CASE 도구 및 정보시스템과의 연동을 지원할 수 있는 기능을 포함하여야 한다. 지금까지의 리파지토리 시스템 연구는 기업 리파지토리 시스템 개념의 일부분만 수용하였으며 본 연구에서 제시하는 개념을 기반으로 한 연구는 미비하였다.

이하의 논문의 구성은 다음과 같다. 2절에서는, 리파지토리 시스템에 대한 분류기준(Taxonomy)를 제시하고, 이를 통해 기업 리파지토리 시스템의 위상을 정립한다. 3절에서는, 기업 리파지토리 시스템의 구현을 위한 표준적 지침을 제안하는 맥락에서, 기업 리파지토리 시스템의 아키텍처(Architecture)를 제시하고, 4절에서는, 아키텍처에 포함된 구성 요소들의 수행 기능들을 구체적으로 제시한다. 5절에서는, 전사적 자원관리(Enterprise Resource Planning : ERP) 시스템을 위한 리파지토리 시스템 구축 사례 분석을 통해 제시된 구성 요소들이 리파지토리 시스템에 대한 보완 및 비교 척도로 활용될 수 있음을 보였다.

2. 리파지토리 시스템 분류기준

리파지토리 시스템은 관리대상으로 삼고 있는 정보자산의 유형, 기능 및 구축 환경을 비롯한 다양한 관점에서 분류될 수 있다. 본 연구는 (그림 1)에서 볼 수 있듯이, 정보자산 영역(Resource Domain), CASE 도구 및 관련 정보시스템에 대한 의존성(Dependency), 그리고 분산정도(Distribution)의 세 가지 기준(Dimension)에서, 리파지토리 시스템을 분류한다. 각 기준을 상세히 설명하면 다음과 같다.



(그림 1) 리파지토리 시스템의 분류기준

과거 리파지토리 시스템의 구현이 기업의 관심 대상이 되는 정보자산을 중심으로 이루어져 왔기 때문에, 각각의 리파지토리 시스템에서 다루는 정보자산의 유형이 상이하다. 이러한 주요 정보자산에 대한 리파지토리 연구로는 기업의 정보 모델[Lee & Jung, 1999 ; Jorgenson, 1996], 문서[Sutton, 1996], 멀티미디어 데이터[Kashyap et al., 1995 ; Shkar et al., 1995], 객체, 클래스 및 관계 등의 객체 지향 프레임워크 요소[Shin et al., 1998], 또는 프로그램 및 데이터와 같은 시스템 자원[Bernstein et al., 1994, 1997 ; Lefkovits, 1999]에 관한 것이 있다. 최근, 주목 받고 있는 지식관리 연구에서도, 리파지토리에 대한 연구

는 중요한 비중을 차지하고 있다[Breslin & McGann, 1998 ; Jones, 1995]. 예를 들면, 프로젝트 관련 지식을 관리하기 위한 프로젝트 추적 데이터(Project-specific Trace Data) 리파지토리 [Domges & Pohl, 1998], 또는 리스크 관리를 위한 도메인 모형과 추론 방법을 근간으로 하는 리스크 관리를 위한 지식 리파지토리[Benaroch, 1997]에 관한 연구가 있다. 그러나, 동질적인(Homogeneous) 정보자산을 대상으로 별도의 리파지토리 시스템을 구축하기보다는 이들의 연관 관계에 대한 메타데이터를 기반으로 하여 이질적인 정보자산들을 통합적으로 관리하는 방향으로 발전시킴으로써, 정보자산 관리 및 활용에 있어 시너지 효과를 창출할 수 있다.

리파지토리 시스템이 추구하는 주요 조건 중 하나는, CASE 도구 및 관련 정보시스템 간의 독립성이다. 독립성은 리파지토리 시스템과 여타 도구 및 정보시스템 간의 메타데이터에 대한 호환체계를 기반으로 확보될 수 있다. 이러한 구조를 바탕으로, 리파지토리 시스템은 각기 다른 CASE도구 또는 정보시스템으로부터 생성된 다양한 형태(Format)의 정보자산들을 통합적으로 관리할 수 있는 유연한 구조를 가질 수 있는 것이다. 또한, 보다 발전적인 개념과 분석 기능을 보유한 CASE 도구를 도입하거나 새로운 정보시스템의 기능을 확장하는 경우, 이것들로부터 제공되는 메타데이터를 수용할 수 있는 확장 가능한 구조를 가져야 한다. 이러한 확장성(Extensibility)을 통해, 특정 개발 환경에서 요구되는 메타데이터에 종속되지 않는 독립성(Independence)을 확보할 수 있는데, 이러한 기능은 리파지토리 시스템의 필수적인 기능으로 강조되고 있다[Bernstein et al., 1997 ; MacGauhey, 1993 ; Simon, 1995]. 그러나, 독립성 보장은 기존의 리파지토리 시스템들의 주된 약점으로 지적되는 부분이다. 독립성 및 확장성을 지원하기 위

한 측면에서, CDIF(CASE Data Interchange Format)[Ashrafi & Kuilboer, 1995], PCTE(Portable Common Tools Environment)[ECMA, 1990 ; Hefernan, 1992], ATIS(A Tool Integration Standard)[Ashrafi & Kuilboer, 1995 ; Tannenbaum, 1994]와 IRDS (Information Resource Dictionary Systems)[ISO, 1990]와 같은 리파지토리 구조에 대한 표준안들이 제시되었다. 이와 같은 표준안은 이회석 외[1999]의 연구에서 구체적으로 비교되었다.

리파지토리 시스템의 적용 범위 관점에서는 분산정도를 기준으로 구분해 볼 수 있다. 전사적인 차원에서, 정보자산 관리를 위한 인프라가 되기 위해서는 분산형으로 구축되는 것이 바람직하다. CALS(Commerce At Light Speed) 리파지토리의 경우에는 분산환경을 기반으로 구축되는 전형적인 예이다. 이 리파지토리는 CALS의 최종 목표라 할 수 있는 통합 데이터베이스 환경(Integrated Database Environment) 구축을 위한 필수적인 부분으로서, 기업간 정보자산 공유를 지원하는 미들웨어 기능의 수행을 목적으로 한다[Kidwell & Richman, 1994 ; Kidwell et al., 1994].

