

메타 데이터 기반의 기업용 애플리케이션 구축 방안에 관한 연구

장길상*

A Study on the Scheme of Implementing Meta-data Based Applications for Enterprises

Gil-Sang Jang*

요 약

일반적으로, 기업에 요구되는 정보시스템의 개발 사이클은 시스템 계획 및 선정, 시스템 분석, 시스템 설계, 시스템 구현 및 운영의 4단계로 이루어진다. 이런 절차에 따라 한번 개발되고 도입된 정보시스템들은 잦은 업무환경의 변화 및 그에 따른 빈번한 업무 프로세스의 변경 등으로 시스템의 개발 과정에서 뿐만 아니라 유지보수 단계에서도 애플리케이션의 추가적인 개발 및 수정 요구 등을 지원하기 위하여 많은 노력과 비용이 요구되고 있다. 특히, 개발과정 중에는 정확한 사용자 요구사항 파악의 어려움 및 잦은 요구사항 변경 등으로 인한 시스템 설계의 빈번한 변경과 그에 따른 프로그램 개발의 어려움으로 인하여 시스템 설계와 구현 사이에 항상 불일치가 발생하게 되고, 더 나아가서 시스템 개발의 표준화 정립이 어려운 실정이었다. 본 논문에서는 이러한 시스템 설계와 구현 사이의 불일치를 줄이고, 신속한 응용프로그램의 개발 및 기존의 한번 개발되어지면 폐기 때까지 사용하여야 하는 기존 개발방식의 한계를 극복하기 위하여, 메타 데이터 기반의 애플리케이션 개발 방안을 제시하고, 이를 위하여 메타 데이터를 체계적으로 관리 위한 리파지토리 시스템의 프레임워크를 제시한다. 본 논문에서 제시된 메타 데이터 기반의 애플리케이션 구축 방안은 시스템 분석 및 설계 단계에서 발생하는 산출물 정보를 메타 데이터화 하고, 이것을 기반으로 한 리파지토리 시스템을 구축 · 활용함으로써 애플리케이션의 개발 및 고객화를 용이하게 하는 개념이다. 본 논문에서 제시한 메타 데이터 기반의 애플리케이션 구축 방안의 적용가능성을 보여주기 위하여, 'H' 자동차부품업체의 ERP 시스템 구축에 적용되어졌다. 그 결과, 기존의 코딩 기반 개발 방식보다 제시된 메타 데이터 기반 개발방식이 개발 용이성, 개발 생산성, 유지보수 용이성, 프로그램 구성요소들의 재사용성 등 여러 측면에서 향상되었다.

Abstract

Generally, the phases of constructing information systems are consisted of systems planning and selection, system analysis, system design, and system implementation and operation. These systems require many efforts and costs for additional development or modification requirements due to a frequent changes of business environments and business processes. Especially, inconsistencies between system design and system implementation usually happen during

• 제1저자 : 장길상
• 투고일 : 2009. 09. 07, 심사일 : 2009. 09. 11, 게재확정일 : 2009. 09. 16.
* 울산대학교 경영정보학과 교수

development steps because of the difficulties of program developments due to difficulties of capturing exact user requirements and frequent changes of user requirements. This paper proposes a scheme of implementing meta-data based applications for enterprises in order to reduce inconsistencies between system design and system implementation and to overcome limits of the existing coding-based development methods of applications which must use until disuse if they are developed once. Also, this paper presents a framework of repository system to systematically manage and utilize meta-data. The core concept of the proposed scheme makes outputs generated in the phases of system analysis and design into meta-data and is to easily develop and customize application programs using meta-data repository. Also, to show the applicability of the proposed scheme, it is applied to implement ERP system of 'H' automotive part manufacturer. As a result, the proposed scheme can gain improvements such as easiness and productivity of program development, easiness of maintenance, reusability of program components, etc.

- ▶ Keyword : 메타 데이터(meta-data), 애플리케이션 개발 방안(application development scheme), 시스템 분석 및 설계(system analysis and design), 리파지토리 시스템(repository system)

I. 서론

최근 기업의 비즈니스 환경은 급속도로 변화하고 있으며, 또한 이를 둘러싼 정보기술 환경도 급속도로 발전하고 있다. 따라서 기업들은 이러한 정보기술을 기업 경영에 효과적으로 활용 할 수 있느냐의 여부에 따라 향후 기업 경쟁력을 좌우한다는 인식으로 적극적으로 ERP 같은 기업용 애플리케이션을 개발 및 도입하고 있는 추세이다. 그러나 기업의 비즈니스를 효율적으로 지원하기 위한 애플리케이션의 개발환경은 복잡하고, 다양하며, 글로벌시대에 대응하기 위해 빠르게 진화하고 있다. 이러한 환경에 능동적으로 대처하기 위해 소프트웨어 부품의 재사용을 기반으로 하는 CBD(Component Based Development)나 객체지향 설계에서 반복되는 구조를 재사용하는 디자인 패턴(Design Pattern)과 같은 객체기술을 적극적으로 도입해서 소프트웨어 개발 생산성을 향상시켜가는 추세이다. 객체기술이 소프트웨어의 개발 생산성과 품질향상에 효과적인 개발 기법이라는 하지만, 실제로 단기간 내에 기업의 요구에 따른, 혹은 비즈니스 도메인의 변화에 따른 애플리케이션의 수정보완 및 추가개발이나, 시스템 업그레이드 작업은 여전히 매우 어렵고 때로는 노력과 비용이 많이 요구되는 작업이 되기도 한다.

기업에서 요구하는 이러한 정보시스템 구축은 일반적인 SDLC(System Development Life Cycle) 모델에 의해서 개발 및 도입되고 있는데, 이러한 SDLC 모델은 시스템 계획 및 선정, 시스템 분석, 시스템 설계, 시스템 구현 및 운영의 4 단계로 개발 단계가 분명하게 구별되어 있고 각 단계의 산출물

이 명확하다는 것이다. 그러나 이러한 정보시스템 개발 방법론은 다음과 같은 한계점을 갖고 있다. 첫째, 분석과 설계 단계의 결과가 구현단계에서 직접적인 코드로의 어떤 도움도 제공하지 못한다는 것이다. 이것은 잘 설계된 시스템이라 하더라도 시스템의 구현에 있어 프로그램 개발자가 순수 시스템 설계를 컴퓨터 언어로 변환해야 한다. 그래서 코드구현을 지원하는 몇몇의 CASE 도구가 도움을 주고 있지만, 기본적으로는 한 줄씩 코드를 작성해 나가야 한다. 둘째, 시스템의 변경을 요하는 요구사항이 있을 때, 재설계와 재구축의 이중 작업을 하여야만 한다. 이것은 소프트웨어의 개발 및 유지보수에 있어서 비효율적이다. 셋째, 기존에 나와 있는 CASE 도구나 프로그램 생성기는 요구사항이 있을 때마다 소스코드를 생성하고 컴파일하며 실행모듈을 만들어서 배포하고 있는데, 이것은 반복적이고 정적인 작업이어서 잦은 기업의 요구사항들에 능동적으로 대처하기가 어렵고, 시간이 많이 요소 되며, 최종사용자 입장의 요구사항들을 실시간으로 반영하지 못한다.

