

# 레거시 시스템 진화를 위한 효율적 재공학 프로세스

최 일 우<sup>†</sup> · 류 성 열<sup>††</sup>

## 요 약

1980년대 이후, 소프트웨어 위기에 대응하기 위한 다양한 소프트웨어 공학적 기법들이 출현하기 시작 하였고, 현재 소프트웨어의 질적 향상 및 생산성 향상을 꾀하기 위한 개발 지침으로 소프트웨어 개발 프로세스를 적용하는데 관심이 집중되어지고 있다. 그러나 대부분의 방법론들이 앞으로 구축할 새로운 시스템의 재사용성 확보에 치중할 뿐, 기존의 레거시 시스템 자원을 재사용하여 새로운 시스템을 구축하는 재공학 방법은 지원하지 못하고 있는 실정이다. 본 논문에서는 레거시 시스템의 효율적 진화를 위한 재공학 프로세스(Reengineering Process)로 RUP를 확장, 커스터마이징하여 구성한 RUP+re를 제시한다. RUP+re는 Small-h 모델을 바탕으로 크게 역공학 워크플로우(Reverse Engineering Workflow), 변환 워크플로우(Transformation Workflow), 진화 워크플로우(Evolution Workflow)로 구성되어진다. RUP+re의 각 워크플로우와 세부 스텝을 제공하고 재공학 사례 연구를 통하여 실질적으로 프로세스를 검증함으로써 레거시 시스템 자원의 진화를 위한 소프트웨어 재공학 프로세스 구축에 효율적인 지침을 제공한다.

## An Efficient Reengineering Process for Legacy System Evolution

Il Woo Choi<sup>†</sup> · Sung Yul Rhew<sup>††</sup>

### ABSTRACT

The various software engineering techniques have been come out in order to cope with the software crisis since 1980's. These days the software Engineering has focused on the process of software development which is the guide for a quality and productive improvement of software. But, most of the methodologies assume that a new system will be constructed and reused in the future. these do not support how we reuse legacy system's resources and construct a new system efficiently. In this paper, we present an efficient Reengineering Process for legacy system evolutions, RUP+re, which extends and customizes RUP. RUP+re consists of Reverse Engineering Workflow, Transformation Workflow and Evolution Workflow based on Small-h model. We describe RUP+re's workflows with their own detail steps and verify the process through the case study practically. So, we provide efficient guidelines to the software Reengineering process construction which evolves the resources of a legacy system.

**키워드 :** 소프트웨어 개발 방법론(S/W Development Methodology), 재사용(Reuse), 진화(Evolution), 재공학(Reengineering), 역공학(Reverse Engineering), 프로세스 커스터마이징(Process Customizing), RUP

### 1. 서 론

빠르게 변화하는 환경을 만족시키기 위해 정보시스템은 점점 복잡해지고, 서비스 공급자나 개발자들이 사용자 요구에 맞는 시스템을 개발, 공급하기가 어려워지게 되었다. 이런 소프트웨어 개발의 한계성을 극복하기 위해 새롭게 대두된 것이 소프트웨어 공학(S/W Engineering) 및 소프트웨어 개발 방법론(S/W Development Methodology)이다. 현재 소프트웨어 산업계에서는 새로운 응용 시스템 개발에 집중을 하면서 품질향상, 개발기간 단축 등 사용자의 욕구 충족을 위한 효율적인 소프트웨어 개발 프로세스(S/W Development Process)를 만드는 데 치중하고 있다. 특히 소프트

웨어 재사용을 통하여 생산성을 최대화 하는 컴포넌트 기반 소프트웨어 공학(Component Based S/W Engineering) 및 컴포넌트 개발 방법론(Component Development Methodology)에 관심이 모아지고 있다.

대표적인 소프트웨어 개발 프로세스인 Rational Unified Process(RUP)를 비롯하여 컴포넌트 개발 중심의 Catalysis, UML Components 등 여러 방법론들이 있으나 새로운 시스템의 변경 가능성 및 재사용성 확보에만 치중할 뿐, 기존 시스템을 재사용하는 재공학은 크게 고려하지 못하고 있다. 기업들은 새롭고 다양한 기술적 요구사항들을 시스템에 반영하기를 원하며 이를 위한 정확한 분석 작업을 필요로 한다. 그러나 이러한 새로운 기술적 요구 사항들을 시스템에 반영하기 위하여 기존의 자원을 모두 포기하고 완전히 새로운 시스템을 구축하는 것은 비용적으로 매우 어려운 일이다.

기 개발되어진 레가시 시스템(Legacy System)은 실질적

<sup>†</sup> 준 회원 : 숭실대학교 대학원 컴퓨터학과

<sup>††</sup> 종신회원 : 숭실대학교 컴퓨터학부 교수

논문접수 : 2003년 5월 20일, 심사완료 : 2003년 7월 4일

인 운용을 통하여 이미 검증 되었을 뿐만 아니라 유사 문제영역(Domain)의 시스템 개발에 필요한 문제영역지식(Domain Knowledge)이 이미 확보되어 있는 상태[1]이다. 그러나 유사 문제 영역의 시스템을 구축하기 위해 레거시 시스템 자원을 효율적으로 재사용 할 수 있는 실용적이며 체계적인 방법의 부재로 기존의 자원들을 재사용하지 못하고 시스템을 교체(Replacement)하고 있는 것이 현실[2]이다.

