

레거시시스템의 마이그레이션을 위한 지원도구의 설계 및 구현

양 해 승[†] · 박 병 형^{††} · 양 해 술^{†††}

요 약

소프트웨어 개발비용의 증대로 레거시시스템을 재사용하기 위한 방안과 이를 지원하기 위한 자동화 도구에 관한 관심이 고조되고 있다. 따라서, 본 논문에서는 레거시시스템에 대한 정의와 개념 나아가 등장 배경을 실질적인 시장과 공급사와 기술적 동향을 통해 이를 밝히고 있으며, 레거시시스템을 처리하는 구체적인 개발절차 그리고 구축사례, 평가기준과 이를 효율적으로 처리하기 위한 자동화 도구를 설계하고 개발하였다. 또한, 본 논문에서 제시하는 LM(Legacy to Modern System) 지원도구를 활용하여 레거시 시스템을 분석하고 후보 컴퍼넌트를 추출하여 정제 한 다음 저장하여 재사용하는 과정을 구현하였다. 본 LM지원도구를 레거시 시스템의 마이그레이션에 적용할 경우 소프트웨어 재사용의 효과를 통해 비용절감, 개발기간 단축 나아가 기존의 지적재산을 승계하는 효과를 기대할 수 있다.

키워드 : 레거시 시스템 마이그레이션, 레거시 개발방법론, 프로그램 변환

Design and Implementation of Tool Constructing Migration of Legacy System

Hae-Seung Yang[†] · Pyung-Hyung Park^{††} · Hae-Sool Yang^{†††}

ABSTRACT

Due to increased software development cost, interest about the method and automated tool for reusing legacy system have been running high. Therefore, In this study, it proves the definition, concept and also background of legacy system by showing real market and technology trend. Automation tool is designed and developed to handle the concrete development methodology, procedure and reference. Also, we implemented the process that analyze the legacy system and extract, refine, store and reuse the candidate components by using LM(Legacy to Modern System) supporting tool presented in this paper. If LM supporting tool is applied to legacy system migration, we can expect cutting costs, shortening the developing period and succeeding the intellectual property by software re-engineering.

Key Words : Legacy System Migration, Legacy Development Methodology, Program Conversion

1. 서 론

21세기에 이르러 기업은 e-비즈니스 모델을 어떻게 추진 하느냐에 따라 성공 여부가 결정되게 되었다[1]. 기업의 경영은 정보화시스템을 어떻게 효율적으로 적용하느냐에 따라 기업의 성과가 결정된다[2]. 1960년대 말에 이르러 소프트웨어가 미국 NATO를 중심으로 공학기법을 적용한다는 원칙 아래 체계적인 작업을 시작하였고 프로그래밍어, 데이터베이스 등이 하드웨어와 접목되면서 메인프레임시대를 여는데 기여하였다. 그 이후 많은 하드웨어산업이 기하급수적인 발

전을 하면서 클라이언트/서버시스템을 통하여 다운사이징이 시작되었지만 소프트웨어의 기본골격은 기존방식을 그대로 유지하고 있다.

또한 세기말 제앙인 Millenium Bug(이하, Y2K)는 대부분의 기업들이 정보기술 분야에 투자를 위축시키는 요인으로 작용하였지만, Y2K 문제가 원활히 해결되어 대부분 기업들이 정보기술 분야에 대규모 투자를 진행하고 있다[3]. 기업은 e-비즈니스와 맞지 않은 메인프레임시스템이나 폐쇄적인 환경의 시스템에 투자하는 것을 억제하고 새로운 오픈 환경 시스템에 관심을 가졌다.

그리고, 기존의 폐쇄적인 환경에서 운영되는 레거시시스템은 반드시 해결하고 넘어가야 할 부분으로 그 중요성이 강조되고 있다. 기업의 신정보시스템을 도입하는 세가지 방법에는 첫째, 새로운 관점에서 업무 분석, 설계에서 구현단계 까지 개발하는 것, 둘째, 패키지화되어 있는 전사적 자원관

※ "본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음." (IITA-2007-(C1090-0701-0032))

† 정 회 원 : 한국IT진흥(주) 엔터프라이즈사업본부 영업본부장 / 이사

†† 정 회 원 : (주)케미스 대표이사/CEO, 미국현지법인 NexxIT(회장)

††† 중신회원 : 호서대학교 벤처전문대학원 교수

논문접수 : 2007년 6월 19일, 심사완료 : 2007년 11월 1일

리시스템(Enterprise Resource Planning, 이하 ERP)을 도입하는 것, 셋째, 레거시시스템을 활용하여 기존에 사용 하였던 소프트웨어를 재구성 형태로 사용하는 것들이 있다. 이 중에서 레거시시스템을 재활용하는 방법이 지적재산 승계, 비용절감, 자원 활용, 개발기간 그리고 신규개발에 따른 위험을 줄이는 방법으로, 가장 이상적이라고 할수 있다.

레거시시스템은 국가별 사회적 환경에 따라 사용하는 형태가 다르다. 국내 역동적인 사회로 전개됨에 따라 소프트웨어 시스템도 목적에 따라 시스템이 생성과 소멸이 매우 긴박하게 이루어져 왔다. 따라서 많은 시행착오, 과도한 투자, 개발 시간의 증가 등 비효율적인 생산체제로 전개되었으며, ERP기술이나, 새로운 Web-enable시스템 도입으로 기업의 중요한 지적자산을 지닌 레거시시스템을 폐기하는 결과를 초래하고 있다.[1] 또한, 많은 IT 예산 집행과 신규개발에 대한 위험을 감수하면서 신규로 프로젝트를 개발함으로써, 많은 문제점을 그대로 안고 있는 실정이다. 이와는 반대로 일본, 미국, 유럽 등은 안정된 사회적인 시스템 기반 아래서 체계적으로 개발되어 왔다. 이는 개발자들이 레거시시스템에 대한 관점을 중요한 자산의 가치로 판단하고 있기 때문이다. 정보기술이 발달된 나라들은 레거시시스템에서 비즈니스-규칙을 추출하여 새로운 오픈시스템으로 전개하는 방식의 마이그레이션 체제로 전환의 필요성이 활발하게 전개되고 있다.

즉, 본 논문에서는 다음과 같은 목적을 위해 레거시시스템의 마이그레이션 구축을 효과적으로 추진하기 위한 지원 도구에 관한 연구를 수행하였다.

- 첫째, 기존의 개발 방식보다 비용과 시간 절감, 효율성을 높이는데 기여할 수 있다.
- 둘째, LM(Legacy to Modern System) 지원도구를 활용하여 소프트웨어 프로그램의 90%이상을 차지하는 레거시시스템의 재활용도를 높이는데 기여할 수 있다.
- 셋째, 마이그레이션 지원도구를 구현하여 개발공정의 표준화, 불필요한 요소제거를 통한 작업의 간소화, 작업공정에서 개발자의 의사소통 원활히 할 수 있다.