이와 같은 세 가지 관점에서 볼 때, 리파지토리 시스템은 다양한 정보자산들을 통합적으로 관리할 수 있고, CASE 도구나 여타 정보시스템에 대해 독립성을 가지며, 분산 환경에서 운영 가능하도록 구축되는 것이 이상적이다. 본 연구에서는 이러한 발전성에 부합하는 리파지토리 시스템의 개념을 기업 리파지토리 시스템(Enterprise Repository System : ERS)이라 명명하고자 한다.

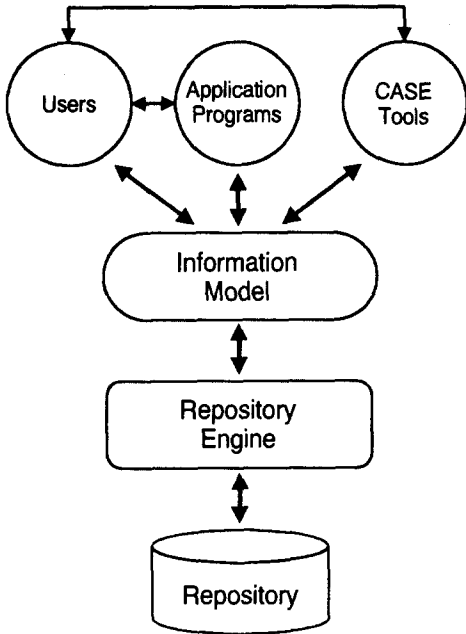
기존의 리파지토리 시스템들이 보다 효과적인 정보 인프라가 되기 위해서는, 궁극적으로 기업 리파지토리 형태로 발전해 나가는 것이 바람직하다. 예를 들어, 전사적 업무에 대하여 연

관된 모든 정보 자산들의 효율적 관리를 목적으로 하는 ERP 개념에 입각하여 볼 때, ERP 리파지토리 시스템은, 기업의 전체 업무 영역을 대상으로 응용시스템, 데이터베이스 등과 같은 시스템 자산을 비롯하여, 조직 모형, 데이터 모형 및 워크플로우(Workflow) 모형과 같은 다양한 기업 모형들을 통합적으로 관리하며, 업무간 상호 연동과정까지도 지원을 할 수 있도록 구축되어야 한다[Bancroft et al., 1997 ; SAP, 1997]. 이와 같은 맥락에서, 본 연구에서 개발 사례로 소개하고자 하는 ERP 리파지토리 시스템은 기업 리파지토리 시스템을 지향하여 구축하고 있다. 즉, 기업 리파지토리는 기업의 정보 자산 및 시스템 운영에 있어 중요한 영향을 미칠 것으로 기대된다.

3. 기업 리파지토리 시스템 아키텍처

리파지토리에 대한 정의는 다양한 관점에서 기술되어 왔는데[Moriarty, 1990 ; MacGaughey & Gibson, 1993 ; Bernstein & Dayal, 1994 ; Tannenbaum, 1994 ; Ashrafi & Kuilboer, 1995 ; Simon, 1995], 리파지토리 시스템과 혼용되어 사용되기도 하였다[Ashrafi & Kuilboer ; 1995 ; Lee & Joung, 1999]. 그러나, 본 연구는 리파지토리와 리파지토리 시스템에 관한 개념을 구분하여 사용한다. 왜냐하면, 리파지토리 시스템에 요구되는 기능들도 중요한 이슈로 보기 때문이다. 이러한 관점에서, 리파지토리 시스템은 (i) 메타데이터가 저장되는, 물리적 저장소인 리파지토리(Repository), (ii) 리파지토리에 저장되어 있는 메타데이터를 활용하고 관리하는데 필요한 다양한 메커니즘들을 제공하는 리파지토리 엔진(Repository Engine), 그리고 (iii) 인터페이스 역할을 하는 정보 모형(Information Model)으로 구분할 수 있다[Bernstein et al., 1997]. 이러한 요소들로 구성된

아키텍처는 (그림 2)와 같이 도식화 할 수 있다. 본 아키텍처는 기존의 일반적인 리파지토리 시스템과 기업 리파지토리 시스템에 있어 공통적이지만, 구체적인 기능면에서는 많은 차이가 있다.



(그림 2) 기업 리파지토리 시스템 아키텍처

기업 리파지토리 시스템의 리파지토리는 기업 내에 존재하는 다양한 CASE 도구 및 정보 시스템들간의 메타데이터 호환을 위한 미들웨어 구실을 할 수 있도록 독립성과 확장성을 보장할 수 있는 구조를 가져야 한다. 과거의 리파지토리에 대한 정의는 다양하게 이루어져 왔지만, 본 연구에서는, 기업 리파지토리를 “정보자산에 대한 개발 및 관리, 그리고 재사용을 위해 필요한 메타데이터를 포함하는 데이터베이스”로 정의하고자 한다. 기업 리파지토리가 분산 데이터베이스로 구축되어도, 정보자산에 대한 통합적인 관리를 위해, 논리적으로 단일한 통제 관점에 의해 관리되고 공유되어야 한다.

기업 리파지토리 시스템의 정보 모형의 역할

은 크게 두 가지 관점에서는 살펴볼 수 있다. 하나는, 다른 CASE 도구 및 정보시스템의 응용프로그램(Application Programs)과의 연결 역할이고, 다른 하나는 리파지토리 시스템 자체를 통제하고 관리하는 역할이다. 이는 각기 리파지토리 시스템 응용 그룹(Application Group)과 관리 그룹(Management Group)에 대응되는 개념이다. 기업 리파지토리 시스템의 경우, 리파지토리의 구조에 대한 변경 및 보안 등에 관련된 기능들을 전담하는 관리자가 정보인프라로써의 유연한 기반 역할을 할 수 있도록 관리하는 것이 중요하다. 이러한 역할을 담당하는 관리 그룹은 설치(Setup), 성능(Performance), 보안 및 접근통제(Security/Access Control) 및 운영 안내(Operational Guideline) 등에 관련된 업무를 통해, 리파지토리 시스템 자체를 관리하는 역할을 수행한다. 사용 그룹은 메타데이터 생성과정을 수행하고, 응용업무 수행을 위해 리파지토리 시스템을 사용한다. 메타데이터 생성과정은 새로운 메타데이터를 등록하는 과정을 포함하는데, 기업의 시스템 분석가나 개발자들이 이에 속한다. 이들은 주로 시스템 개발 도구를 이용하며, 다른 도구의 데이터를 리파지토리 시스템을 통해 요청하기도 한다. 리파지토리 시스템이 각종 응용도구간의 데이터 공유를 위한 중계 기능을 하고 정보자산의 통합적인 활용 및 관리를 담당하는 미들웨어 역할을 한다.