따라서 이러한 기존의 SDLC 기반의 정보시스템 개발상의 문제점 및 애로사항들을 체계적이고 효과적으로 관리하기 위하여, 시스템 분석설계 정보 및 관련 정보시스템의 구성요소들을 식별 및 정의하여 메타 데이터(Meta-data)화 하고, 이러한 메타 데이터를 체계적으로 관리하고 활용하기 위한 리파지토리 시스템(Repository System)에 대한 연구가 필요하다.

II. 관련 문헌연구

본 연구와 관련된 기존 연구들은 크게 애플리케이션 자동 생성 관련 분야와 메타 데이터 활용 분야의 연구가 있어 왔다.

먼저, 애플리케이션 자동 생성 관련 분야에서는 웹 환경에서 애플리케이션의 구성 요소인 폼이나 리포트 같은 객체를 자동 생성시켜주는 소프트웨어 개발 도구나 생성기(Generator) 설계에 대한 연구가 주로 이루어져 왔다. 기존의 이러한 연구들을 간략하게 살펴보면, 다음과 같다. 조준구 등[1]은 인터넷 환경에서의 e-Business 상황에서 거래 주체들 간에 교환되어지는 XML 기반의 비즈니스 문서를 생성하기 위한 Form 생성기에 관하여 연구하였다. 홍영일 등[2]은 자바 서블릿을 이용해 웹 플랫폼 상에서 실행이 될 수 있는 전자 결제 시스템을 위한 기안문 생성기를 설계하고 구현하였다. 한상훈 등[3]은 SGML에서 규정하는 DTD 작성 규칙에 따라 DTD를 생성 및 편집할 수 있고, 한글 처리가 가능한 GUI를 기반으로 하는 DTD 생성기를 설계 및 구현하였다. 음두현 등[4]은 웹 기반 데이터베이스 응용의 생산성 향상을 위해, 사용자 인터페이스로 이용될 HTML 폼들과 이 폼들을 통해 이루어질 질의를 처리하는 EJB 및 JSP 컴포넌트들을 자동적으로 생성하는 도구인 WebSiteGen2를 제시한다. 황기태[5]는 데이터베이스나 기타 데이터 소스로부터 출력할 리포트를 설계하고 프린트하는 리포트 생성기의 기능을 가진 MoonLight 시스템을 설계 및 구현하였다. 그러나 위에서 설명한 이러한 연구들은 단지 애플리케이션의 특정 모듈만을 자동 생성시켜주는 개념이거나 개발도구이므로, 다양하고 복잡한 기업용 애플리케이션을 통합적으로 생성하고 지원하는 데에는 한계가 있다.

다음으로, 위에서 설명한 부분적인 애플리케이션 객체 생성기 개발의 한계를 극복하고, 응용프로그램 개발 지원 및 데이터 통합 관리를 위하여, 기존의 많은 연구들에서는 메타 데이터 개념을 사용하고 있다[6]. 메타 데이터란 데이터를 위한 데이터로서, 본 논문에서는 시스템 분석 및 설계 산출물 정보와 애플리케이션 구현에 필요한 프로그램의 구성요소 및 구조들에 관한 데이터를 의미한다. 최근 이러한 메타 데이터 기반의 애플리케이션 구축에 관한 몇몇 연구가 국·내외적으로 이루어져 왔다. 김치수 등[7]은 비즈니스 애플리케이션을 사용자 정의에 맞추면서, 소프트웨어의 개발이 신속하고 유연하게 이루어지도록 비즈니스 로직을 인식하여 응용프로그램을 생성시켜주는 도구를 설계하였다. 응용프로그램 생성기의 핵심 아이디어는 같은 영역에 있는 비즈니스 애플리케이션을 공통의 비즈니스 로직과 화면표시 로직으로 나누어 인식하고, 시스템 설계의 결과를 메타데이터로 저장하며, 저장된 메타데이터를 사용해서 요구사항에 맞는 비즈니스 애플리케이션을 구축하거나 커스터마이징하도록 하였다. 그러나 이 연구는 메타 데이터 기반의 애플리케이션 구축을 위한 개념 설계 및 프로토타입 구축 수준의 연구이므로, ERP 시스템과 같은 실제 복잡한 기업용 애플리케이션 구축에 적용하기에는 한계가 있다. 이희석 등[8][9]은 메타 데이터를 기반으로 하는 기업 리파

지토리 시스템(Enterprise Repository System)의 개념을 제시하고, 제시된 기업 리파지토리 시스템의 아키텍처 및 컴포넌트에 근거하여, ERP 시스템을 위한 리파지토리 시스템의 프로토타입을 제시하였다.

Kevin[10]은 향후 애플리케이션 개발에 있어서 메타 데이터의 중요성을 설명하고, 메타 데이터가 어떻게 애플리케이션의 설계 및 개발에 적용할 수 있는지에 대한 개념적인 활용 방안을 제시하였다. Ferreira와 Moura-Pires[11]는 정보 시스템의 다양한 구성 요소들과 계산 프로세스들의 통합 및 모형화 방안과, 그리고 기업의 사업 및 경영 모델, 인적 물적 자원, 그 밖의 다른 경영자원을 통합 관리하기 위한 메타 데이터 리파지토리 아키텍처를 제시하였고, 이의 적용가능성을 입증하기 위하여 사례연구를 실시하였다. Ferreira et al.[12] 및 Schindler와 Diepenbroek[13]는 웹 정보시스템을 위한 XML 기반의 메타 데이터의 정의하고, 이를 위한 지파지토리 아키텍처를 제시하였다.