따라서, 본 논문의 2장에서는 레거시시스템에 대한 개념 및 이론적 배경에 대한 관련 연구를 기술하였으며, 3장에서는 레거시시스템의 마이그레이션을 위한 지원도구 구현에 관하여 기술하였다. 끝으로, 4장에서는 정리된 개발방법론에 의한 지원도구를 개발하여 실제로 적용사례와 기대효과에 대해 기술하였다.

2. 관련 연구

2.1 레거시시스템 재사용 방법론

기존 레거시시스템의 재사용을 지원하는 방법론으로는 CORVMII(Common Object Based Reengineering Unifield Model), Renaissance, USRM(UIRich System Redevelopment Methodology), MORALE(Mission Oriented Architecture Legacy Evolution) 등을 들수 있다.

그러나 이와 같은 레거시시스템 재사용 방법은 역공학을 통하여 기존의 레거시시스템 정보가 추출 가능하다는 전제하에 구성되어 있고 프로세스 각 단계의 액티비티와 산출물이 명확히 정의되어 있지 않다[14]. 기존 레거시시스템의 재사용을 지원하는 방법론의 특징을 소개하면 <표 1>과 같다.

2.2 LM 구축 절차

시스템을 한단계 높은 단계로 개발하는데 그 단계를 살펴보면 (그림 1)과 같다.

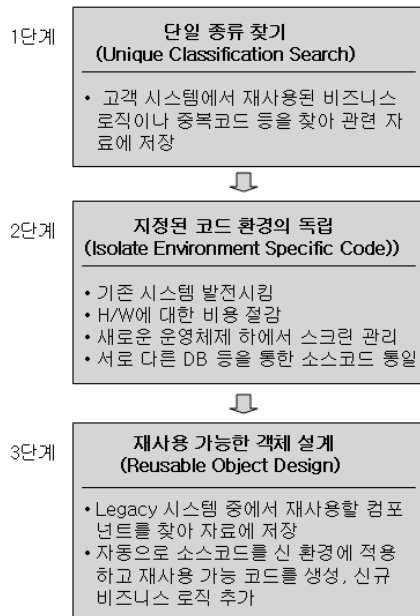
2.3 LM 지원도구의 구현 배경

LM의 영역은 시스템 전체의 마이그레이션에 해당하는 부분이기 때문에 시스템을 통합하는데 거의 전 영역에 해당한다.

본 연구자가 레거시시스템에 대하여 관심을 가진 계기는 Y2K 문제가 레거시시스템인 COBOL, PL/I, C 등의 언어에서 대부분 발생하여 이를 해결하는 지원도구를 개발하면서 부터였다. 이미 프로그램 지원도구를 전문적으로 개발하면서 차츰 마이그레이션 솔루션에 대하여 깊은 관심을 갖게 되었다. 본 연구자 논문인 "CASE Tool개발에 관한 연구"[12]와 "레거시

<표 1> 레거시시스템을 지원하는 방법론

| 방법론 | 특징 | 문제점 |
|-------------|--|---|
| CORVMII | <ul style="list-style-type: none"> • 아카텍처와 코드기반의 재공학 도구들을 통합하기 위한 모델 • 코드 및 코드 구조 모델, 기능 및 개념모델, 아카텍처 레벨 수준으로 변환을 목표 | <ul style="list-style-type: none"> • 소프트웨어 개발 프로세스 전단계를 지원하지 못함 • 구체적인 역공학 방법 및 지침 없음 |
| Renaissance | <ul style="list-style-type: none"> • 진화계획, 구현, 인도, 분배의 4단계로 구성 • 재공학 기술별 기술 레포트 제공 • 단계별 산출물 템플릿 제공 | |
| USRM | <ul style="list-style-type: none"> • 계획, 분석, 포지셔닝, 변화의 4단계 구성 • 데이터, 기능위주의 접근방식 | |
| MORALE | <ul style="list-style-type: none"> • 분석, 설계, 진화의 3단계구성 • 기존 구시스템을 분석하여 부품을 추출하는 방법 • UI를 중심으로 재공학 접근방식 채택 | |



(그림 1) 신 단계 시스템의 개발 과정

시스템을 웹환경 시스템 구축에 관한연구”[13]를 작성하였다. 본 연구에서는 다음과 같이 마이그레이션에 필요한 지원 도구를 개발하였다.

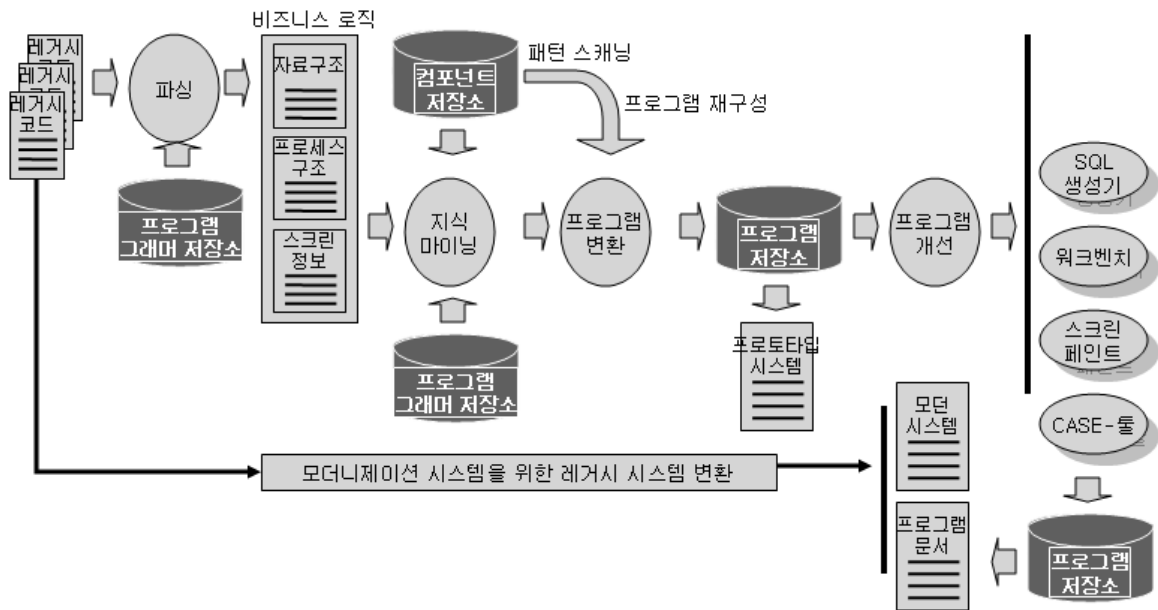
- 프로그램 지원도구 : Java, C#, Visual Basic, COBOL 지원도구
- 데이터베이스 구현도구 : ERD, 스키마 자동생성 도구
- 프로그램 변환도구, AS-IS 분석 도구, 리아키텍처 도구 등을 종합적으로 지원하는 도구를 구현하였다.