리파지토리 엔진은 리파지토리에 대한 통제 메커니즘을 제공할 뿐만 아니라 정보 모형을 통해 사용자가 수행할 수 있는 기능들에 대한 논리를 구현한 것이다. 따라서, 기본적으로 메타데이터의 생명주기(Life-cycle)의 전과정을 지원할 수 있어야 한다. Bernstein & Dayal[1994]는 체크인/체크아웃(Checkout/Checkin), 버전 통제(Version Control), 구성 통제(Configuration Control), 알림기능(Notification), 내용 관리

(Context Management), 워크플로우 관리(Work-flow Management)와 같은 기능을 수행하는 메커니즘이 필요하다고 지적하고 있다. 한편, Bordoloi et al.[1998]의 연구에서는 주로 데이터에 대한 보안, 무결성, 그리고 응용시스템의 인터페이스와 같은 시스템 자원에 대한 관리에 중점을 두고 있다.

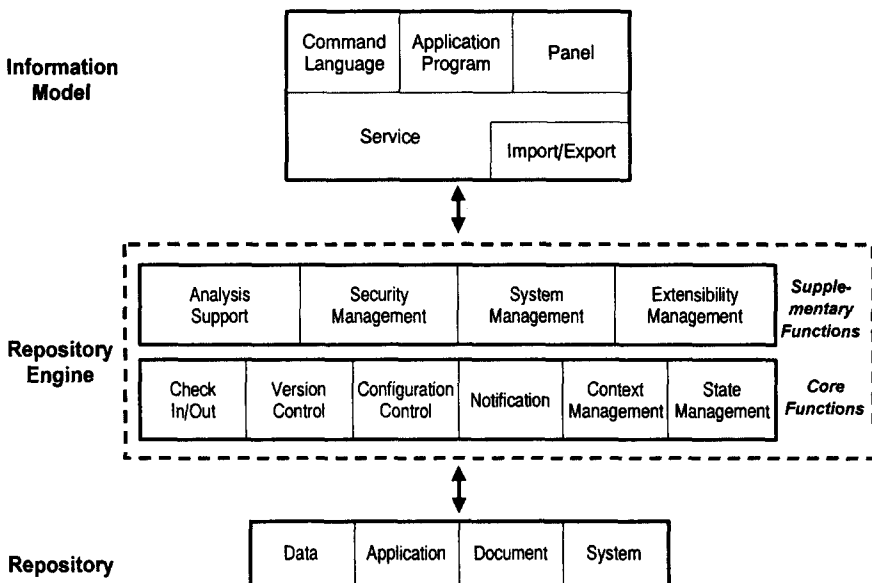
기업 리파지토리 시스템의 장점은 (i) 특정 개발 환경에 독립적이고, (ii) 이질적인 개발 도구들을 상호 유기적으로 연결하여 통합적인 개발 접근이 가능하며, (iii) 장기간 사용 시 기업 정보자산 축적이 가능하다는 것이다. 이에 따라 기업 리파지토리 시스템은 기업 지식 자산의 저장 및 재사용성을 제고하고 향후 신 정보시스템 구축을 효과적으로 지원할 수 있다.

4. 기업 리파지토리 시스템 아키텍처 세부 구성

기존의 연구는 리파지토리 시스템의 구성 요

소 중 리파지토리 엔진 또는 정보 모형 중심으로, 구체적인 기능 또는 관리 요소들을 제시하고 있다. Bordoloi et al.[1998]의 연구에서는, 데이터와 응용프로그램 모듈 중심의 시스템 관점에서 접근하고 있으며 리파지토리 엔진의 기능과 정보 모형에 해당하는 인터페이스 기능에 대한 지원기능에 대해 구체적인 내용들을 제시하고 있다. 한편, Berstein & Dayal[1994]의 연구는 리파지토리 엔진에 초점을 맞추어, 이것의 주요 기능들을 제시하고 있는데, Bordoloi et al.[1998]의 연구에 비해, 시스템 관리 기능은 미비하다.

리파지토리 시스템의 역할은 다양한 관점에서 고려될 수 있으나, 기업 리파지토리 시스템은 기업의 다양한 활동을 지원할 수 있어야 하므로, 시스템 개발 및 관리의 지원, 응용 업무지원에 대한 기능 및 관리 대상들이 포괄적으로 고려되어야 한다. 이에 따라, 기업 리파지토리 시스템의 구성요소별 내역을 도식화하면 (그림 3)과 같다.



(그림 3) 기업 리파지토리 시스템의 구성요소별 세부 내역

정보모형(Information Model)에 나타난 내역은 기능에 따른 유형을 표현한 것인데, 이는 IRDS 표준에서 제시된 인터페이스의 주요 요소를 기반으로 재정리되었다[ISO, 1990]. 그리고, 리파지토리 엔진(Repository Engine)은 핵심 기능(Core Functions)과 보조적 기능(Supplementary Functions)으로 구분될 수 있는데, 본 논문에서 제시된 핵심 기능은 Bernstein & Dayal[1994]의 연구에서 제시된 것들을 기준으로 하였다. 리파지토리(Repository)에서는, 기업의 정보자산을 구성하는 객체의 유형을 네 가지로 구분하였다. 이러한 유형화는 객체들에 대한 메타데이터를 결정하는데, 유용한 단서가 될 것이다. 객체의 유형별 특징을 차별적으로 관리함으로써, 효과적인 정보자산에 대한 관리 및 재활용의 성과를 얻을 수 있다. 기업 리파지토리 시스템의 기능과 관리 요소에 대한 세부적인 항목은 <표 1>과 같이 요약될 수 있다. (그림 3)에서는 정보모형, 리파지토리 엔진의 주기능만 표시되었는데, <표 1>은 이들을 세분화하여 설명하고 있다.

기업 리파지토리 시스템의 구체적인 내역은 기존 연구를 기반으로 제시되었으나, 각 요소들에 대한 전체적인 내역을 통합적으로 재정립하였다는데 의의가 있다. 또한, 이와 같은 내역은 향후 기업 리파지토리 구축에 있어 가이드라인으로 유용하게 활용될 수 있을 뿐만 아니라, 기존의 리파지토리 시스템의 위상을 평가하는 지표(Checklist)로도 활용될 수 있다. 평가 결과, 리파지토리 시스템이 제공하지 못하는 기능 및 관리요소 중심의 개선 전략이 파악될 수 있다. 이를 위하여 다음 절에서 구체적인 사례분석을 하였다.