그 밖에도 메타 데이터 개념을 활용하여 조직의 정보 통합 및 시스템 통합, 그리고 데이터 웨어하우스 구축에 효율적으로 적용한 다양한 연구들이 있어 왔다[14][15][16][17][18].

그러나 위의 연구들이 기업 애플리케이션을 위한 이상적인 메타 데이터 리파지토리의 개념 및 아키텍처를 잘 제시하였지만, 복잡한 기업 업무들에 적용하기에는 한계가 있다.

III. 메타데이터 기반의 기업용 애플리케이션 구축 방안

본 논문에서 제시하고자 하는 애플리케이션 개발 방법론은 현업의 요구사항 정보, 시스템 분석 및 설계 정보, 즉 화면설계 정보, 프로그램 로직 정보, 데이터베이스 관련 정보, 메뉴 정보, 유저 관련 정보, 그리고 이들의 관계 정보들을 메타데이터화 해서 직접적으로 애플리케이션 구현에 활용하고자 하는 것이다. 이것을 위하여 메타 데이터를 기반으로 동적으로 애플리케이션을 생성하는 리파지토리 시스템을 제시하였고, 이것에 의하여 메타데이터에 대한 사용자 정의만으로 애플리케이션의 주요 구성요소인 메뉴, 폼, 비즈니스 로직, 그리고 데이터베이스 등 응용프로그램 가동에 필요한 모든 요소가 정의되고, 이와 더불어 애플리케이션이 동적으로 생성되고 가동된다. 여기서, 메타데이터란 애플리케이션 관련 정보를 데이터화 한 것을 의미하고, 이 메타데이터를 이용하여 애플리케이션을 동적으로 생성시켜주는 프로그램 모듈을 리파지토리 시스템이라고 부른다. 기존의 응용프로그램 개발 방식에서의 프로그램 품질은 주로 개발자의 기술에 의존되고 있지만, 본

논문에서 제시하는 방법론은 메타데이터의 정의 수준에 따라 응용프로그램의 품질이 결정되고, 응용프로그램의 변경 또한 메타데이터의 변경만으로 이루어진다. 따라서 본 논문에서 제시하는 애플리케이션 개발 방법론은 비즈니스 애플리케이션을 사용자 요구사항에 맞추면서, 애플리케이션 개발자의 과중한 업무를 줄여주고, 애플리케이션 개발을 보다 용이하게 하는데 기여한다.

본 논문에서 제시하는 메타데이터 기반의 리파지토리 시스템을 이용한 애플리케이션 구축의 기본 개념은 일반적인 정보 시스템 개발 수명주기인 SDLC 프레임워크(Framework)의 구현 및 운영단계에서 이전의 시스템 설계 단계의 산출물 정보를 메타데이터화 하고, 이 정보를 이용하여 리파지토리 시스템에서 유저별/권한별로 설정된 애플리케이션을 생성하고, 가동하며, 관리하게 해 주는 것이다. 제시된 애플리케이션 구축 프로세스는 다음 <그림 1>과 같다.

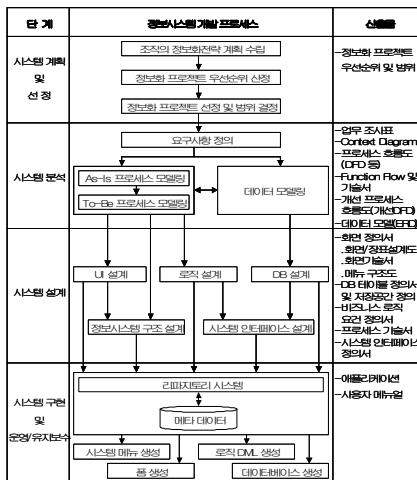


그림 1. 애플리케이션 구축 프로세스
Fig. 1. Application Development Process

• 단계1: 시스템 계획 및 선정

시스템 계획 및 선정 단계는 조직 전체 시스템에 대한 요구를 파악하고 조정하여 정보시스템에 대한 잠재적 프로젝트를 정의하고, 이들의 우선순위를 산정하며, 이것을 기반으로 향후 구축될 정보화 프로젝트를 선정하고, 선정된 프로젝트의 범위 및 소요자원, 일정계획 등을 수립하는 단계이다. 이 단계의 주요 산출물은 조직의 정보시스템 구축계획서, 프로젝트별 우선순위 및 그에 따른 프로젝트 선정 및 그것의 범위 등을 기술한 프로젝트 기본계획서 등이다. 여기서 선정된 애플리케이션 개발 프로젝트에 대하여 다음 단계에서 사용자 요구사항 정의 및 시스템 분석에 들어간다.

• 단계2: 시스템 분석

시스템 분석 단계에서는 크게 세 가지 활동이 수행된다. 첫 번째가 조직 및 사용자의 요구사항을 정의하는 것이다. 이것은 사용자와의 인터뷰 및 업무 양식들을 기반으로 업무조사표를 작성함으로써 이루어진다. 두 번째는 대상 업무를 이해하기 위하여 현행 업무 프로세스 분석(As-Is 분석)을 수행하고, 조직의 요구사항을 반영하여 목표 업무 프로세스(To-Be 프로세스)를 정의한다. 세 번째는 조직의 요구사항 및 업무 분석을 기초로 하여 데이터 모델링을 수행한다. 분석 단계의 주요 산출물은 업무 조사표, Context Diagram, 현행 및 개선 프로세스 흐름도(DFD 등), Function Flow Diagram 및 Function 기술서, 데이터 모델(ERD 등) 등이다.

• 단계3: 시스템 설계

시스템 설계 단계에서는 시스템 분석 단계에서 개발하기로 결정한 시스템의 구현을 위하여 요구되는 컴퓨터 플랫폼과 기술적 내역들로 변환하는 단계이다. 주요 산출물은 화면 및 장표 설계도, 화면 기술서 등의 화면 정의서와 메뉴 구조도, DB 테이블 정의서, 비즈니스 로직 정의서 및 프로세스 기술서, 정보시스템 구조 정의서, 그리고 시스템 인터페이스 정의서 등이다.

• 단계4: 시스템 구현 및 운영

시스템 구현 및 운영 단계에서는 설계 단계에서 산출된 결과물들을 리파지토리 시스템을 이용하여 메타데이터로 저장하고, 이 메타데이터를 이용하여 애플리케이션을 생성하고 구현하는 단계이다. 그리고 테스트 및 운영 과정을 통하여 추가 요구사항이나 수정 사항들에 대해서도 메타데이터를 수정함으로써 시스템 개발을 완성하게 된다. 이 단계의 주요 산출물은 메타데이터 DB, 애플리케이션, 사용자 매뉴얼 등이다.