3. LM을 위한 지원도구의 구현

3.1 LM 핵심 기술

LM하는데 필요한 요소기술은 매우 다양하며 LM분야의 영역은 소프트웨어 개발단계와 이를 역공학적인 단계까지 광범위하고 다양하다. 전 세계적으로 마이그레이션을 전문으로 하는 공급자의 수가 적고 기술의 정도가 무척이나 깊고 까다로워 검증된 핵심기술을 보유한다는 것은 매우 어렵다.

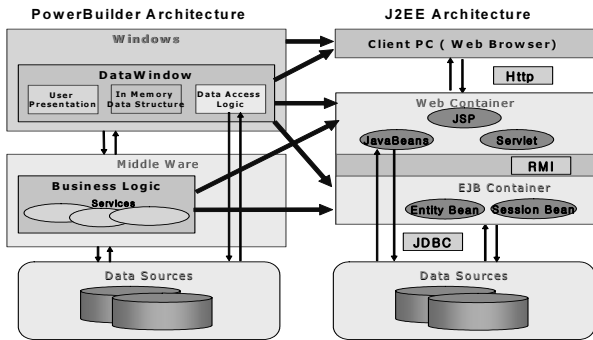
본 연구의 LM지원도구에서 활용하는 핵심기술을 정리하



(그림 2) Modernization 도구의 개발 전개 과정

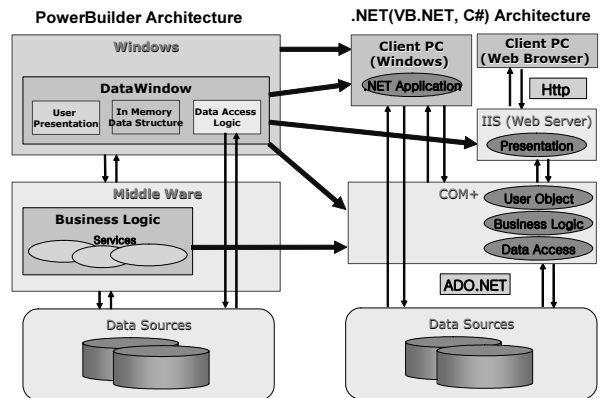
- ① Legacy 소스 코드(Source Code) : 새로운 환경(Windows/Web)으로 변환할 대상 프로그램, 즉 기존의 소스코드와 데이터베이스 내용을 일컫는다.
- ② 파싱(Parsing) : Legacy 소스 코드를 분해하여 데이터, 프로세스 부분을 분리하여 해당 내용을 저장소(Repository)에 저장한다.
- ③ 비즈니스 규칙(Business Logic) : 데이터, 사용자 인터페이스(User Interface), 동사(Verb) 부분으로 분류한다. 즉, 방법, 데이터, 컴포넌트, 인터페이스 분류를 자동으로 하며, Legacy 코드를 Web 환경의 시스템(Modernization)으로 변경시 분석 정보로 활용한다.
- ④ 컴포넌트 저장소(Component Repository) : 표적 언어에 대하여 각각의 컴포넌트를 작성하여 저장소에 저장하고, 표준화에 따른 프로그램-패턴으로 자동 변환 및 비즈니스 로직을 조립하여 프로그램을 자동으로 작성한다.
- ⑤ 지식 광산(Knowledge Mining) : 프로그램 소스코드를 파싱하여 문법 저장소와 비교하여 표적 언어의 형태로 변환을 하며, 중요한 로직 부분은 집중적으로 자료를 분해하여 변환할 수 있는 분석 정보로 사용한다.
- ⑥ 프로그램 변환(Program Conversion) : 변환에는 2가지 방법을 사용하고 있는데 단순히 변환하는 부분과 표적 프로그램의 특징에 맞게 재구조화하여 프로그램을 변환하는 방법을 지원하고 있다.
- ⑦ 프로토타이핑(Prototyping) : 마법사 기능을 지원하여 사용자가 쉽게 프로그램을 변환하는 기능을 제공한다.
- ⑧ 프로그램 향상(Program Enhancement) : CASE tool의 화면 편집기능을 지원한다. 즉, 동화상에서 Web에서 지원하는 기능을 지원한다. 즉, 동화상 등 테마에 따라 기능을 지원하며 다양한 기능을 제공한다.

모형-1. 비주얼언어(PB)에서 Java플랫폼으로의 변환 절차



(그림 3) 비주얼 언어에서 Java 언어 전환도

모형-2. 비주얼언어(PB)에서 .NET플랫폼으로의 변환 절차



(그림 4) 비주얼언어에서 .NET언어 전환도

면 다음과 같다.

- 정보저장 형태 : 메타데이터, 컴포넌트저장소, 프로그램 문법 데이터베이스 저장소
- 분리기술 : 비즈니스 로직 추출, LM 전환 방법, 패턴 컴포넌트 생성

3.1.1 어플리케이션 프로그램 마이닝 기법

핵심적인 어플리케이션 분석은 마이닝 기법을 사용하여, 진보된 소스코드를 분석하는 데 기반을 두었다. 소스코드 분석의 전체과정은 아래와 같다.

- 소스코드를 라인단위로 프로그램에서 읽어들인다.
- 라인단위로 읽은 문장을 하나의 워드 단위로 분리작업을 실시한다.
- 분리된 토큰에 대한 의미 분석작업을 실시한다.이 때 사전에 등록된 각종 정보 데이터베이스를 이용하여 토큰에 대한 정확한 성분을 파악한다
- 다음은 분리된 토큰과 토큰에 대한 의미가 파악이 되면 해당 프로그램문법 데이터베이스를 찾게 된다.
- 해당 데이터베이스에서 찾으면 다음 단계로 넘어가고, 해당 데이터베이스에서 찾지 못하면 문법오류리스트를 내보낸다.

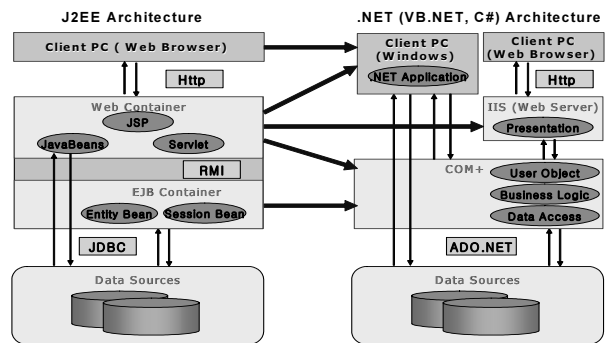
프로그램을 분석하는 과정은 크게 Lexical Parsing과 Syntactic Parsing 2가지로 나누어 진다.