현재 제시되어진 몇몇 재공학 방법론들도 소프트웨어 라이프 사이클 전체를 수용하지 못하고 있으며, 실질적으로 레거시 시스템에서 재사용 가능한 재사용 단위를 추출하는 방법 및 재공학 가능성 판단 방법등을 명확히 제시하지 못하고 있다.

레거시 시스템 자원을 재사용하여 소프트웨어 개발 생산성을 향상 시키는 방법 중 하나는 효율적인 재공학(Reengineering)기법들을 소프트웨어 개발 프로세스의 일부분으로 통합, 지속적인 소프트웨어의 진화(Evolution)를 지원하는 것이다.

본 논문에서는 이러한 필요성을 충족하고 문제점을 해결하기 위한 방안으로 다양한 재공학 기법을 소프트웨어 개발 프로세스에 통합, 기 개발되어진 레거시 시스템의 자원을 이용, 어플리케이션 개발을 지원하는 효율적인 재공학 프로세스(Reengineering Process)를 제시한다. 또한 사례연구를 통하여 제시된 재공학 프로세스가 레거시 시스템 자원 재사용에 기반한 소프트웨어의 진화에 효과적으로 적용될 수 있음을 보였다.

## 2. 관련 연구

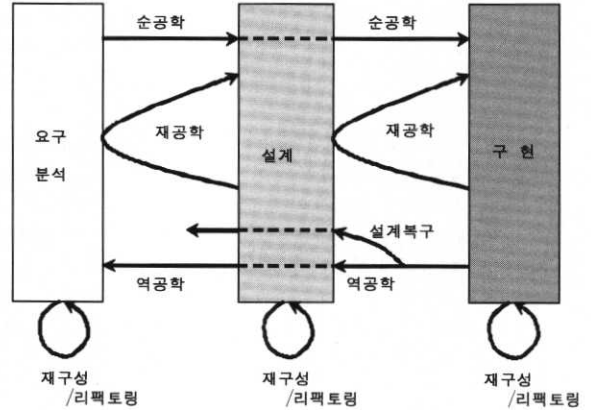
### 2.1 재공학(Reengineering)

소프트웨어 역공학(Reverse Engineering)이란, 자동화 도구(Case Tool), 수작업 등을 통하여 소프트웨어의 물리적 계층(Physical Layer)의 표현을 데이터 요소와 프로세스등을 설명해 주는 명세 계층(Specification Layer)의 표현으로 재구성하는 과정[1]을 말한다.

소프트웨어 재공학(Reengineering)이란, 자동화 도구, 수작업 등을 통하여 기존의 소프트웨어 시스템의 기능 및 디자인을 수정, 성능 향상 및 유지 보수를 피하는 과정[3]을 말한다.이 과정은 개발단계의 산출물이나 직접적인 소스의 수정을 통하여 이루어진다.

일반적으로 재공학 프로세스(Reengineering Process)는 역공학과 순공학(Forward Engineering)을 포함하고 있다. (그림 1)의 재공학 수행절차를 살펴보면 역공학, 분석, 재구성, 변환 등의 4단계를 거치게 되어 있다. 역공학과 분석 단계를 통하여 소프트웨어의 명세와 그들간의 관계를 재구축하기 위한 활동을 수행하고, 재구성(Restructuring)을 통하여 소프트웨어의 외부행위에는 변환을 미치지 않고 소프트웨어의 내부 기능이나 구조를 변경하고, 변환(Transition)단

계를 통하여 언어, 운영 체제, 아키텍처등이 상이한 환경으로 전환 하거나 기능 및 기술력을 향상하는 활동을 수행[4]한다.



(그림 1) 역공학, 순공학, 재공학간의 관계

일반적으로 재공학 목적에 따른 방법으로는 시스템의 버그 수정 및 외부적 행위의 변경 없이 개선을 수행하는 유지보수(Maintenance), 유지보수 보다 광범위한 변경을 수행하며 기존 시스템의 주요 부분을 재사용하여 새로운 시스템으로 개발해 나아가는 현대화(Modernization) 및 진화(Evolution), 레거시 시스템의 유지보수 및 진화가 비용적인 측면이나 기술적인 측면에 부적합하여 시스템을 새로 재개발하는 교체(Replacement)[4, 6]등이 있다. 또한 재사용 수준 및 목적에 따른 레거시 시스템의 재사용 기술도 화이트 박스(White Box), 블랙박스(Black Box), 랩핑(Wapping), 스크린 스크래핑(Screen Scraping), 미들웨어(Middleware), EAI (Enterprise Application Integration)[5, 6]등 현재 다양하게 연구되고 있다.