IV. 메타데이터 기반의 리파지토리 시스템의 개발 사례

1. 리파지토리 시스템의 개념 및 프레임워크

메타데이터 기반 리파지토리 시스템은 애플리케이션이 윈도우 상에 로드 되는 시점에 메타 데이터의 정보를 리파지토리 시스템으로부터 호출하여 폼 객체들을 생성하고, 사용자의 요구에 따라 DML문을 생성하고 실행한다. 그리고 애플리케이션이 윈도우 상에서 종료되는 시점에 폼의 객체와 DML문은 메모리 상에서 해제된다. 본 논문에서의 기업용 애플리케이션

이런 생성을 위한 리파지토리 시스템의 프레임워크는 다음 <그림 2>과 같다.

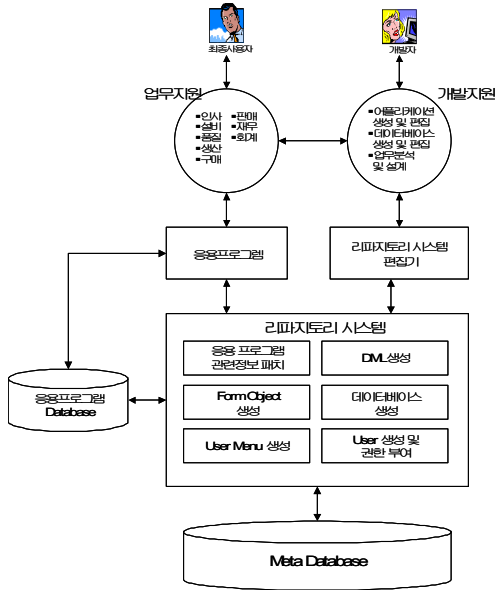


그림 2. 기업용 애플리케이션을 위한 리파지토리 시스템 프레임워크
Fig. 2. A Framework of Repository System for Enterprise Applications

• 동적인 DML문 구현

애플리케이션에서 동적인 DML문 구현은 메타데이터와 리파지토리 시스템의 상호작용이 필수적이며 이러한 상호작용은 애플리케이션 상에서 품의 객체와 메타정보 상에서 이루어진다. 하지만 상호작용을 하기 위하여서는 애플리케이션으로부터 특정한 데이터를 입력 받거나 리파지토리 시스템이 특정한 데이터를 제공하여야 한다. 즉 애플리케이션에서 사용되는 DML문의 스크립트(Script)는 사전에 만들어져 있는 것이 아니라 사용자가 애플리케이션을 사용할 때 자료검색, 저장, 변경 등의 요청이 있으면, 리파지토리 시스템으로부터 호출하여, DML문 스크립트를 생성시키고, 실행한다. 실행 후에는 애플리케이션의 메모리에서 DML문 스크립트는 해제된다.

• 동적인 품 객체(Form Objects) 구현

애플리케이션에서의 사용되어지는 품 객체들의 정보는 메타데이터화 되고, 품의 실행시 품의 각각의 객체들의 정보를 리파지토리 시스템으로부터 호출하여, 객체를 위치, 속성명 등의 정보를 기반으로 생성시키고, 사용자의 요구에 해당하는 자료를 검색 또는 등록 및 처리할 수 있도록 한다. 그리고 사

용자가 리파지토리 시스템에서 품의 정보를 갱신하고, 품을 재실행할 때에 변경된 정보에 의한 품의 객체가 생성된다. 품을 종료하면 모든 객체들은 메모리상에서 해제된다. 다음 표는 위의 <표 1>에서 리파지토리 시스템의 세부 컴포넌트 모듈을 설명한 것이다.

표 1. 리파지토리 시스템의 컴포넌트의 상세
Table 1. Detailed Components of Repository System

Component	Component의 세부 모듈	설명
응용 프로그램 관련정보 패치	ColumnAlign	칼럼의 수평정렬
	ColumnTable	칼럼의 정보
	PgmSumLevel	조회 프로그램 소계 개수
	ProgramType	프로그램 타입
	SearchWinPos	검색 객체의 위치정보
	SheetColLock	시트 칼럼 LOCK
Form Object 생성	ColumnLength	폼에서의 객체 길이
	ColumnTitleLen	폼에서의 타이틀 객체의 길이
	ConSearchObject	조건검색에 사용할 객체
	HeadTitleCnt	조회 폼에 사용할 타이틀의 개수
	SheetHeadTitle	시트 헤드 항목
User Menu 생성	CreateMenu	메뉴 생성
User 생성 및 권한부여	CreateUserGrant	사용자 계정 및 권한부여
데이터베이스 생성	CreateTableInfo	테이블 생성 및 변경
	CreateTablespace	테이블 스페이스 생성 및 변경
DML 생성	ComboAdditemSQL	콤보객체의 Item추가 SQL문
	CreateInsertSQL	Insert DML문 생성
	CreateOptionSQL	기타 SQL문 생성
	CreateOrderSQL	SQL문에서 ORDER BY문 생성
	CreateTableSQL	테이블 SQL문 생성
	CreateWhereSQL	SQL문에서 WHERE절 생성
	TabColJoinType	테이블 조인시 조인 타입

이러한 리파지토리 시스템의 데이터베이스 구조를 설명하는 ERD는 다음의 <그림 3>과 같다. 이 ERD는 전체 리파지토리 시스템 중에서 프로그램 마스터 관리 정보와 프로그램 사용자 및 권한 관련 정보를 표현하고 있는 메타 데이터의 일부분을 표현하고 있다.

2. 리파지토리 시스템 개발

리파지토리 시스템 개발을 위한 기술적 환경을 살펴보면, 다음 <그림 4>과 같다.

리파지토리 시스템의 개발을 위해서 사용된 주요 소프트웨어들에 대한 세부적인 내용은 다음 <표 2>와 같다. 위와 같은 기술적 환경 하에서 개발된 리파지토리 시스템의 세부 기능은 다음 <그림 5>과 같다.