Lexical Parsing은 프로그램소스코드의 한문장을 읽어와서 어휘별로 분리작업을 하게 되는 데 이를 토큰 열이라 부른다. 이를 SmartParser를 통하여 구현하였다.

Syntax란, 사전적인 의미로는 ‘문법 혹은 구문’의미로 사용되는 데 Syntactic Parsing은 실행 파일을 만드는 일반적인 컴파일러와는 달리 추구하는 대상이 달라 문법 DB와 Mapping하여 기능을 활용하였다.

어플리케이션 분석은 사용자의 어플리케이션에 대하여 개발과 유지보수라는 관점에서 접근하며, 가장 핵심적인 활동에 자동화 기법을 사용하였다.

모형-3. Java플랫폼에서 .NET플랫폼으로의 변환 절차



(그림 5) Java플랫폼에서 .NET언어 전환도

3.1.2 메타데이터(Metadata)

메타데이터는 정보자원을 관리하거나 쉽게 검색이 가능하도록 만든 것이며, 내용을 설명하고, 내용정보를 구성하는 것을 말한다. 사전적 의미의 메타데이터는 자원을 설명하는데 일정한 형태를 가지고 사용하는 경우를 의미하기도 한다[8].

3.1.3 비즈니스 로직 추출 기술

비즈니스 로직추출은 비즈니스 유스케이스 모델링과 객체 모델링을 통해 이해 당사자들의 요구사항을 명확히 파악하여 비즈니스로직을 추출한다. 비즈니스 로직 추출기능은 비즈니스와 관련된 코드를 빠르고 정확하게 기능을 추출하는 것이며 레거시시스템의 재작성(Re-documentation)을 통해 비용을 대폭 절감하여 준다.

3.1.4 비즈니스 로직 변환 기술

비즈니스 변환은 어플리케이션 분석 및 후보 컴포넌트 추출과정을 통해 추출된 후보컴포넌트의 비즈니스 로직을 변환한다. 비즈니스로직의 변환은 AS-IS 프로그램의 언어 및 TO-BE프로그램의 언어의 문법 DB의 Mapping을 통하여, 비즈니스로직 자동변환을 수행한다.

3.2 LM 모형

프로그램언어의 특성에 따라 아키텍처 구조가 바뀌면서 변환되는 형태가 있다. 물론 프로그램언어에 따라 각각의 특성에 따라 변환하는 내용이 서로 다르겠지만, 프로그램언어의 구조상 텍스트기반 프로그램언어, 비주얼기반의 프로그램언어 그리고 웹기반의 프로그램언어로 크게 3가지 유형으로 구분되어 진다. 본 논문에서는 3가지 유형의 프로그램언어를 기반으로 4가지 변환 모형을 기술하였다.

Client/Server 프로그램언어의 대표격인 PowerBuilder 프로그램언어를 웹 환경에서 운영되는 Java플랫폼으로의 전환하는 과정을 보여준다. powerBuilder 언어의 특징인 DataWindow의 기능에서 데이터처리 부분을 웹 브라우저 혹은 웹 컨테이너 부분으로 전환하는 것을 볼 수 있다. 비즈니스 로직부분은 EJB의 Entity Bean이나 Session Bean으로 전환되도록 하고 있다. 4세대언어인 Visual basic, Delphi, SQLWindows 라는 언어는 Power Builder와는 약간의 다른 구조를 취하고 있다. 즉, DataWindow가 아니라 일반적인 어플리케이션의 3계층의 구조를 취하고 있다. 전환의 방법은 앞과 동일하게 처리된다.

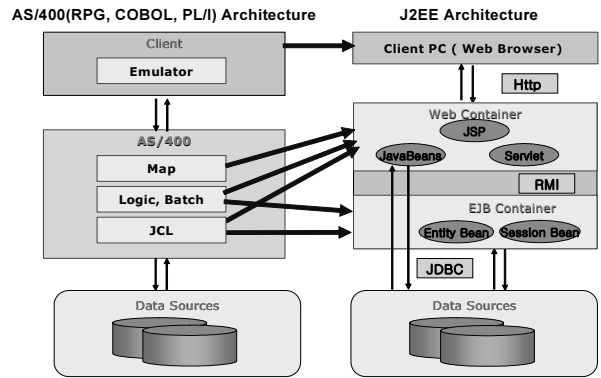
(그림 4)는 java플랫폼과 동일하게 처리되는데 Java플랫폼뿐만 아니라 Microsoft사의 .NET플랫폼의 COM+로 전환 작업의 과정을 표시한 것이다.

웹기반 언어의 전환이라 비교적 기술이 난해하지 않고 구조가 무척이나 단순하여 플랫폼의 구조만 이해하여도 나름대로 쉽게 접근할 수 있는 장점을 가지고 있다.

다음은 전형적인 레거시시스템에서 Java 혹은 .NET플랫폼으로의 마이그레이션 부분과 순수한 Java플랫폼(클라이언트/서버 구조)으로 마이그레이션하는 방법을 나타낸다.

(그림 6)은 IBM의 레거시시스템의 하나인 미들레인지의 컴퓨터 시스템의 하나인 AS/400기종에서 운영되는 COBOL, RPG, PL/I 등의 레거시시스템의 프로그램 언어를 Java플랫폼으로 마이그레이션 하는 과정이다. 전형적으로 화면, 로직, 배

모형-4. COBOL플랫폼에서 Java플랫폼으로의 변환 절차



(그림 6) AS/400 플랫폼에서 Java 플랫폼 전환도

치, 출력 그리고 JCL(명령어) 등의 레이어를 컴포넌트 기반의 Java기반으로 대칭하여 마이그레이션하는 것을 나타낸 것이다.

3.3 LM 지원도구의 구현

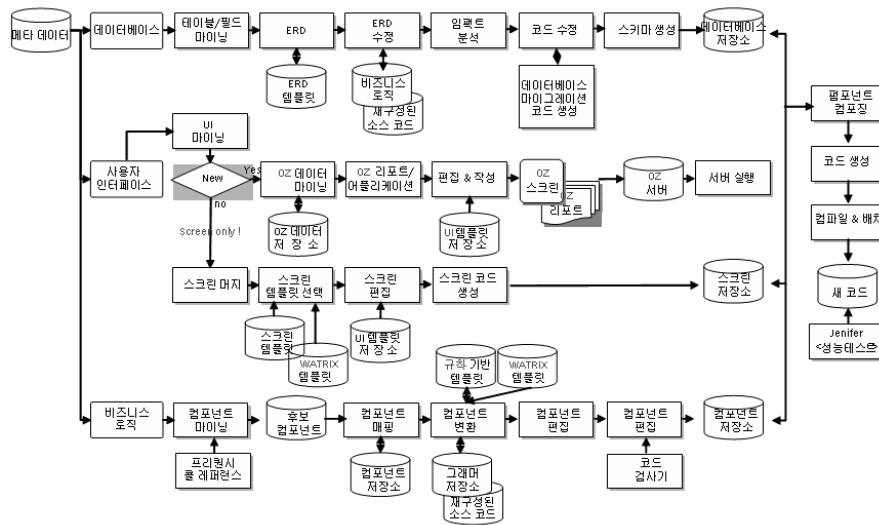
3.3.1 LM 지원도구 구조

본 연구는 앞장에서 기술된 지원도구를 정리하였으며 이를 총괄적으로 기능을 포함하고 있는 LM 지원도구를 구현하였다.