### 2.2 재공학 방법론

기존 레거시 시스템의 재사용을 지원하는 재공학 방법론으로는 Renaissance, Common Object based Reengineering Unified Model II (CORUM II), UIRich System Redevlopment Methodology(USRM), Mission Oriented Architecture Legacy Evolution(MORALE)등을 들 수 있다. 그러나 이러한 재공학 방법론들은 역공학을 통하여 기존 레거시 시스템 정보가 추출 가능하다는 전제하에 구성되어 있고 프로세스 각 단계의 액티비티와 산출물이 명확히 정의 되어 있지 않으며 소프트웨어 개발 라이프 사이클의 전 단계를 지원하지 못한다. 각 방법론들의 특징은 <표 1>과 같다.

일반적으로 재공학 프로세스의 재공학 전략은 크게 기술적 관점과 비즈니스적 관점에 따라 구분[6]한다.

기술적 관점은 레거시 시스템의 현재 상태와 관련된 산출물 정의, 재사용 단위 추출, 재사용 기술 등이다. 산출물을 역공학 및 재공학 하기 위해 적용하고자 하는 목표

<표 1> 재공학 방법론들의 특징

	Renaissance	CORUM II	USRM	MORALE
특징	<ul style="list-style-type: none"> <li>진화 계획, 구현, 인도, 분배의 4단계로 구성되어 있다.</li> <li>단계별 산출물 템플릿을 제공한다.</li> <li>재공학 기술별 기술 레포트를 제공한다[7].</li> </ul>	<ul style="list-style-type: none"> <li>아키텍처와 코드 기반의 재공학 도구들을 통합하기 위한 요구사항과 프레임워크를 제공하는 모델이다.</li> <li>코드 레벨, 코드구조 레벨, 기능 레벨, 개념 레벨, 아키텍처 레벨수준의 변환을 목표로 한다[8].</li> </ul>	<ul style="list-style-type: none"> <li>데이터, 기능 위주의 재공학 접근방식을 제공.</li> <li>계획, 분석, 포지셔닝, 변환의 4단계로 구성되어 있다[9].</li> </ul>	<ul style="list-style-type: none"> <li>기술적인 요소보다 조직의 미션(Mission)에 중심을 둔 방법.</li> <li>분석, 설계, 진화의 3단계로 구성.</li> <li>시스템의 구조적 변경에 대한 영향을 예측, 시스템 진화 초기에 변경에 대한 위험요소를 알아내고, 기존 시스템을 효과적으로 분석, 재사용 부품을 추출할 수 있도록 한다.</li> <li>UI를 중심으로 한 재공학 접근방식을 취하고 있다[10].</li> </ul>
	<ul style="list-style-type: none"> <li>재공학 프로세스내에 역공학을 포함하고 있기는 하나 구체적인 역공학 방법 및 지침이 없다.</li> <li>소프트웨어 개발 라이프사이클의 전단계를 포함하지 못한다</li> </ul>			

<표 2> 재공학의 문제점 및 요구사항

	재공학의 문제점	재공학의 요구사항
내용	<ul style="list-style-type: none"> <li>역공학을 통한 현재 시스템의 수준의 명확한 판단.</li> <li>재공학 수행이 최선책인지에 대한 판단.</li> <li>역공학 산출물을 근거로 재공학 목적에 따른 재공학 전략의 수립 및 다양한 수준의 재사용 기술의 적합한 선택.</li> <li>역공학 산출물에 직접적인 요구사항을 반영하는 재구성, 변환 단계의 수행 방법.</li> <li>재구성, 변환 단계에서 요구 사항이 레거시 시스템에 직접 반영되어도 요구 사항이 반영된 시스템과 또 다른 산출물간의 일관성 유지[3] 등이 문제점으로 제시된다.</li> </ul>	<ul style="list-style-type: none"> <li>재공학을 수행해야하는 범위 영역의 정의, 명확한 재공학 목적 및 목표 수립.</li> <li>레거시시스템의 양/질적인 충실한 문서화.</li> <li>실용적이고 실질적인 레거시 시스템의 재공학 지원.</li> <li>적합한 재공학 전략 및 기술의 선정.</li> <li>재공학의 대안 제시, 재공학 전략 및 기술의 검증.</li> <li>재공학을 위한 비용산출, 자원/스케줄/위험관리 지원.</li> <li>시스템 재공학을 통한 이득 예측.</li> <li>요구사항의 충실한 반영[3]등이 요구되어진다.</li> </ul>

수준(예, 소스 레벨, 디자인 레벨, 아키텍처 레벨, 프로세스 레벨 등) 및 재사용 기술(예, 화이트 박스, 블랙박스, 랩핑, 스크린 스크래핑, 미들웨어, EAI 등) 그리고 적용 가능한 자동화 도구(Case Tool)등이 주요 변수가 된다[3, 6].

비즈니스적 관점은 재공학을 수행할 경우의 비용과 그에 대한 효율성간의 상충관계(trade-offs) 및 기술적 타당성이 주요 변수가 된다.