5. 구현 사례 : ERP 리파지토리 시스템

사례로 제시할 ERP 리파지토리 시스템은, 1997

년 말부터 시작된 표준정보시스템(ERP) 구축 프로젝트로서 호연연구원, 국내 10여 개 시스템 개발업체와 한국과학기술원에서 수행 중이다. 약 50여명의 개발, 설계 및 프로젝트 관리 인원이 참여하고 있다[이희석, 1999]. 한국과학기술원은 특히 리파지토리의 개념적 설계 및 구축을 수행하고 있다. 1차년도에는 리파지토리의 기본적 구조가 제시되었으며 현재는 기본 구조의 개선 방안을 제시 중이다.

(그림 1)에서 제안된 분류기준에 따르면, 현재 ERP 리파지토리 시스템은 정보자산 영역에서만 기업 리파지토리 시스템으로 향하고 있다. 비독립성과 비분산형은 이 프로젝트의 본질적인 한계에서 비롯된 것이지만 프로젝트가 계속 진행됨에 따라 이 부분도 보완될 예정이다. 시스템 구성도는 (그림 2)와 동일한 형태를 가졌다. ERP 리파지토리 부분은 데이터베이스 편집 및 어플리케이션 편집(1차년도 과제), 모니터링 통제, 보안 및 권한, 워크플로우 및 조직모델(2차년도 과제) 부분으로 이루어져 있다. 데이터베이스 편집은 <표 1>에서 Data에 해당되며, 어플리케이션 편집과 워크플로우 및 조직모델은 Application에 해당되며, 모니터링 통제, 보안 및 권한은 System에 해당된다. 이 ERP 리파지토리 스키마 부분은 <표 1>의 Document 부분이 고려되어 있지 않다. ERP 리파지토리 엔진은 Context Management, Business Analysis Support, Treatment Control, Access Control, Resource Control, System Configuration, Performance Control의 기능을 가지고 있다.

정보 모형 부분은 Service, Command Language, Application Program의 세가지 기능을 가지고 있으며, 그 외 부분은 고려되어 있지 않다. <표 2>는 현재의 구축 중인 ERP 리파지토리 시스템을 <표 1>의 체크리스트 항목에 따라 구현 여부를 표시한 것이다. 본 현황에 따라 궁

〈표 1〉 기업리파지토리 시스템의 구성요소별 기능과 관리요소

시스템 구성 요소	주기능 및 관리요소		설 명	
		세부기능 및 세부관리요소		
Information Model	Service		리파지토리 정보에 대한 접근과 조작에 필요한 기본적인 서비스를 제공하며, Panel과 Command Language, Application Program에 의해서 사용되는 인터페이스.	
	Panel		Panel과 스크린으로 구성되어 있으며, 사용자가 GUI 환경을 통하여 리파지토리 정보에 대한 접근을 제공하는 인터페이스.	
	Command Language		Statement Expression 형태의 명령 체계를 통하여 사용자의 리파지토리 정보에 대한 접근을 제공하는 인터페이스.	
	Application Program		프로그램 언어 내부에서의 Command Language 사용을 통하여, Application이 리파지토리 정보에 대한 접근을 제공하는 인터페이스.	
	Import/Export		리파지토리 정보를 이질적인 도구간에 상호교환을 제공하는 인터페이스.	
Repository Engine	CORE	Check In/Out	CheckOut은 객체에 대한 잠금(Lock)을 설정하는 것이며 CheckIn은 잠금을 해제하는 것임.	
		Version Control	하나의 객체가 조금씩 수정을 거쳐 변화해 가는데, 최종적인 객체의 상태뿐 아니라 이전의 객체 상태도 접근할 수 있도록 하는 기능.	
		Configuration Control	Configuration 제어는 복합객체(Composite Object)의 버전을 관리하는 기능이다. 복합객체는 상호 관련된 객체들을 그 구성원으로 하는 객체로 복합객체의 구성객체들은 모두 함께 실체화되며, 또한 각각의 구성객체들도 다시 복합객체가 될 수 있음.	
		Notification	버전의 변경을 감시하고 변경이 일어났을 경우에 이와 관련된 다른 객체들에게 변경 사실을 알려 줄 수 있는 기능.	
		Context Management	리파지토리 객체에 대한 뷰(View)를 Context라고 한다. 사용자 프로파일, 특별한 규칙, 제약조건 등도 포함됨.	
		State Management	요구사항, 분석 및 설계, 구축, 테스트 등의 생명주기 단계에 관련된 객체의 상태(State)를 추적하는 기능.	
	SUPPLEMENTARY	ANALYSIS SUPPORT	Analysis Support	비즈니스 전략 및 분석 수행에 필요한 표준적 프레임과 기능들을 제공하고, 이에 대한 시스템 기능에 대한 요구사항들을 정의하고 분석할 수 있는 기능.
			Business Analysis Support	업무에 대한 전략 계획 및 워크플로우와 같은 업무분석 모형을 생성 및 관리, 시뮬레이션 등에 필요한 기능을 제공함. 업무 분야별 수행 단계들의 기능 및 데이터에 대한 상호관련성 분석 기능을 제공해야 함.
		SECURITY	System Requirement Support	업무 분석을 기반으로 시스템이 지원해야 할 기능에 대한 요구사항들을 정의하고, 분석할 수 있는 기능들을 제공함.
			Security Management	리파지토리 시스템 및 정보자산에 대한 보안 기능을 제공함.
			Treatment Control	데이터 처리에 대한 권한을 통제함.
		SYSTEM	Access Control	데이터에 대한 접근을 통제함.
			System Management	하드웨어 및 소프트웨어와 관련된 시스템 자원에 대한 관리.
Resource Control			시스템 자원에 대한 상태를 파악할 수 있는 정보를 제공하고, 이들에 대한 조정과 관련된 기능을 제공함.	
System Configuration			시스템 자원들간의 관계를 관리함.	
EXTENSIBILITY		Performance Control	시스템 자원들의 성능에 대한 정보를 제공하고, 이들의 기능이 최적화될 수 있도록 지원하고 관련된 통제 기능들을 제공함.	
	Extensibility Management	새로운 메타데이터를 등록할 수 있도록, 메타데이터 스키마에 관련된 정보를 제공하고, 이것의 생성 및 갱신을 지원하는데 필요한 기능.		
Repository OBJECT	Data	텍스트 기반의 Transactional Data, Analytical Data 등에 대한 메타데이터 저장.		
	Application	워크플로우 모형과 같이 애플리케이션의 설계를 위한 모형 또는 애플리케이션 수행을 위한 Code 그리고 표현양식에 관련된 Form 과 같은 애플리케이션 관련 요소들에 대한 메타데이터 저장.		
	Document	멀티미디어 데이터와 이를 포함하는 전자문서들에 대한 메타데이터 저장.		
	System	하드웨어 및 소프트웨어 관련 요소들에 대한 메타데이터 저장.		