(그림 7)은 LM하기 위한 지원도구의 전체적인 기능의 흐름도를 정리한 그림이다. 다음은 단계별로 수행하는 도구의 특징적 기능을 나타낸 것이다.

3.3.1.1 역공학단계 - 프로그램 분석

(그림 8)은 레거시시스템의 어플리케이션 카타로그 정보를 수집하여 도구의 저장소에 저장을 하여 이를 파싱 작업을 거쳐 인벤토리저장소에 정보를 저장하게 된다. 저장된 정보를 참조하여 프로그램에 관련된 인벤토리문서를 자동으로 출력하게 된다. 출력된 문서에서 가장 중요한 누락 데이터, 사용



(그림 7) LM 지원도구 전체구조

하지 않은 데이터 등을 분류하여 레거시시스템의 정보를 보완하는 작업을 거친다. 다음 단계로 인벤토리저장소로 정보를 넘기는데, 이때 메타데이터 형태로 제공됨으로 사용자가 원하는 형태로 직접 Query가 가능하도록 설계하였다.

3.3.1.2 역공학단계 - 어플리케이션 분석

프로그램은 소스코드를 포함하여 프리젠테이션 부분, 데이터 접근부분으로 구성이 되어 있는데 이 정보를 통하여 역공학 파싱 작업을 통해 어플리케이션 정보를 추출하는 과정을 거치게 된다. 순공학 접근으로는 업무분석 - 데이터베이스 설계 - 프로그램 작성 순으로 이어지지만, 역공학 접근은 프로그램 정보에서 설계정보와 업무분석 정보를 추출하는 과정을 거치게 되는데 (그림 9)의 프로세스를 통하여 정보를 얻게 된다.

3.3.1.3 역공학단계 - 데이터베이스 분석

(그림 10)은 레거시시스템에서 데이터정보를 추출하여 이를 기반으로 ERD 정보를 추출하는 과정을 나타낸 것이다. ERD를 통하여 자동으로 생성된 관계형 데이터베이스 스키마정보는 기본의 프로그램과 연결이 가능하다.

3.3.1.4 역공학단계 - 프로그램 재구조화

(그림 11)은 프로그램 재구조화 작업을 통해 레거시시스템의 비구조화된 소스코드를 체계화 작업을 실행하게 되는

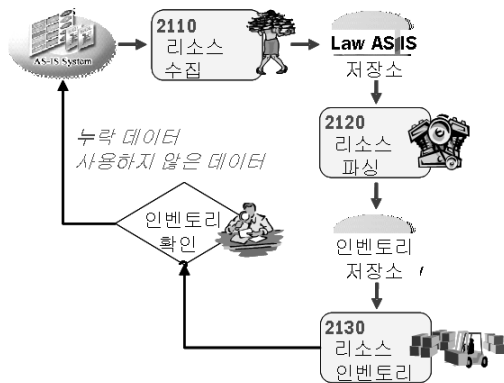
데, 불필요한 코드 제거-코드 정제-코드 비교 작업-재구조화 정보제공 작업을 거친다.

3.3.1.5 역공학단계 - 후보컴포넌트 마이닝

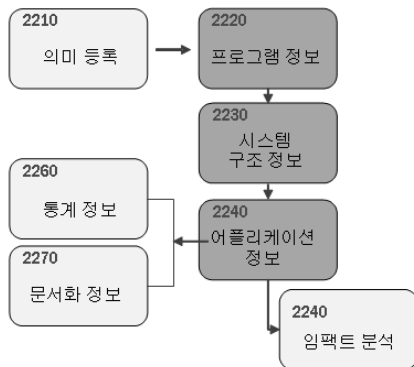
(그림 12)는 구조화된 레거시시스템에서 재사용이 가능한 컴포넌트를 추출하게 되는데 데이터베이스 정보는 데이터 추출을 하여 ERD과정으로 거쳐 자동으로 생성된 스키마정보를 활용하고, 프리젠테이션을 템플릿과 연결하여 재사용하는 과정을 거치고, 비즈니스로직은 이를 자동으로 변환하는 과정을 거치게 된다. 여기서 추출된 후보-컴포넌트를 기존에 사용하고 있는 컴포넌트저장소와 매핑하여 이를 사용하는 최종적인 절차를 거친다.

3.3.1.6 컴포넌트화 단계 - 데이터베이스 마이그레이션

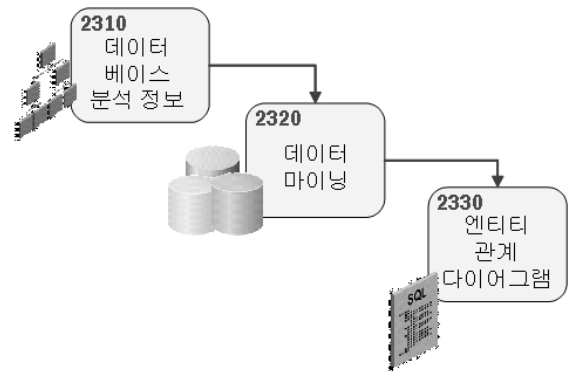
(그림 13)은 레거시시스템에서 관계형 데이터베이스로 구조 변경은 물론 실제 데이터를 마이그레이션 하는 프로세스를 나타내는 절차이다. 개괄 흐름은 레거시시스템의 구조(혹은 레이아웃)를 추출하여 이를 ERD로 바꾸어 보완한 후에