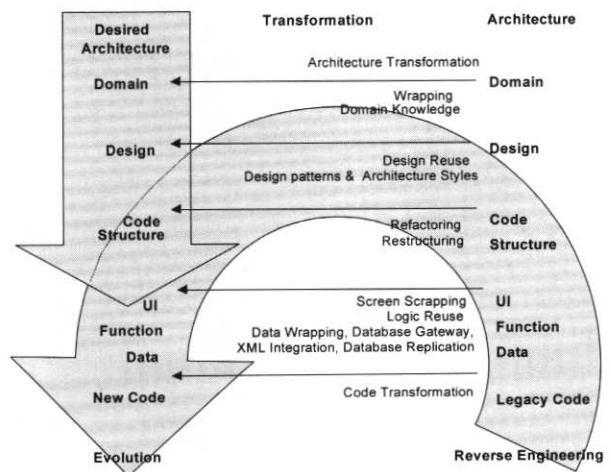
이와 같은 두 가지 관점은 레거시 시스템을 새로운 환경으로 변환시키는데 있어서 재공학을 통한 진화를 수행할 것인지, 유지보수 할 것인지, 새로운 시스템으로 교체할 것인지 등의 결정을 필요로 하며 효율적인 재공학 프로세스는 이러한 두 가지 관점의 판단을 효율적으로 지원하여야 한다[3, 6]. 일반적으로 재공학에 접근하기 어려운 이유와 재공학 방법론이 갖추어야할 요구사항은 <표 2>와 같다.

그러나 현재까지 연구되어진 재공학 기법들은 이러한 관점 고려하여 재공학 수행을 판단하게 할 수 있는 방법들이 부족하며 재공학 프로세스와 일반적인 소프트웨어 개발 프로세스내의 중복되는 유사 활동(Activity)에도 불구하고 각 프로세스별 단계간의 연계 방법은 정의되어 있지 않다.

### 3. 재공학 프로세스 RUP+re의 구성

본 논문에서 제시하는 재공학 프로세스 RUP+re는 다음(그림 2)와 같은 Small-h 모델 기반의 접근법을 가진다. Small-h 모델은 이미 연구되어진 재공학 모델들을 바탕으로 구성되었다. 소프트웨어를 추상화 수준에 따라 6계층으로 나누

고 각 계층의 산출물을 구성하는 역공학, 효율적인 재공학을 수행하기 위한 변환, 레거시 시스템 자원을 이용 새로운 시스템으로 발전시키는 진화의 3단계로 구성되어진다.



(그림 2) Small-h Model

재공학 프로세스의 각 단계를 지원하는 워크플로우에 대한 설명은 다음과 같다.

**첫째, 역공학 워크플로우(Reverse Engineering Workflow):** 역공학 워크플로우는 기존 레거시 시스템 코드 및 산출물로부터 역공학을 수행, 다양한 계층의 산출물을 정의, 추출하고 산출물간 일관성 검증을 수행 한다. 시스템에 따라 역공학 산출물의 추상화 수준은 가변적이며 다음 단계

워크플로우의 입력물을 제공하는데 그 목적이 있다.

**둘째, 진화 워크플로우(Evolution Workflow) :** 진화 워크플로우는 역공학, 변환 워크플로우를 개발 프로세스에 수용할 수 있도록 RUP[11]를 기반으로 확장, 재정의 하였다. 재사용 산출물을 이용하여 새로운 요구사항이 반영된 시스템을 개발하는 워크플로우이다. 새로운 요구사항이 반영된 시스템을 역공학, 변환 워크플로우를 통하여 정의, 연계된 재사용 산출물을 획득, 적용하여 재공학 목적에 알맞게 개발하는 효율적인 방법을 제시한다.

진화 워크플로우는 새로운 요구사항이 반영된 시스템의 분석, 설계를 레거시 시스템의 역공학과 병렬적으로 수행 가능하며 역공학을 통하여 추출된 산출물의 추상화 수준에 따라 그 진행 정도가 가변적이다. 적어도 역공학 산출물의 최상위 추상화 수준까지는 새로운 요구사항을 반영하여 시스템을 분석, 설계하여야 변환 워크플로우를 통하여 재사용 가능한 산출물을 수용할 수 있다.

**셋째, 변환 워크플로우(Transformation Workflow) :** 변환 워크플로우는 역공학 산출물의 수준 및 재공학 목적에 따라 재사용 가능 요소, 기법, 가능성을 효율적으로 판단, 정의하며 적용 가능한 기술적, 비즈니스적 역공학 전략 등의 판단을 수행한다.

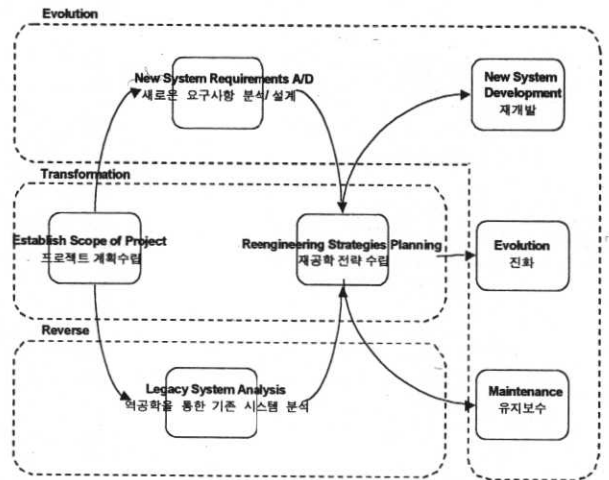
또한 Bottom Up 방식으로 수행되어진 역공학 워크플로우의 재사용 산출물을 진화 워크플로우의 새로운 요구사항이 반영된 Top Down 방식의 분석, 설계의 특정 단계(Phase)로 역공학 산출물 세트를 분기하는 효율적 방법을 제공한다. 또한 재공학이 불가능한 경우, 대체 가능한 프로세스로 연계 가능하다.