<표 2> 현재 ERP 리파지토리 시스템 현황

시스템 구성 요소	주기능 및 관리요소	세부기능 및 세부관리요소	구현 ERP 리파지토리 시스템 현황	
Information Model	Service		0	
	Panel		X	
	Command Language		0	
	Application Program		0	
	Import/Export		X	
Repository Engine	Check In/Out		X	
	Version Control		X	
	Configuration Control		X	
	Notification		X	
	Context Management		0	
	State Management		X	
	Analysis Support	Business Analysis Support		0
		System Requirement Support		X
	Security Management	Treatment Control		0
		Access Control		0
	System Management	Resource Control		0
		System Configuration		0
		Performance Control		0
Extensibility Management		X		
Repository	Data		0	
	Application		0	
	Document		X	
	System		0	

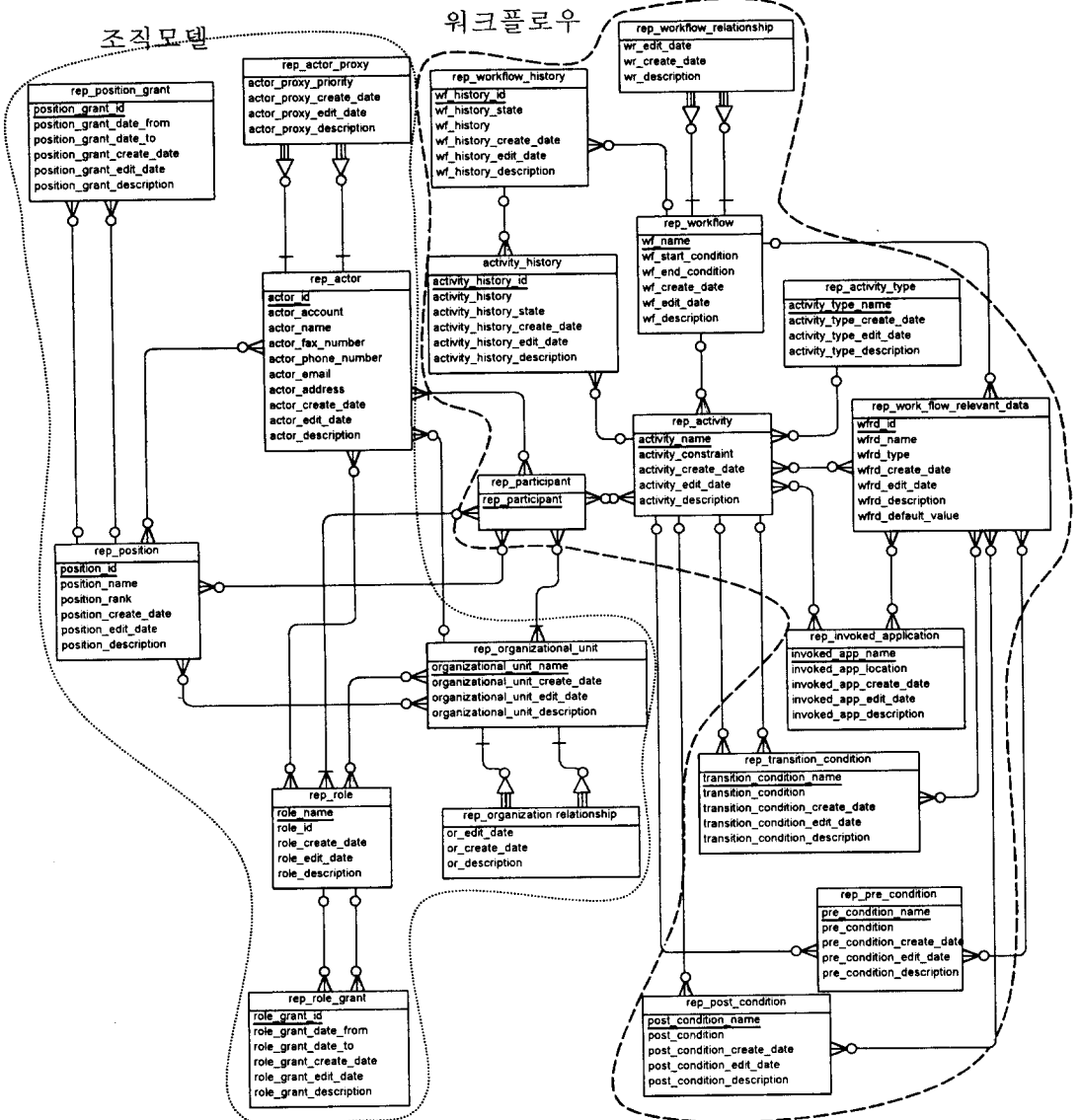
극적인 기업 리파지토리 시스템으로 발전하기 위한 방향이 제시될 수 있다.

본 연구에서 분석하고자 하는 ERP 리파지토리 시스템 사례는 IRDS 표준안을 기반으로 구현되었다. IRDS는 4계층 구조를 기반으로 한다. 한 계층의 정보는 다음 하위계층에 저장된 정보를 설명하고 제어한다. IRDS 표준안은 계층 개념을 통해서 두 계층을 하나의 쌍으로 구성하며 상위층은 정의(Definition), 하위층은 인스턴스(Instance)로 이루어진다. 즉, 다음 하위계층의 개체 타입을 그 상위계층에서 정의하게 되고, 하위계층은 상위계층에서 정의된 개체 타입에 관한 실제 값을 갖는다[이희석 외, 1999]. IRDS 표준안의 IRD 계층(제3계층)은 실세계 정보를 관리하기 위한 계층이다. 구현된 시스템의 제3계층은 인사, 설비, 품질, 생산, 판매, 재무, 회계, 프로젝트 관리와 산업 솔루션 등을 지원하

기 위한 개체로 500여 개가 산출되었다. 이 IRD 계층을 추상화한 계층이 IRD스키마 계층(제2계층)이다. 제2계층은 어플리케이션, 데이터베이스, 모델링 부문, 워크플로우 및 조직모델 부문으로 분류되며, 총 80여 개의 개체가 도출되었다. 제2계층을 관리하는 IRD스키마 정의계층이 제1계층인데, 2계층의 내용을 추상화한 개체, 관계, 속성의 3종류의 개체로 구성된다.