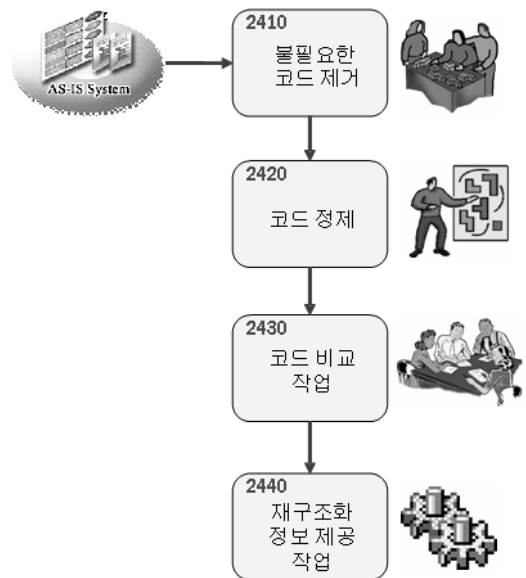
(그림 8) 역공학단계 - 프로그램분석



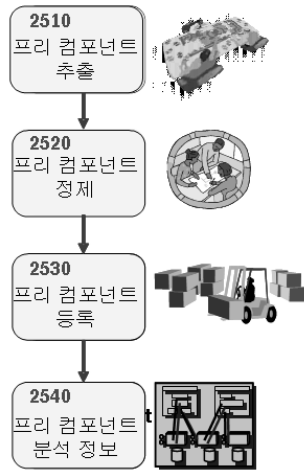
(그림 9) 역공학단계 - 어플리케이션 분석



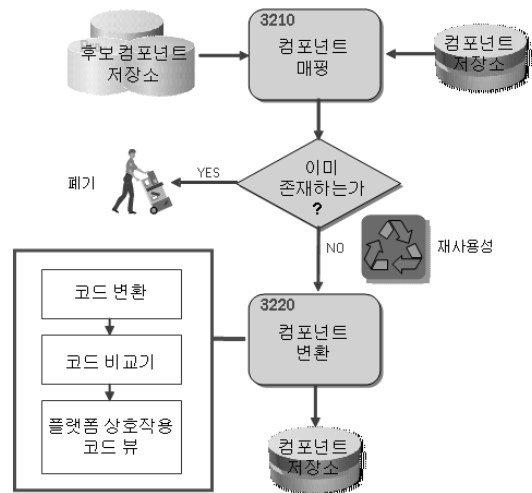
(그림 10) 역공학단계 - 데이터 분석



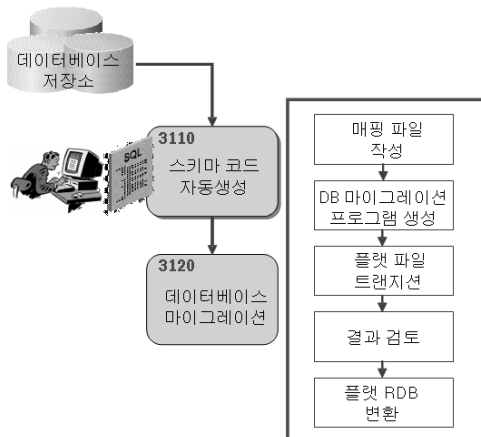
(그림 11) 역공학단계 - 프로그램 재구조화



(그림 12) 역공학단계 - 후보컴포넌트 마이닝



(그림 14) 컴포넌트화 단계 - 비즈니스 로직 변환

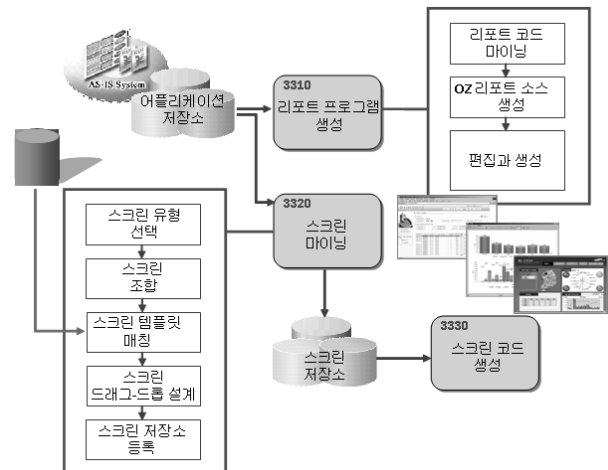


(그림 13) 컴포넌트화 단계 - 데이터베이스 마이그레이션

자동으로 스키마를 작성한다. 그림 오른쪽 박스 내용은 신규시스템의 구조를 파악하여 구시스템의 데이터를 신시스템의 데이터베이스로 자료를 이관하는 절차를 표현하였다. 신규시스템의 데이터 구조를 서로 연결하는 매핑절차와 이를 자동으로 전환하는 프로그램(Java나 C#)을 자동으로 작성하며, 기존의 다양한 형태의 파일 구조를 가장 평이한 Flat 파일 구조로 변환을 하여준다. 그리고 전환 프로그램을 작동하여 구시스템의 데이터를 신시스템의 데이터베이스 구조로 전환을 한다.

3.3.1.7 컴포넌트화 단계 - 비즈니스로직변환

(그림 14)는 프로그램을 세 가지 형태로 분리중의 하나인 비즈니스 로직 부분을 자동으로 변환하는 절차이다. 변환 대상인 후보 컴포넌트 단위로 쪼개진 조각을 기존의 컴포넌트 저장소와 비교하여 그런 내용이 이미 있는지를 파악하고 없다면, 재사용이 가능한 형태로 하기위해 코드를 변환한다. 변환 내용도 Code conversion - Code Comparator - Platform Interaction code view 과정을 거쳐 새로운 컴포넌트로 컴포넌트저장소에 저장되어 사용자에게 의하여 재사용된다.



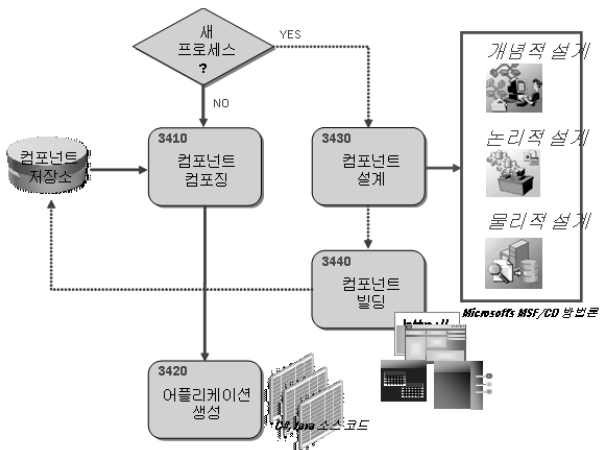
(그림 15) 컴포넌트화 단계 - UI 마이그레이션

3.3.1.8 컴포넌트화 단계 - 사용자 인터페이스 마이그레이션

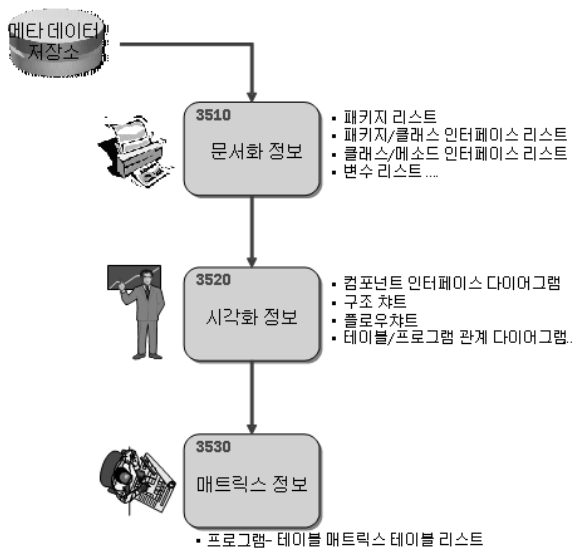
(그림 15)는 두번째 프로그램 조각인 사용자인터페이스(User Interface, 이하 UI) 부분을 마이그레이션하는 과정이다. 우선 출력 프로그램을 추출하여 기존의 상용화된 레포트도구와 연결이 가능하도록 정보를 제공한다. 다음으로 UI부분인 화면 처리 부분을 추출하여 조각을 편집하여 가장 근접한 UI템플릿저장소에서 템플릿과 연결하여 새로운 템플릿으로 정보를 자동 혹은 수동으로 이동작업을 실시한다. 참고로 템플릿저장소는 사용자가 미리 만들거나 기본적으로 도구에서 제공하는 형태로 구성되어 있다.