이 모델은 역공학의 수준이 상위레벨로 추상화 될수록 새로운 시스템의 요구사항 기반으로 분석, 설계 하여야 하는 진화워크플로우의 수행 단계가 최소화 되어지게 되며 추상화 수준이 낮을수록 새로운 요구사항을 기반으로 분석, 설계하여야 하는 부분이 많아지게 된다. 즉 명확한 역공학 산출물의 추출 및 추상화 수준이 변환 워크플로우를 통한 재사용 프로세스의 효율성을 좌우하게 된다.

본 논문에서 제시하는 재공학 프로세스 RUP+re은 RUP를 확장하여 Small-h 모델에서 제시하는 개념을 수용할 수 있도록 커스터마이징 하여 구성 하였다. 전체 프로세스 워크플로우 개념도는 (그림 3)과 같다.

RUP+re의 정의 및 구성은 다음과 같다.

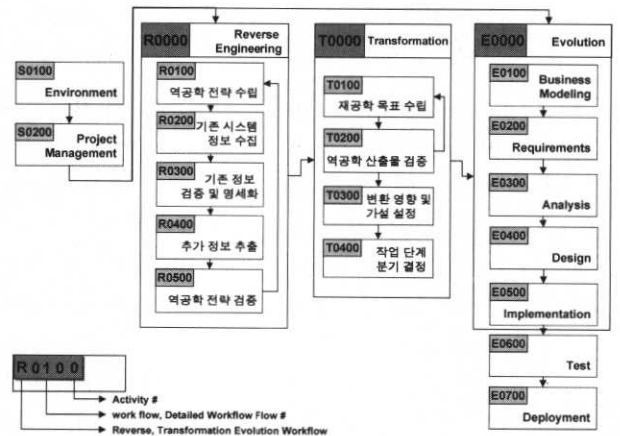
- RUP + re 정의 : Process = Phase(\*).
- Phase = Workflow(\*) + Iteration(\*)
- Workflow = Detailed Workflow(\*)
- Detailed Workflow = Activity(\*) + Sequence(\*)
- (\*) = 반복.



(그림 3) 재공학 프로세스 RUP+re의 워크플로우 개념도

(그림 4)는 본 논문에서 제시하는 RUP+re의 전체적인 프로세스 세부 구성도이다.

RUP+re 구성 :



(그림 4) RUP+re의 워크플로우 및 세부 워크플로우

RUP+re는 역공학 워크플로우를 통한 효율적인 역공학 산출물 추출 방법[12] 및 변환 워크플로우를 통한 재사용 산출물과 개발 프로세스의 효율적인 연계를 통하여 재공학 프로세스의 효율을 극대화 할 수 있다. 본 장에서는 RUP+re 워크플로우별 특징을 살펴보고 다음 장에서는 프로세스를 활용한 재공학 사례연구를 통하여 프로세스의 효율성 검증 및 문제점을 살펴본다.

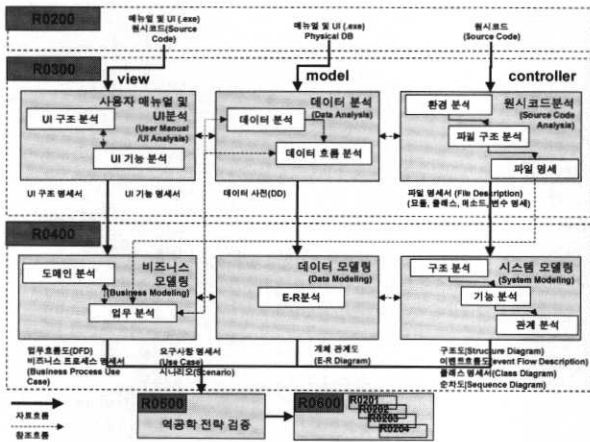
3.1 역공학 워크플로우(Reverse Engineering Workflow : R0000)

역공학 워크플로우(R0000)의 세부 워크플로우 및 목적은 <표 3>과 같다.

(그림 5)는 R0300과 R0400 단계의 액티비티 및 산출물간의 관계를 도식화 한 것이며 <표 3>을 통하여 액티비티와 산출물들의 설명을 제시하였다.