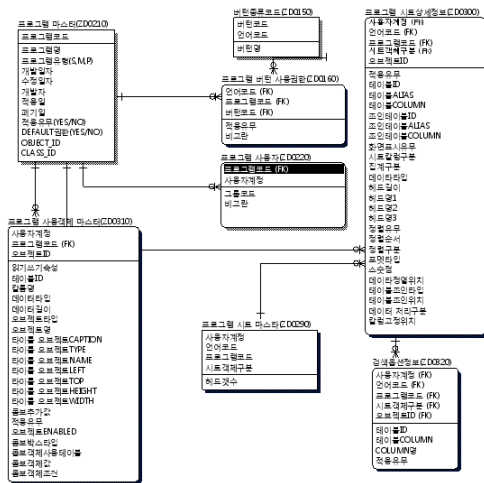


그림 3. 리파지토리 시스템의 ERD
Fig. 3. ERD of Repository System

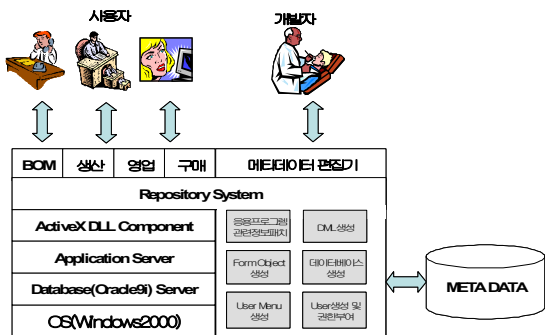


그림 4. 리파지토리 시스템의 기술적 환경
Fig. 4. Technical Environment of Repository System

또한, 본 논문에서 개발된 리파지토리 시스템의 메뉴 구조는 다음 <그림 6>과 같다. 이러한 리파지토리 시스템을 기업 업무에 효율적으로 사용하기 위한 세부 기능 및 절차는 다음 <그림 7>과 같다.

표 2. 리파지토리 시스템의 소프트웨어 세부 사양
Table 2. Detailed Specifications about Softwares of Repository System

구성요소	소프트웨어명	사양	공급처	비고
DBMS	*Oracle 9i *SQL 2000 Server	*Process Lic	*Oracle *Microsoft	30 User 이상
OS Server	*Windows 2000 Server	*P-700 이상 2GB RAM *Dual CPU 80GB HDD	*Microsoft	-
Developer Tool	*SpreadSheet *Crystal Report *Visual Studio 6.0 (Visual Basic 6.0)	*1 Copy	*FeaPoint *Seagate *Microsoft	-

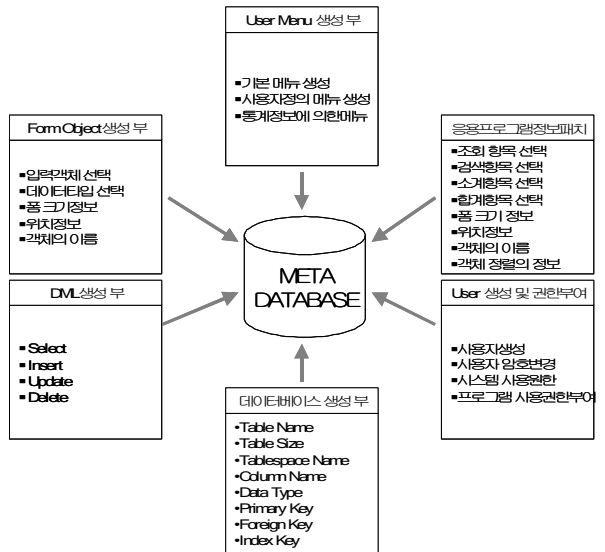


그림 5. 리파지토리 시스템의 세부 기능도
Fig. 5. Detailed Function Diagram of Repository System

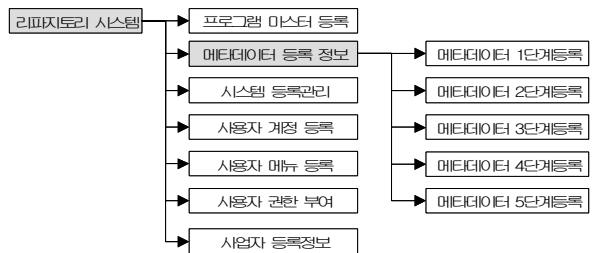


그림 6. 리파지토리 시스템의 메뉴 구조
Fig. 6. Menu Structure of Repository System

지금부터, 본 논문의 리파지토리 시스템에서 구현된 주요 기능들은 리파지토리 시스템의 메인 메뉴, 프로그램 등록, 메타데이터 등록정보, 시스템 등록관리, 사용자 계정등록, 사용자권한 부여, 사업자 등록 화면 등이 있다. 이 중에서 메타데이터 등록 메뉴를 예로 들어 개발 사례를 설명하고자 한다.

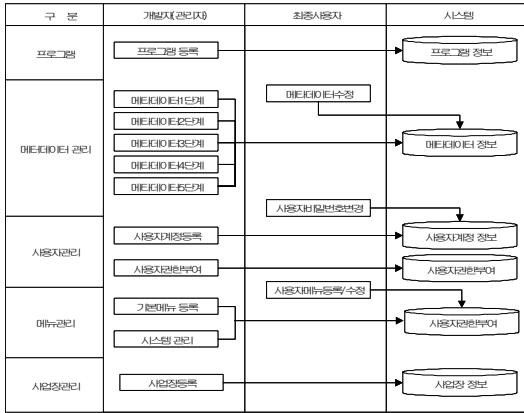


그림 7. 리퍼지토리 시스템의 세부 기능 및 절차
 Fig. 7. Detailed Functions and Usage Procedure of Repository System

그림 8. 메타데이터 정보등록 및 조회 화면
 Fig. 8. A Screen of Entering and Querying Meta-data

〈그림 8〉화면은 메타데이터 정보 등록 및 조회 화면을 보여주고 있고, 또한 메타데이터를 총괄적으로 관리하는 프로그램으로 정보를 등록과 조회를 동시에 처리할 수 있으며, 주 역할은 메타데이터 단계별 등록의 시작 위치가 된다.

그림 9. 메타 데이터 등록 1단계 화면
 Fig. 9. The First Screen of Entering Metadata

〈그림 9〉화면은 메타데이터 1단계 등록 화면을 보여주고 있으며, 사용자계정, 프로그램ID, 타이틀구분이 기본 키 (Primary Key)이며, 사용될 메인 테이블과 조인될 테이블의 정보를 관리한다.