본 절에서는 설명의 편의상 IRDS의 중요 계층인 제2계층의 워크플로우 및 조직모델을 중심으로 내용을 요약하겠다. 적용 현실성을 고려하여 IRDS 기반 메타스키마 모델링에 개체 관계도(Entity Relationship Diagram : ERD)를 이용하였다. ERP 리파지토리 시스템에 이용된 리파지토리 중 Application 안에 포함되어 있는 워크플로우 및 조직모델 스키마는 (그림 4)와 같다. 워크플로우 및 조직모델 스키마를 좀 더 설명하면 다음과 같다. 다른 부문에 대한 상세한 설명은 [이희석 외, 1999]에서 참조할 수 있다.

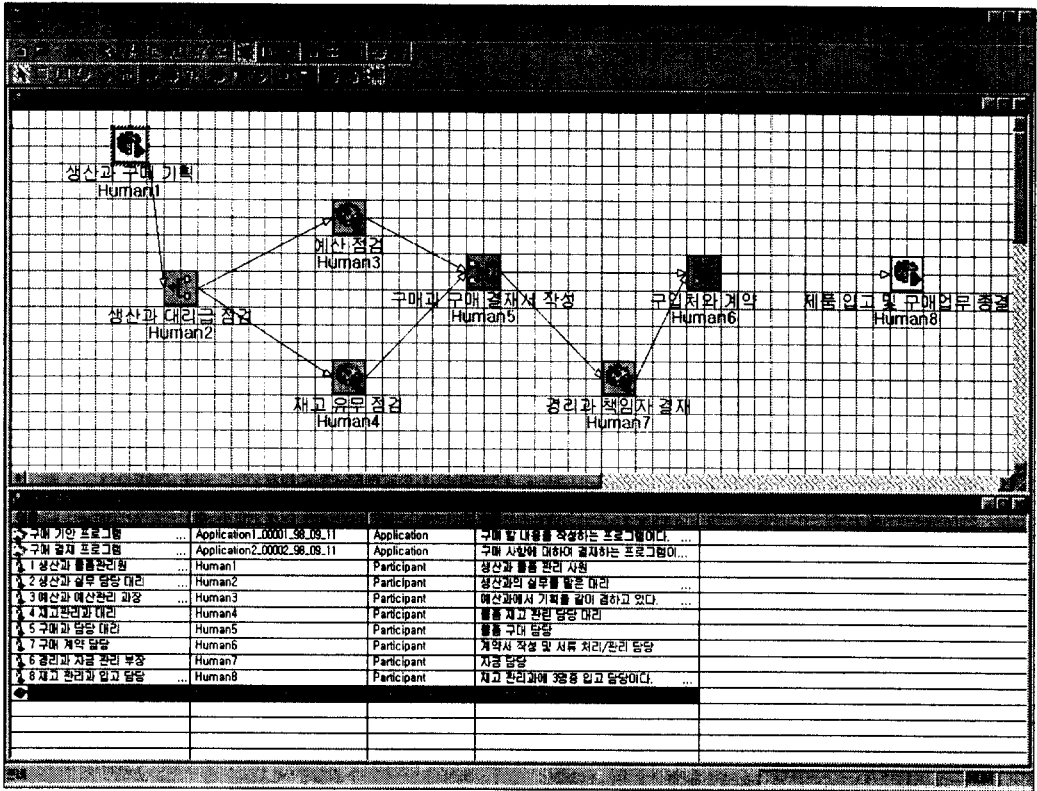
워크플로우 및 조직모델 부문에 대한 리파지토리 스키마는 조직모델 부문과 워크플로우 부문의 2부문에 대별된다. 조직 모델 부문은 기업의 조직구조를 표현하는 부문으로 워크플로우 작성 시에 필요한 조직 정보를 제공한다. 조직 단위(rep_organizational_unit)는 기업 내부에 존재하는 부서 또는 팀과 관련된 개체이며, 조직 단위 관계(rep_organization_relationship)는 조직 단위 간의 관계를 표시하는 개체이다. 액터(rep_actor)는 업무를 처리하는 사람에 대한 개체이며, 조직 단위의 업무에 대한 역할(rep_role) 개체 및 직위(rep_position) 개체와 관계가 있다. 액터는 자신의 업무를 대행할 수 있는 대행자(rep_actor_proxy)를 가질 수 있으며, 역할은 다른 역할에 역할 부과(rep_role_grant)를 할 수 있으며, 직위도 다른 직위에 대한 직위 부과(rep_position_grant)를 가질 수 있다.



(그림 4) 리파지토리 제2계층 워크플로우 및 조직모델 스키마

워크플로우 부분은 워크플로우에 대한 정의와 이에 필요한 정보들을 제공하는 부분이다. 워크플로우(rep_workflow) 개체는 워크플로우에 대한 정의를 나타내며, 워크플로우 간의 상호 관계를 표현하기 위한 워크플로우 관계(rep_workflow_relationship) 개체가 있으며, 워크플로우의

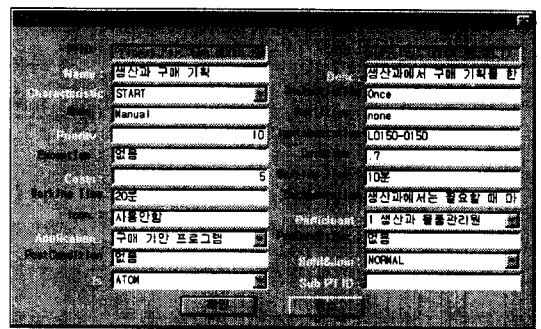
수행된 정보 개체로 워크플로우 이력(rep_workflow_history)이 있다. 액티비티(rep_activity)는 워크플로우 수행에 필요한 단위 업무에 대한 개체이며, 액티비티 수행 정보들을 다루는 액티비티 이력(rep_activity_history)이 있다. 액티비티는 액티비티 유형(rep_activity_type)에 따라 분



(그림 5) 워크플로우 편집기 화면

류되며, 액티비티 수행을 위해 필요한 전위조건 (rep_pre_condition), 수행 후 완결 여부를 확인하는 후위조건(rep_post_condition), 다른 액티비티로의 전이를 위한 전이조건(rep_transition_condition)과 액티비티 수행을 위해 호출되는 어플리케이션(rep_invoked_application)을 가진다. 워크플로우에서의 각 개체간의 정보의 전달은 워크플로우 연관 데이터(rep_workflow_relevant_data)로 표현된다.