<표 3> 역공학 워크플로우의 세부 워크플로우 및 액티비티 설명

Detailed Workflow	Activities	내용 및 목적	산출물
R0100 기존 시스템 정보 수집	현 레거시 시스템의 일반적인 자원 및 정보 수집.		
	도메인 정보 수집 시스템 정보 수집 기초 산출물 수집 시스템 inventory 구축	<ul style="list-style-type: none"> <li>기존 레거시 시스템의 분석에 필요한 기본적인 도메인 정보 수집.</li> <li>기존 레거시 시스템의 시스템 아키텍처를 포함한 개발 정보 수집.</li> <li>기존 레거시 시스템의 실행코드, 개발 소스, 매뉴얼등의 개발 산출물 수집.</li> <li>각 단계의 결과물을 가지고 Inventory를 구축.</li> </ul>	입력물 : 산출물 : System Inventory
R0200 역공학 전략 수립	전체 프로젝트의 목적과 재공학의 목표에 부합하는 역공학 전략 수립.역공학 전략에 따라 추출되는 산출물의 종류 및 방법 정의.		
	역공학 전략 수립 역공학 가능성 조사 역공학 비용 예측	<ul style="list-style-type: none"> <li>프로젝트의 목적과 재공학의 목표에 부합하는 역공학 전략 및 목표 수립.</li> <li>전 단계의 산출물을 기반으로 적절한 기법, 지원 CASE 도구 등 가능성 조사.</li> <li>역공학 수준, 지원 인력등을 기반으로 역공학 비용 예측</li> </ul>	입력물 : 산출물 : Reverse Plan
R0300 기존 정보 검증 및 명세화	전단계를 통하여 수집된 산출물을 MVC 모델 기반으로 분류, 일차 분석을 통한 명세화.		
	사용자 매뉴얼 및 UI 분석	<b>UI 구조 분석</b> * 사용자 매뉴얼 및 실행을 통하여 각 UI 요소간의 구조를 파악, UI 구조도 작성. <b>UI 기능 분석</b> * 사용자 매뉴얼 및 실행을 통하여 각 UI 요소의 기능을 파악, 구조와 각 기능에 따른 이벤트를 네비게이션 트리 로 기술.	입력물 : Manual, Executable Source, UI. 산출물 : UI Structure/ Function Description.
	데이터 분석	<b>데이터 분석</b> * UI를 통한 데이터 분석 : 사용자 매뉴얼 및 UI를 분석, 입력과 출력에 관계된 데이터를 추출, 데이터 사전 구축. * 원시코드를 통한 데이터 분석 : 원시코드를 분석 데이터 유형과 입·출력 되는 데이터 파악. * DB를 통한 데이터 분석 : 직접 데이터베이스를 분석하여 데이터와 유형과 관계 추출. <b>데이터 흐름 분석</b> * UI, 원시코드, Data Base를 통하여 데이터의 입력, 변환, 출력 과정 정의.	입력물 : Manual, Executable Source, UI, Physical DB. 산출물 : DD(Data Dictionary), DFD(Data Flow Diagram)
	원시코드 분석	<b>환경 분석</b> * 대상 원시소스 파일의 전체적인 개발 환경(언어, 개발 툴, 아키텍처, etc), 개발 규칙 분석. <b>파일 구조 분석</b> * 대상 원시소스 파일의 전체적인 구조 파악, 각 파일간의 구조, 관계 정의, 서브시스템 정의. <b>파일 명세</b> * 파일단위, 모듈단위, 클래스단위, 함수단위, 변수 등의 정보를 CASE 도구를 이용하여 명세화 한다.	입력물 : Source Code. 산출물 : File Description (Environment, File, Relation, Module, Class, Function, Variable).
R0400 추가정보 추출	R0103을 통해 일차 분석, 추출된 산출물간의 상호 관계 추적. 재공학 목적에 적합한 추상화된 이차 산출물 추출, 일관성 정의.		
	비즈니스 모델링	<b>도메인 분석</b> * 전 단계의 UI 기능/구조 분석, 데이터 흐름 및 도메인 지식을 기반으로 문제 영역 및 도메인 지식 정의. <b>업무 분석</b> * 전 단계의 UI 기능/구조 분석, 데이터 흐름 및 도메인 지식을 기반으로 업무 프로세스 추출, 추출 업무를 기반으로 데이터 사전과 데이터 흐름 분석, 자료를 이용 데이터의 변환 과정을 정의.	입력물 : UI Structure/ Function Description, File Description 산출물 : DFD, Business Process Use Case, Use Case, Scenario
	데이터 모델링	<b>ER(Entity-Relation) 분석</b> * 전 단계의 산출물을 기반으로 E-R 추출, 물리적인 데이터베이스에서 E-R 추출.	입력물 : Physical DB, 산출물 : ERD
R0500 역공학 전략 검증	역공학을 통한 최종 산출물이 프로젝트의 목적과 재공학 목표와 적합성 정의.산출물과 목표간의 영향은 R0100 단계로 피드백		
	시스템 인벤토리 구축 역공학 전략 검증	<ul style="list-style-type: none"> <li>산출물 수준 및 관계를 파악하고 시스템 인벤토리를 작성한다.</li> <li>최종산출물 수준이 역공학 목표에 부합하는가를 판단, 전략을 재정립한다.</li> </ul>	입력물 : 산출물 : System Inventory