그림 10. 메타 데이터 등록 2단계 화면
 Fig. 10. The Second Screen of Entering Metadata

〈그림10〉화면은 메타데이터 2단계 화면을 보여주고 있으며, 화면표시 유무, 집계구분, 데이터타입, 헤드길이, 헤드명 등을 관리한다.

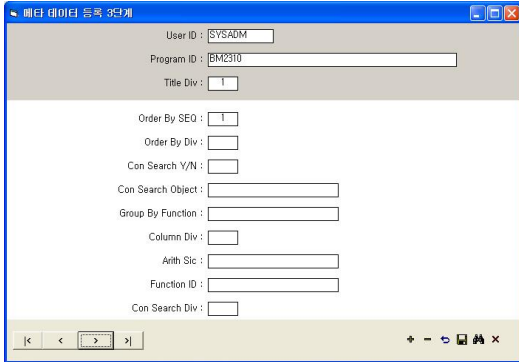


그림 11. 메타 데이터 등록 3단계 화면
Fig. 11. The Third Screen of Entering Metadata

〈그림 11〉화면은 메타데이터 3단계 화면을 보여주고 있으며, SELECT문장에서 ORDER BY를 할 때 순서와 ASC(오름차순), DESC(내림차순)을 결정하고, 검색 콤보 버튼의 객체를 등록한다.

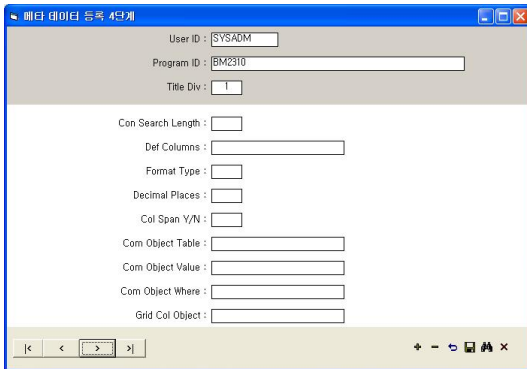


그림 12. 메타 데이터 등록 4단계 화면
Fig. 12. The Fourth Screen of Entering Metadata

〈그림 12〉화면은 메타데이터 4단계 화면을 보여주고 있으며, DISPLAY될 데이터의 FORMAT과 소숫점 관리, 콤보 객체에 사용될 테이블, 값, 조건절 그리고 그리드 객체 칼럼의 사용될 객체를 선택한다.

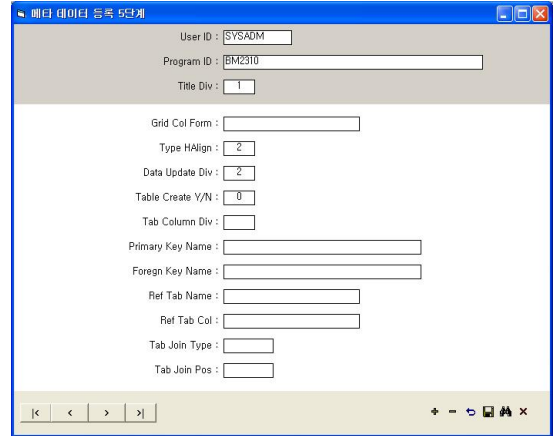


그림 13. 메타 데이터 등록 5단계 화면
Fig. 13. The Fifth Screen of Inputting Metadata

〈그림 13〉화면은 메타 데이터 5단계 등록 화면을 보여주고 있으며, 프로그램에 사용될 DML 문장 중에서 처리를 어떻게 할 것인지와 테이블을 생성 유무, 그리고 테이블이 생성될 경우 기본 키와 외래 키(Foreign Key)의 명을 등록하고, 참조될 테이블의 정보, Outer-Join을 할 경우 처리를 등록한다.

이렇게 앞에서 설명한 리파지토리 시스템의 기능과 등록 화면을 이용하여 구현하고자 하는 업무 도메인의 애플리케이션에 대한 시스템 분석설계 산출물 정보를 메타데이터화 하여 입력 등록한다.

다음의 〈그림 14〉은 이러한 메타 데이터와 리파지토리 시스템을 기반으로 본 연구의 대상 기업인 'H' 자동차부품 제조업체의 애플리케이션 구축 모듈 중에서 가장 중요하면서 기본적인 제품정보를 관리하는 생산기술 BOM(Bill of Materials) 응용프로그램 화면을 보여주고 있다. 현재 연구 대상 기업인 'H' 기업은 본 논문에서 제시하는 메타 데이터 리파지토리 기반 하에 전사적인 ERP시스템 자체개발을 진행 하였고, 현재 완료되어 성공적으로 가동 중에 있다.

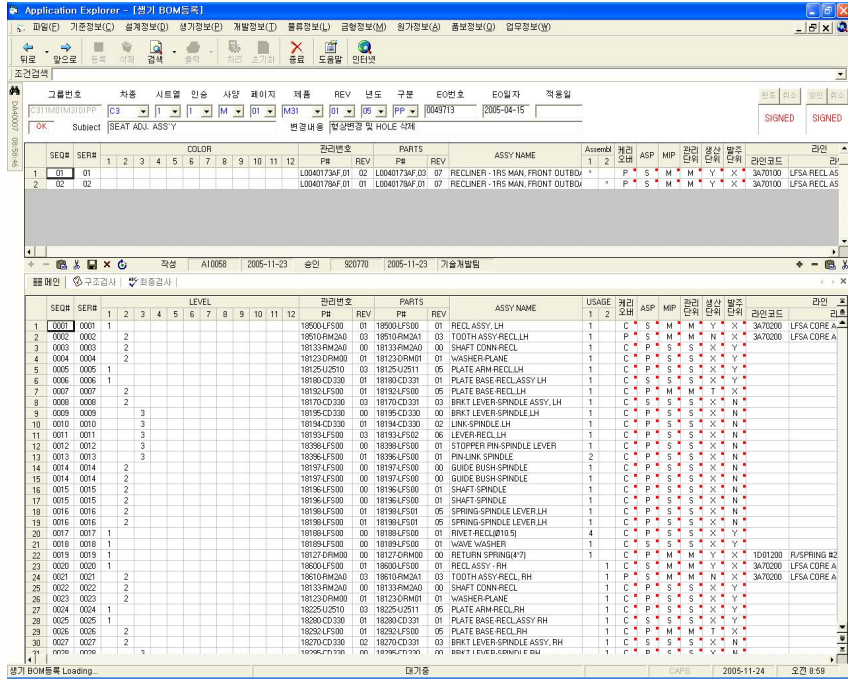


그림 14. 생산기술 BOM 진행현황 화면
Fig. 14. A Screen of Engineering BOM

3. 기존 방법론과 비교한 제안 방법론의 장점

구조적 방법론, 정보공학 방법론, 그리고 객체지향 방법론과 같은 기존의 정보시스템 개발 방법론(19)(20)은 크게 요구사항 파악 및 업무분석, 시스템 분석, 시스템 설계, 그리고 시스템 구현 및 운영 단계로 시스템이 개발된다. 이러한 정보시스템 개발 절차의 기본적인 개념은 각 단계의 결과물이 다음 단계의 입력물이 된다는 것이다.