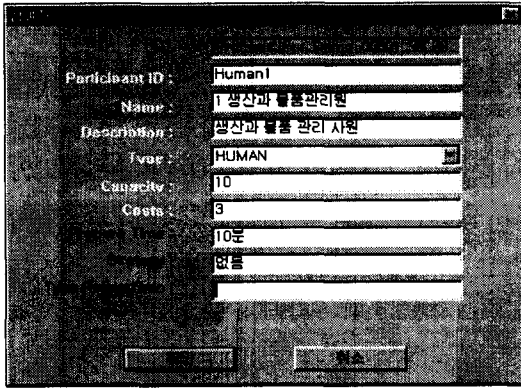
(그림 5)는 이러한 워크플로우 스키마를 기반으로 개발된 워크플로우 작성기의 화면을 나타낸 것이며, 워크플로우 작성기에서는 업무에 요구되는 개별 액티비티의 흐름과 참여자, 응용프로그램, 관련 데이터 등을 설계하게 된다. (그림 6)은 워크플로우에 사용되는 개별 액티비티를 작



(그림 6) 액티비티 편집 화면

성하는 화면을 나타낸 것이며, (그림 7)은 액티비티를 수행하는 참여자를 작성하는 화면이다.

워크플로우 편집기는 기업 리파지토리 시스템 구성((그림 2))의 관점에서 볼 때, Application Programs에 해당된다고 할 수 있다. 워크



(그림 7) 참여자 편집 화면

플로우 편집기는 리파지시스템에서 제공하는 정보모형과 리파지토리 엔진을 이용하여 앞에서 설계된 리파지토리에 저장된 정보를 활용하게 된다.

6. 결 론

기업 정보자산의 일부 범위에 적용되던 CASE 도구들간의 이질성을 극복하기 위한 대응 방안으로 리파지토리 시스템의 구축이 주목 받아 왔다. 리파지토리 시스템이 기업내 존재하는 CASE 도구 및 정보시스템들의 이질성을 극복하는 효과는 얻었으나, 이들이 다루는 정보자산의 한정적인 범위로 인해, 기업의 다양한 정보자산들간의 시너지 효과를 얻는데 한계가 있었다. 결국, 이러한 리파지토리 시스템들은 기업의 전사적인 차원에서 정보자산에 대한 통합적 관리 및 재활용의 기능을 수행할 수 있는 기업 리파지토리 시스템으로 발전하는 것이 바람직한데, 아직 이에 대한 이해가 미흡한 실정이다.

따라서, 본 논문에서는, 우선 리파지토리 시스템에 대한 분류기준을 제시하고, 이를 기반으로 기업 리파지토리의 개념을 제시하였다. 제시된 분류기준을 기반으로 기존의 다양한 리파지토리 시스템의 위상을 가늠해 볼 수 있다. 또한,

기업 리파지토리 시스템의 본원적인(Generic) 구조 및 요구 기능들을 재정립하였다. 이는 향후 리파지토리 시스템 개발에 있어 특정 벤더에 종속적이지 않는 표준적 지침으로써 활용될 수 있을 뿐만 아니라, 여타 리파지토리 시스템들에 대한 구체적인 역량을 측정 및 비교할 수 있는 척도로 활용할 수 있다는데 의의가 있다. 본 연구에서 제시된 기업 리파지토리 시스템은 리파지토리 시스템들의 궁극적인 발전 형태에 해당하는 개념으로써, 사례분석을 통해 소개한 ERP 리파지토리 시스템은 이에 근접한 형태라 할 수 있다. 대부분의 기존 리파지토리 시스템들이 그러하듯이, 소개된 리파지토리 시스템 또한 독립성을 충분히 확보하지 못하고 있으나, 고도의 기술수준이 필요한 만큼 향후 이 부분에 대한 연구가 집중될 필요가 있다.

참 고 문 헌

- [1] 이희석, 이재, 이충석, 조창래, 손주찬, 백종명, "전사적 자원관리 시스템을 위한 기업 리파지토리 구축", *경영정보학연구*, 9(1), 1999, pp.59-75.
- [2] N. Ashrafi and J.P. Kuilboer, "The Information Repository : A Tool for Metadata Management," *Journal of Database Management*, 6(2), 1997, pp.3-10.
- [3] N.H. Bancroft, H. Seip, and A. Sprengel, *Implementing SAP R/3*, Second Ed., Manning Publications Co., 1997.
- [4] M. Benaroch, "Toward the Notion of a Knowledge Repository for Financial Risk Management," *IEEE Transaction on Knowledge and Data Engineering*, 9(1), Jan.-Feb. 1997, pp.161-167.
- [5] P.A. Bernstein and U. Dayal, "An Overview