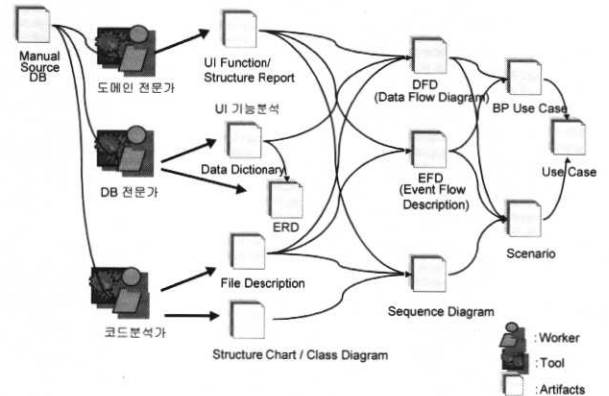


(그림 5) MVC 모델을 기반으로 분류한 R0103, R0104간 액티비티와 산출물의 연관도

R0000 단계의 역공학 기법은 효율적 역공학 산출물 추출을 위하여 기존의 레거시 시스템의 정보를 MVC 모델[13]을 기반으로 분리하여 분석을 수행한다. Model 관점은 레거시 시스템의 데이터(Data) 정보, View 관점은 레거시 시스템의 UI와 이를 통한 행위와 이벤트 분석, Controller 관점은 레거시 시스템의 기능(Function) 및 로직 분석을 그 목적으로 한

다. 이러한 서로 다른 관점에 의하여 추출된 산출물들은 상호 연관성에 의하여 또 다른 관점 및 다양한 추상화 수준의 정보를 추출 할 근거로 활용되어진다. 또한 UI/Function/Data 관점의 역공학 산출물은 Small-h 모델의 하부계층 구성요소이기도 하다.

(그림 6)은 역공학 단계를 통하여 추출되는 산출물들과 추상화된 산출물로 전개 되어지는 단계 및 주요 산출물간의 참조 관계를 정의하였다.



(그림 6) 주요 역공학 산출물간의 연관도

〈표 4〉 변환 워크플로우의 세부 워크플로우 및 액티비티 설명

T0000	Activities	내용 및 목적	산출물
T0100 재공학 목표 수립	역공학을 통하여 추출된 산출물을 기반으로 재공학의 목표 및 재사용 수준, 재공학 기법 정의.		
	재공학 목표 설정	<ul style="list-style-type: none"> <li>유지보수(Maintenance), 교체(Replace)등 재공학(Re Engineering) 수행 목표 및 수행 범위를 판단.</li> <li>재공학이 가능하다면 재공학 목표 설정.</li> </ul>	
T0200 역공학 산출물 검증	재공학 기법 설정	<ul style="list-style-type: none"> <li>재공학이 가능하다면 재사용 수준 및 재공학 기법 정의.</li> <li>역공학 산출물의 수준에 따라 소스 레벨, 디자인 레벨, 아키텍처 레벨 등의 다양한 재사용 수준 정의.</li> <li>화이트박스(White Box), 블랙박스(Black Box) 혹은 래핑(Wrapping), 스크린 스크래핑(Screen Scraping), 미들웨어(Middleware), EAI(Enterprise Application Integration)등의 재공학 기법 정의.</li> </ul>	재공학 계획서 (Reengineering Plan)
	산출물 정의	R0201의 목표 및 기법에 적합한 역공학 산출물 선별, 추출, 추가 산출물 정의.	
T0300 변환 영향 및 가설 설정	산출물 검증	<ul style="list-style-type: none"> <li>R0201의 목표 및 기법에 적합한 형태, 수준, 일관성 등을 검증.</li> <li>목표에 부적합하다면 R0201 단계로 피드백.</li> </ul>	재공학 프레임워크 (Reengineering Framework)
	현 산출물을 활용하여 재공학을 수행할 경우 현 산출물들과 새로운 요구사항이 반영된 새로 생성될 산출물간의 일관성 및 변환의 영향을 예측, 설정.		
	재공학 기법에 따른 변환 영향 요소 예측.	재공학 기법에 따른 시스템 전반에 미치는 영향 및 위험 요소를 파악, 기술.	
T0400 작업 단계 분기 결정	재공학 기법 결정에 따른 변환 가설 설정.	재공학 기법에 따른 시스템 및 산출물들의 변환 가설 및 위험을 추정.	
	추정된 산출물 세트(Artifacts Set)의 척도를 기준으로 순공학 단계(Phase)를 최소화하기 위한 분기 결정.		
	재사용 산출물 식별	T0200, T0300의 결과를 바탕으로 이해 타당한 재사용 단위를 식별한다.	
	분기 결정	<ul style="list-style-type: none"> <li>유지보수(Maintenance), 교체(Replace)등 재공학(Re Engineering) 수행 목표 및 수행 범위 결정.</li> <li>T0100, T0200, T0300의 산출물을 근거로 어플리케이션 개발 프로세스의 특정 단계 (Inception, Elaboration, Construction, Transition Phase)로 분기 판단.</li> <li>재공학의 목표에 부합되는 산출물들은 특정 단계의 반복을 수행하는 산출물 세트 (Artifacts Set)로 정의.</li> <li>산출물 세트를 해당 워크플로우들로 분기 수행, 순공학의 단계를 최소화하는 것이 목표.</li> </ul>	

3.2 변환 워크플로우(Transformation Workflow : T0000)

변환 워크플로우(T0000)의 세부 워크플로우 액티비티 및 산출물은 앞의 <표 4>와 같다.