여기서 주요 문제점 중의 하나는 설계 단계의 산출물을 구현 단계의 입력물로 직접적으로 활용하기 어렵다는 것이다. 즉 시스템 분석가가 시스템 설계서를 작성한 후 개발자가 설계서 내용을 검토하고 이해하여 프로그래밍 하게 되는데, 이때 시스템 분석가와 개발자 사이의 이해력에 있어서 불일치가 발생할 수 있다. 또한 개발 중에 수정 및 추가요구나 운영 중에 유지보수로 인한 작업이 많을수록 설계 결과와 개발결과의 불일치는 커진다고 볼 수 있다.

이것과 비교하여, 본 논문의 메타 데이터 기반 애플리케이션 개발 방법은 정보시스템의 분석 및 설계 단계에서의 산출물 정보를 메타 데이터화 하여 리파지토리 시스템에 축적

을 하고, 이것을 시스템 구현시 직접적으로 사용함으로써 시스템 설계 단계와 구현 단계 사이의 불일치를 줄이고, 또한 메타데이터의 조작을 통하여 누구나 쉽게 애플리케이션을 구축 및 수정할 수 있고, 유지보수가 용이하며, 따라서 EUC(End User Computing)를 달성할 수 있다는 장점이 있다. 이러한 사항은 'H' 자동차부품 제조업의 ERP 시스템 구축 시에 시스템 분석가 및 개발자들에 의하여 확인되었다.

위에서 간략하게 설명한 기존 정보시스템의 개발방식과 비교하여 본 논문에서의 개발방식의 장점을 애플리케이션의 관리 항목별로 비교하면, 위의 <표 3>와 같다.

표 3. 기존 방법론과 제안 방법론의 비교
Table 3. Comparison between existing methodology and proposed methodology

항목	기존 방법론	제안 방법론
개념 비교	시스템 분석가 작성한 시스템 분석 및 설계 정보를 개발자가 검토하고 이해하여 전적으로 코딩을 통한 애플리케이션을 구축함.	시스템 분석 및 설계 정보를 메타데이터화 하여 축적하고, 이것을 기반으로 한 리파지토리 시스템을 이용하여 시스템을 구현함. 즉, 메타데이터 조작에 의하여 시스템을 구현함.

개발 용이성	개발환경 및 개발자의 능력에 의해 좌우됨. 필요로 하는 개발자를 구하기가 어려움.	초급 개발자도 업무분석 및 설계 자료를 참고로 메타데이터를 생성만 해도 고품질의 프로그램 개발 가능
유지관리	추가 개발 및 수정 요구시 재코딩, 재컴파일, 재배포 등 많은 노력이 요구됨.	메타데이터를 리퍼지토리 시스템에서 관리하기 때문에, 개발된 시스템에서는 별도의 유지관리가 적고 용이함.
개발 생산성	개발자의 숙련도 및 업무 능력에 좌우.	개발자의 숙련도 보다는 업무분석 및 시스템설계 능력에 좌우. 메타 데이터의 정합성에 좌우.
확장성	시스템 확장시 재코딩, 재컴파일, 재배포 등 많은 추가 작업으로 확장성이 떨어짐.	ActiveX Component DLL로 구성되어 있기 때문에 확장성이 용이함. 메타데이터 정의만으로 확장이 가능.
업무 표준화	시스템 분석 및 설계 단계의 산출물의 문서화 및 구현된 비즈니스 로직에 의해 결정. 업무 변경시 업무 표준화 수정 및 시스템 반영이 어려움.	업무분석 및 설계 단계의 산출물들에 대한 메타데이터화에 의해 결정됨. 업무 변경시 이러한 정보를 메타데이터 수정만으로 가능하기 때문에 업무 표준화 수정 및 시스템 반영이 용이함.
재활용 용이성	완전한 객체기반, 컴포넌트 기반으로 애플리케이션이 구축되어 있지 않는 한 재활용 측면에서는 한계가 있음.	메타데이터가 템플릿 개념을 갖고 있기 때문에 재활용 100% 가능함
유연성	사전에 코딩된 형태로 실행되기 때문에, 새로운 환경, 새로운 업무에 적용하기 위해서는 재코딩, 재컴파일, 재배포 하여야 함.	메타데이터에 의해 프로그램이 생성되기 때문에, 실행시에 동적으로 다양한 형태의 프로그램이 구현됨. 새로운 업무환경에서도 메타데이터의 수정에 의해서 시스템 구현이 가능함.

V. 결론 및 향후 연구방향

본 논문에서는 기업용 애플리케이션을 효율적이고 효과적으로 구현하기 위한 메타데이터 기반의 리퍼지토리 시스템의 개념 및 프레임워크를 제시하고, 개발하였다. 제시된 리퍼지토리 시스템은 동질의 업무 및 시스템 개발 환경 하에서 애플리케이션의 분석 및 설계 정보를 메타 데이터 리퍼지토리에 저장하고, 이것을 구현 단계에서 직접적으로 이용하여 동적으로 애플리케이션을 생성시켜 주는 시스템이다. 제시된 리퍼지토리 시스템의 유용성을 보여주기 위하여, 자동차 부품업체인 'D'사의 ERP 시스템 구축에 적용되었다. 그 결과, 기업의 방대하고 복잡한 ERP 시스템을 메타데이터

기반 리퍼지토리 시스템 상에서 성공적으로 구현할 수 있었다. 본 논문에서는 제조업체의 ERP시스템에서 가장 기본적인 기능인 생산기술 BOM관리 화면을 그 예로서 보여주었다. 그러나 본 논문에서 제시한 리퍼지토리 시스템은 ANSI나 ISO의 IRDS(Information Resource Dictionary Systems)와 같은 국제표준화기구에서 제시하는 이질적이고 분산된 기업의 정보자원들을 통합적으로 관리하는 일반적인 리퍼지토리 시스템의 표준 프레임워크가 아니고, 주로 동질의 개발환경에서 애플리케이션 구현을 동적으로 생성시켜 주는 한정된 개념만을 제시하고 구현하였다.