- of Repository Technology," *Proceedings of the 20th VLDB Conference*, Santiago Chile, 1994, pp.705-713.
- [6] P.A. Bernstein, P. Sanders, B. Harry, D. Shutt, and J. Zander, "The Microsoft Repository," *Proceedings of the 23rd VLDB Conference*, Athens, Greece, 1997.
- [7] B. Bordoloi, S. Sircar, and B. Lakhanpal, "Desirable Characteristics of Information Resource Dictionary Systems," *Journal of Database Management*, 9(2), 1998, pp.3-15.
- [8] J. Breslin and J. McGann, *The Business Knowledge Repository*, Quorum Books, 1998.
- [9] R. Domges and K. Pohl, "Adapting Traceability Environments to Project-Specific Needs," *Communications of the ACM*, 41 (12), 1998, pp.54-62.
- [10] ECMA, *Portable Common Tool Environment (PCTE) Abstract Specification*, ECMA European Computer Association Standard ECMA-149, Dec. 1990.
- [11] H. Hefernan, "U.S. looks to Ecma," *Software Magazine*, 12(15), 1992, pp.33.
- [12] ISO, *Information Technology-Information Resource Dictionary System(IRDS) Framework*, ISO/IEC 10027, 1990.
- [13] H. Lee and J. Joung, "An Enterprise Model Repository : Architecture and System," *Journal of Database Management* (In press), 1999.
- [14] C. H. Jones, *Knowledge Based Systems Methods : A Practitioners' Guide*, Prentice Hall, 1995.
- [15] B.R. Jorgenson, "Model Repository Technology for Model Integration," *Proceedings of the 1st International Conference on Enterprise Integration Modeling*, MIT Press, 1995, pp.419-426.
- [16] V. Kashyap, and K. Shah, and A. Sheth, "Metadata for Building the Multimedia Patch Quilt," *Multimedia Database System: Issue and Research Directions*, S. Jajodia and V.S. Subrahmanian (Eds.), Springer-Verlag, 1995.
- [17] R.S. Kidwell, J.G. Richman, R.E. Burkewitz, and W.C. Gorham, "Final Repository Plan report for the OSD CALS IWSDB Project," ManTech International Co., Aug. 1994.
- [18] R.S. Kidwell and J.G. Richman, "Preliminary Integrated Weapon System Database (IWSDB) Implementation Strategy Paper," ManTech International Co., July 1994.
- [19] H.C. Lefkovits, *IBM's Repository Manager/MVS : Concepts, Facilities, and Capabilities*, QED Technical Publishing Group, 1991.
- [20] R.E. MacGauhey, and M. Gibson, "The Repository/Encyclopedia : Essential to Information Engineering and Fully integrated CASE," *Journal of System Management*, 44(3), 1993, pp.8-11/41-42.
- [21] J. Martin, *Information Engineering : A Trilogy*, Vol. 1-3, Englewood Cliffs, NJ, Prentice-Hall, 1989.
- [22] T. Moriarty, "Are You Ready for a Repository," *Database Programming & Design*, 3(3), 1990, pp.60-71.
- [23] T. Moriarty, "Tools for managing metadata," *Database Programming & Design*, 6(10), 1993, pp.69-70.
- [24] SAP, *R/3 System : SAP Business Objects*, Technology Marketing, SAP AG., 1997.

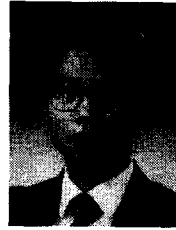
- [25] H. J. Shin, I. W. Choi, S. D. Kim, and S. Y. Rhew, "A Design of Object-Oriented Framework Repository," *Proceedings of the IEEE International Conference on Systems, Mans, and Cybernetics*, 1998, pp. 2686-2691.
- [26] A. R. Simon, *Strategic Database Technology : Management for the Year 2000*, Morgan Kaufmann Publishers, Inc. San Francisco, California, 1995.
- [27] M.J.D. Sutton, *Document Management for the Enterprise : Principles, Techniques, and Applications*, Wiley, 1996
- [28] L. Shklar, A. Sheth, V. Kashyap, and S. Thatte, "InfoHarness : A System for Search and Retrieval of Heterogeneous Information," *Proceedings of ACM SIGMOD*, Jun. 1995.
- [29] A. Tannenbaum, *Implementing a Corporate Repository : The Model meets Reality*, John Wiley & Sons, Inc. 1994.

■ 저자소개



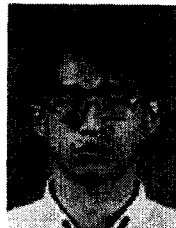
이 희 석

서울대 산업공학 학사, KAIST 산업공학 석사, 그리고 University of Arizona at Tucson에서 경영정보시스템을 전공하여 경영학박사학위를 취득하였으며, 1992년에서 1994년까지 University of Nebraska at Omaha에서 MIS를 강의하였다. 현재는 KAIST 테크노경영대학원 부교수로 재직하고 있다. 주요 관심분야는 인터넷 비즈니스, 지식경영, 정보전략, ERP, 데이터베이스이다.



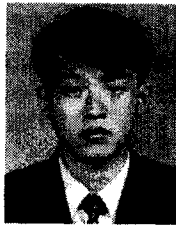
서 우 중

연세대 응용통계학과에서 학사, 석사 학위를 취득하였으며, 현재 KAIST 테크노경영대학원에서 박사과정을 이수 중이다. 주요 관심분야는 하이퍼미디어, e-business 모형, 지식관리시스템, 메타데이터 관리, 워크플로우 분석이다.



김 태 훈

고려대 통계학사 및 동 대학원 석사를 취득하였으며, 현재는 KAIST 테크노경영대학원 경영공학 박사과정을 이수 중이다. 주요 관심분야는 데이터 웨어하우스 구축, 분산 데이터베이스 설계, 메타데이터 관리, 데이터 마이닝 설계이다.



이 충 석

KAIST 경영과학과에서 학사, 동 대학 테크노경영대학원에서 석사 학위를 취득하였으며, 현재 동 대학원에서 박사과정을 이수 중이다. 주요 관심분야는 e-business 구축 방법론, 인터넷, 리파지토리이다.



손 명 호

서울대 산업공학과를 졸업하고 KAIST 테크노경영대학원에서 석사학위를 취득하였으며, 현재 동 대학원에서 박사과정을 이수 중이다. 주요 관심분야는 지식경영, 데이터 웨어하우스 구축, 분산객체 기술이다.



백 증 명

고려대 산업공학과에서 학사, 동 대학 전산학과에서 석사학위를 취득하였으며, 삼성전자와 시스템공학연구소 책임연구원으로 재직한 바 있다.

현재 한국전자통신연구원 정보통합연구팀장으로 재직하고 있으며, 주요 관심분야는 ERP, CALS, EC, ASP이다.



박 성 진

고려대 전산학과에서 학사, 석사, 박사 학위를 취득하였으며, 한국전자통신연구원 정보통합연구팀 선임연구원으로 재직한 바 있다. 현재 한

신대학교 정보처리학과 조교수로 재직하고 있으며, 주요 관심분야는 ERP 및 CALS, 분산 및 통합 데이터베이스, 분산객체 컴퓨팅이다.



손 주 찬

한국외대 서반어과에서 학사, 동 대학 경영정보대학원 경영정보학과에서 석사 학위를 취득하였으며, (주)에스콰이어 전산실과 시스템공학연구소 선

임연구원으로 근무한 바 있다. 현재 한국전자통신연구원 정보통합연구팀 선임연구원으로 재직하고 있으며, 주요 관심분야는 ERP, ASP, SCM, 분산 및 통합 데이터베이스이다.