3.3.1.9 컴포넌트화 단계 - 프로그램(혹은 컴포넌트) 신규개발

(그림 16)은 컴포넌트(혹은 프로그램)를 새롭게 제작이 필요할 때 처리되는 절차이다. 언제나 레거시시스템의 컴포넌트만을 재사용할 수 없는 경우가 많다. 유지보수를 하는 동안 많은 업무의 변경으로 인하여 새로운 기능이 필요할 때 이번 절차를 따르면 된다. 우선 컴포넌트 개발방법론(마르미-III, MSF/CD등)을 활용하여 개발한 Aurora.J2EE,



(그림 16) 컴포넌트화 단계 - 프로그램 신규개발



(그림 17) 컴포넌트화 단계 - To-Be 어플리케이션 분석

Aurora.net 도구를 활용하여 컴포넌트를 설계하고 작성하게 된다. 작성된 컴포넌트를 컴포넌트저장소에 저장하고 이를 프로그램 개발시에 재사용한다.

3.3.1.10 컴포넌트화 단계 - To-Be 어플리케이션 분석

(그림 17)은 레거시시스템을 분석하고, 변환하고 새롭게 구

조를 변경하는 모든 마이그레이션 절차를 완료한 후에 작성된 목표프로그램 환경(예를들면, Java, .NET 등)을 자동으로 분석하는 절차이며, 레거시시스템이 어떻게 Modernization 시스템으로 변환하였는지를 확인하는 절차이다.

4. LM 적용사례 기반의 도구평가

본 논문에서 개발한 지원도구의 적용사례를 중심으로 정량적인 평가와 정성적인 효과로 구분하여 기술하였다.

4.1 정량적인 평가

정량적 효과는 우선 레거시시스템을 다양하게 적용한 표준템플릿의 효과와 세 가지 업종에 따른 유형별로 효과 부분을 나누어 평가하였다. <표 2>는 표준-기대효과 부분의 결과이다. 적용프로그램을 세부적으로 살펴보면 배치처리 프로그램, 데이터처리 프로그램, 정보를 조회하는 프로그램, 출력 프로그램 그리고 일반적인 관리 프로그램으로 구성되어 있는데, 본 논문에서는 프로그램 1,000본을 기본으로 하고 구체적인 자료를 수작업과 대비하여 효과를 산정하는 방식을 취하였다. 결과적으로 57%의 기대효과를 얻는 결과를 산출하였다. 다음은 세 가지 업종 형태의 업체를 선정하여 적용한 결과이다.

정부기관, 제1, 2금융기관 등 대표적인 업체를 선정하여 이를 LM을 적용한 결과이다. 이 적용사례에서 시사하는 바는 여러 가지로서 우선 LM지원도구를 적용한 결과와 그렇지 않은 결과가 업체마다 다르다는 점이다. 여기에는 레거시시스템의 표준화문제 등이 변수로 작용이 되었거나, 언어가 갖는 다양성과 특징에 따라 차이가 있음을 볼 수 있다. 즉, 정량적 효과는 LM개발 생산성 2.3배, 평균 생명주기 연장효과 110% 그리고 레거시시스템 분석 공수의 10% 감소, 개발기간의 20% 감소, 유지보수 비용을 30%이상 절감효과를 볼 수 있으며, 기술습득 기간도 기존의 수작업 대비하여 10% 이상 단축되는 것으로 나타났다.

4.2 정성적인 효과

국방부와 국내 통신회사 K사, 일본 전자회사 N사, 미국 캘리포니아 주 S사를 비롯한 50여개 기관에 LM 지원도구를 적용한 결과 다음과 같은 기대효과를 얻을 수 있었다.

<표 2> LM 적용시 작업구분별 생산성 비교

| 직업구분별 생산성 비교 | | | 신규개발시 | | 리모델링 적용시 | | 효과 |
|--------------|------|-----------------------|-------|-------------|----------|------------|-------|
| 직업구분 | 비율 | 처리내용 | | | | | |
| 처리(Batch) | 5% | 표준 배치 작업 | 1본/4일 | 50본/200일 | 1본/4일 | 50본/200일 | 0% |
| 등록(Entity) | 25% | 유지보수, 관리화면, 참조(3Form) | 1본/4일 | 250본/1000일 | 1본/2일 | 250본/500일 | 50% ▽ |
| 조회(Query) | 40% | 단순한 조회(Without Logic) | 1본/2일 | 400본/800일 | 3본/1일 | 400본/133일 | 83% ▽ |
| 출력 Form | 3% | | 2본/3일 | 30본/45일 | 1본/1일 | 30본/30일 | 33% ▽ |
| 출력 일반 | 27% | | 1본/1일 | 270본/270일 | 2본/1일 | 270본/135일 | 50% ▽ |
| 총 | 100% | | | 1000본/2320일 | | 1000본/998일 | 57% ▽ |

〈표 3〉 LM의 적용사례 결과

| 사례 | 분류(분) | 구분 | 전환율(%) | 기간(개월) | 소요공수(M/M) | 비용(백분율) |
|--------|-------|------|--------|-----------|-------------|---------|
| 공공기관 | 627 | 비적용 | - | 7 | 30 | 100 |
| | | 툴 적용 | 77.5 | 2 | 14 | 46.6 |
| | | 효과 | - | 5(-71.5%) | 16(-53.4%) | -53.4% |
| 제1금융기관 | 3150 | 비적용 | - | 18 | 260 | 100 |
| | | 툴 적용 | 92.3 | 8 | 180 | 69.2 |
| | | 효과 | - | 10(-60%) | 207(-30.8%) | -30.8% |
| 제2금융기관 | 1106 | 비적용 | - | 10 | 179 | 100 |
| | | 툴 적용 | 85.6 | 7 | 95 | 53.1 |
| | | 효과 | - | 3(-30%) | 84(-46.9%) | -46.9% |

정성적 효과는 레거시시스템을 재사용 한다는 것으로 즉 기존의 전산 인력의 활용으로 인력교체의 불안감을 해소하고, 기존 업무 프로세스의 승계로 업무 프로세스 변경의 불안감을 해소하며, 유지보수 비용 절감으로 수익성의 증대효과를 볼수 있다.