따라서 향후 연구방향은 제시된 메타데이터 기반의 리퍼지토리 시스템을 이용하여 ERP 시스템뿐만 아니라 CRM, SCM, PLM(Product Lifecycle Management) 등의 확장형 ERP 시스템과 같은 애플리케이션들에 적용해서 메타데이터를 축적하여 기업용 애플리케이션에 대한 표준 메타데이터 및 리퍼지토리 시스템을 위한 템플릿을 구축하는 것이다. 더 나아가서 이질적이고 분산되어 있는 전사적인 기업 정보 및 지식 자원을 관리할 수 있는 기업 통합정보시스템 구축을 위한 메타데이터 및 리퍼지토리 시스템 구현으로 나아가 할 것이다. 현재에는 EPR 시스템에 대한 메타 데이터 및 그 리퍼지토리 시스템의 프레임워크가 제시되고 개발되어 있으나, 앞으로도 계속적으로 추가 연구가 진행 중에 있고, 멀지 않아 의미 있는 결과를 제시할 수 있을 것으로 사료된다.

참고문헌

- [1] 조준구, 김창수, 정희경, "E-비즈니스 XML 문서 생성을 위한 Form 생성기의 설계 및 구현," 한국정보과학회 2001년도 봄 학술발표논문집, 제28권, 제1호(A), 313-315쪽, 2001년 4월.
- [2] 홍영일, 강상기, 박은영, 최진석, 김수정, 이정배, 이광우, "서블릿을 이용한 기안문 생성기의 설계 및 구현," 한국멀티미디어학회 춘계학술발표논문집, 제2권, 제1호, 405-409쪽, 1999년.
- [3] 한상훈, 이만형, 박세원, 황대훈, "GUI를 기반으로 하는 SGML DTD 생성기의 설계 및 구현," 한국멀티미디어학회 춘계학술발표논문집, 395-400쪽, 1998년.
- [4] 음두현, 고민정, 강이지, "컴포넌트 기반 웹 데이터베이스 응용의 자동 생성기," 정보처리학회논문지 D, 제11-D권, 제2호, 371-380쪽, 2004년 4월.
- [5] 황기태, "XML 폼 구조를 기반으로 하는 리포트 생성기의 설계 및 구현: MoonLight," 정보처리학회논문지

D, 제10-D권, 제4호, 735-744쪽, 2003년 8월.

[6] A. Sen, "Metadata management: past, present and future," Decision Support Systems, Vol. 37, pp.151-173, 2004.

[7] 김치수, 오은진, "메타데이터 기반 응용프로그램 생성기 설계," 정보처리학회논문지 D, 제11-D권, 제7호, 1477-1482쪽, 2004년 12월.

[8] 이희석, 서우중, 김태훈, 이충석, 손명호, 백종명, 손주찬, 박성진, "기업 리퍼지토리 시스템 : 아키텍처 및 ERP 리퍼지토리 사례," 정보기술과 데이터베이스저널, 제7권, 제1호, 1-15쪽, 1999년 4월.

[9] 이희석, 이재, 이충석, 조창래, 손주찬, 백종명, "전사적 자원관리 시스템을 위한 기업 리퍼지토리 시스템 구축," 경영정보학연구, 제9권, 제1호, 60-75쪽, 1999년 3월.

[10] S. P. Kevin, "Metadata-Driven Application Design and Development," Microsoft Architect Journal, pp.28-38, Jan. 2004.

[11] R. Ferreira, and J. Moura-Pires, "Extensible Metadata Repository for Information Systems and Enterprise Applications," ICEIS 2007 - 9th International Conference on Enterprise Information Systems, Funchal, Portugal, 2007.

[12] R. Ferreira, J. Moura-Pires, R. Martins, and M. Pantoquilha, "XML Based Metadata Repository for Information Systems," EPIA 2005 - 12th Portuguese Conference on Artificial Intelligence, Covilha, Portugal, 2005.

[13] U. Schindler, and M. Diepenbroek, "Generic XML-based framework for metadata portals," Computers & Geosciences, Vol. 34, pp.1947-1955, 2008.

[14] 김기운, "사용자의 데이터 웨어하우스 접근과 활용을 위한 메타데이터 관리 사례," 한국컴퓨터정보학회논문지, 제12권, 제5호, 225-233쪽, 2007년.

[15] 김재수, 박형서, "분산데이터베이스를 이용한 메타데이터 기반 실과 교수학습자료 공유 시스템 설계," 실과교육연구, 제13권, 제2호, 185-203쪽, 2007년.

[16] 김종우, 김형도, "기업간 비즈니스 프로세스의 통합적 등록저장을 위한 메타데이터 스키마 설계," 한국전자거래(CALS/EC)학회지, 제12권, 제2호, 195-217쪽, 2007년.

[17] 나민영, "군 정보통합을 위한 메타데이터 기반의 데이

터 그리드 시스템," 한국컴퓨터정보학회논문지, 제13권, 제2호, 95-103쪽, 2008년.

[18] 이상문, 서정민, "지하수관리시스템의 공간 메타데이터 모델에 관한 연구," 한국컴퓨터정보학회논문지, 제12권, 제4호, 229-237쪽, 2007년.

[19] 강성배, 문태수, "객체지향 방법론을 이용한 자동차부품기업의 영업관리시스템 설계 및 구현," 정보시스템연구, 제13권, 제1호, 77-95쪽, 2004년 6월.

[20] 남기형, 김선호, "확장된 컴포넌트 개념에 의한 정보시스템 개발방법론," 한국전자거래(CALS/EC)학회지, 제2권, 제1호, 55-77쪽, 1997년.

저자 소개



장길상

1986: 울산대학교
산업공학과 공학사.

1988: 한국과학기술원
산업공학과 공학석사.

1997: 한국과학기술원
경영정보공학과 공학박사

현재: 울산대학교
경영정보학과 교수

관심분야 :
기업정보시스템, CBR, DB응용,
정보시스템 개발방법론, 6시그마,
ERP, e-Business, 등