5. 결 론

본 논문에서는 LM 구축을 위한 지원도구에 관한 것으로 다음과 같은 연구 결과를 보이고 있다. 레거시시스템 분석경험, 소프트웨어 재사용에 관한 연구들을 분석하고 Web 환경 시스템 구축 개발을 위한 지원도구를 설계하고 개발하였다.

- 레거시시스템을 재사용하기 위한 지원도구를 설계하고 개발하였다.
- 개발된 LM 지원도구를 활용하여 레거시시스템을 분석하고, 후보컴포넌트를 추출하여 정제한 다음 저장하여 재사용하는 과정을 구현하였다.

지원도구는 역공학단계와 컴포넌트화 단계로 구성되어 있습니다. 역공학 단계는 프로그램분석, 어플리케이션 분석, 데이터베이스 분석, 프로그램 재구조화, 후보컴포넌트 마이닝으로 구성되어며, 컴포넌트화단계는 데이터베이스 마이그레이션, 비즈니스로직변환, 사용자인터페이스 마이그레이션, 신규개발, TO-BE분석을 포함하고 있습니다.

이러한 연구를 통하여 도출된 결과를 통하여 LM에 관한 향후 연구과제를 기술하면 다음과 같다.

첫째, 레거시시스템에서 재사용 컴포넌트 정보를 추출하는 과정은 비정형자료 추출에 해당하여 수 많은 변수가 작용하게 된다. 이러한 점으로 처리하는 프로세스 또한 변화에 대응하지 않으면 안 될 것이다. 마르미-RE에서 RE-Method를 거치면서 지속적으로 계승 발전하여 품질 좋은 컴포넌트를 추출할 수 있으며 절차의 최적화를 통해 개발기간 단축의 효과를 얻을 수 있다고 본다.

둘째, 개발지원도구의 자동화 비율을 극대화 하는 전략이

필요하다. 개발방법론과 동일한 프로세스에 맞추어 자동화 비율을 높이는 지원도구가 되어야 할 것이다.

셋째, 레거시시스템에 대한 지적재산의 인지도를 높여가는 전략이 필요하다. 사장되어가는 레거시시스템을 지속적으로 유지보수의 품질을 향상 시켜 레거시시스템의 지적자산을 보호해야 한다.

참 고 문 헌

- [1] Chuck Martin, "e-비즈니스.com," 21세기북스, 1999.
- [2] 김진우, HCL Lab, "인터넷 비즈니스," 영진출판사, 1999.
- [3] 박병형, "밀레니엄 버그," 영진출판사, 1998.
- [4] 박병형 저, 블록놀이와 CBD Component Based Development, 태영출판사, pp.14-17, 2002.
- [5] 박병형 저, "한권으로 끝내는 e-ERP," 태영출판사, pp.25-27, 2001.
- [6] E. J. Chikofsky and J. H. C. II, Reverse Engineering and design recovery; A taxonomy. IEEE Software, 7(1):13-17 1990.
- [7] M. Fowler. Refactoring: Improve the design of existing code. Addison-Weslet, Reading MA, 1997.
- [8] Understanding Metadata National Information Standards Organization ISBN:1-880124-62-9, 2004.
- [9] H. M. Sneed. Objectorientierte Softwaremigration. Addison - Wesley, German, 1998.
- [10] H. A. Muller, M. A. Orgun S. R. Tilley, and J. S. Uhl. A reverse engineering approach to subsystem structure identification. Journal of software Maintenance, 1993.
- [11] Frost & Sullivan, "Frost & Sullivan White Report," 1999.
- [12] 레거시시스템의 컴포넌트화 방법론 "마르미-RE," Magic and Robust methodology Integrated-ReEngineering(v1.0), 주관 기관: 한국전자통신연구원, 참여기관:(주)케미스, 2003.
- [13] 양해술, 박병형, "CBD 환경을 위한 4GL 어플리케이션의 웹어플리케이션으로의 변화," 한국정보처리학회지. 1226-9182, 제 10권3호, pp.89-96, 2003.
- [14] 최일우, 류성열, "레거시시스템 진화를 위한 효율적 재공학 프로

세스,” 한국정보처리학회논문지(D), Vol.10-D, No.5, pp.845-858, Aug., 2003.

[15] 박병형, 이승호, 김영희, “MSF/CD 방법론 기반의 분석/설계 도구설계에 관한 연구,” 한국정보처리학회지, 1226-9182, 제10권3호, pp.63-73, 2003.



양해승

e-mail : hsyang@kitt.co.kr
1993년 순천향대학교 수학과(학사)
1993년 ~ 1995년 한국SGS IST(전산실)
프로그램
1995년 ~ 2000년 SK네트웍스 시스템사업팀
재직

2005년 호서대학교 벤처전문대학원 컴퓨터응용기술학과(공학석사)
2000년 ~ 현 재 한국IT진흥(주) 엔터프라이즈사업본부 영업본부장
/이사
2005년 ~ 현 재 호서대학교 벤처전문대학원 컴퓨터응용기술학과
박사과정
관심분야: S/W개발방법론, 프로젝트관리 및 품질관리방법론,
IT기술 컨설팅



박병형

e-mail : pph@camis.co.kr
1988년 서울신학대학교 신학과(학사)
1994년 단국대학교 경영대학원(MBA)
2000년 연세대학교 산업대학원(공학석사)
2006년 호서대학교 벤처전문대학원(공학박사)
1975년 ~ 1980년 삼성그룹 종합 전산실

(신세계백화점) 대리

1980년 ~ 1981년 숭실대학교 전산실 연구원
1982년 ~ 1989년 (주)유공(현재 SK) 정보시스템부 전문직 과장
1989년 ~ 현 재 (주)케미스 대표이사/CEO, 미국현지법인
NexxIT(회장)

관심분야: LM, CBD, ERP, CASE, S/W개발방법론



양해술

e-mail: hsyang@office.hoseo.ac.kr
1975년 홍익대학교 전기공학과(학사)
1878년 성균관대학교 정보처리학과(석사)
1991년 日本 오사카대학 정보공학과 S/W
공학전공(공학박사)

1975년 ~ 1979년 육군중앙경리단 전자계산실
시스템분석장교

1980년 ~ 1995년 강원대학교 전자계산학과 교수
1986년 ~ 1987년 日本 오사카대학교 객원연구원
1995년 ~ 2002년 한국S/W품질연구소 소장
1999년 ~ 현 재 호서대학교 벤처전문대학원 교수
관심분야: 소프트웨어공학(특히, S/W 품질보증과 품질평가,
품질감리 및 컨설팅, OOA/OOD/OOP, SI), S/W
프로젝트관리, 컴포넌트 개발방법론과 품질평가