변환 워크플로우에서는 재공학을 수행하여야하는 범위 영역의 정의, 명확한 재공학 목적 및 목표 수립, 레거시 시스템의 양/질적인 충실한 문서화, 실용적이고 실질적인 레거시 시스템의 재공학 지원, 적합한 재공학 전략 및 기술의 선정, 재공학의 대안 제시, 재공학 전략 및 기술의 검증을 목표로 한다. 또한 역공학 단계에서 추출된 산출물들을 새로

운 요구사항에 재사용되어질 재사용 산출물들을 정의하고 현재 추출되어진 재사용 산출물들을 진화 워크플로우로 효율적 연계를 제공하여야 한다.

기존 시스템의 명확한 수준 판단과 적용하여야 할 재공학 기법의 선택 및 재공학 전략 결정을 지원하기 위한 기법으로 다음과 같은 재공학 프레임워크(Reengineering Framework)를 제공한다.

<표 5>는 재공학 변환 워크플로우를 지원하기 위한 재공학 프레임워크의 예이다. 이 재공학 프레임워크는 John

<표 5> 재공학 프레임워크(Reengineering Framework)의 템플릿

	DATA Artifacts 자료 관점의 산출물	FUNCTION/UI Artifacts 기능 관점의 산출물	Process Network 역공학 산출물의 분배	People/Work 요구 작업자/작업	Time/Tech Profile 적용 가능한 기술	MOTIVATION/Objectives 재공학의 목적
SCOPE (Contextual) Planner	정의 : List of Business Thing Entity = 도메인 정의서 업무 리스트 업무 정의서 Relationship =	정의 : Class of business process Process = 도메인 정의서 업무 리스트 업무 정의서 Algorithm Pseudo Code I/O =	정의 : List of Phase / Workflow Locations in Which the Reverse Artifacts Phase = Inception Workflow = Project Management	정의 : List of Worker important to the Re-Engineering Process Worker = Owner Work =	정의 : Profile of Re-Engineering Technology Tech = Architecture Domain Knowledge	정의 : List of Re-Engineering Goals/Strategies Ends/Means = Major Re-Engineering Goal/Strategy Mission Statement
BUSINESS MODEL (Conceptual) Owner	정의 : Semantic Model Entity = Business Entity DFD(Business Role, Concept) Data Dictionary Conceptual ER Diagram Relationship = Business Relationship	정의 : Business Process Model Process = DFD Business Process Use Case Model Conceptual Class Diagram I/O = Business Resource	정의 : Phase = Inception, Elaboration Workflow = Business Modeling	정의 : Worker = Domain Expert. Use Case Modeler Work = Usecase Clustering, Restructuring	정의 : Tech = Conceptual Model Model Reuse Wrapping	정의 : Re-Engineering Plan End = Re-Engineering Objective Re-Engineering Plan Means = Re-Engineering Strategy Critical Success Factor
SYSTEM MODEL {Logical} Designer	정의 : Logical Data Model Entity = Data Entity Attribute Entity Class Dataflow Primary/Foreign Key ER Diagram Relationship = Data Relationship	정의 : Application Architecture Process = Use Case Model Analysis Model (Class, Sequence Diagram) STD I/O = User View	정의 : Phase = Inception, Elaboration Workflow = Requirements Analysis	정의 : Worker = Architect Designer Work = Analysis Model Clustering Restructuring	정의 : Tech = Design Pattern Component Data, Logic, Interface Restructuring Refactoring	정의 : Re-Engineering Rule Model End = Re-Engineering Process Structural Assertion Means = Re-Engineering Process Action Assertion Process, Relationship
TECHNOLOGY MODEL (Physical) Builder	정의 : Physical Data Model Entity = Class Class Attribute Index Table Constraint 추출 산출물 원형 Relationship =	정의 : System Design Process = Design Model. (Class, Sequence Diagram) I/O = Data Elements, Sets	정의 : Phase = Elaboration, Construction Workflow = Analysis & Design	정의 : Worker = Designer Developer Work = Design Model Clustering Restructuring	정의 : Tech = Pure Code Reuse Restructuring Refactoring	정의 : Rule Design End = Re-Engineering Process Condition Means = Re-Engineering Process Action Artifacts, I/O, Worker, Activities
DETAILED REPRESENTATIONS {Out Of Context} Source	정의 : Data Definition Entity = source Code Field XML Attribute Element Schema Relationship =	정의 : Program Process = Source Code Function Class Subprogram Code Structure Binary Component I/O = Control Block	정의 : Phase = Construction Workflow = Implementation	정의 : Worker = Developer Coder Work = Code Clustering	정의 : Tech = Pure Code Reuse Code Transformation	정의 : Rule Specification End = Sub-condition Means = Step

Zachman의 프레임워크[14]의 행, 열의 요소를 RUP+re를 지원하는 재공학 프레임워크로 재구성 하였다. 각 행, 열의 구성은 다음과 같다.

열의 요소는 재공학을 지원하기위한 각 관점을 구분한 것이다. 열의 요소는 아래쪽으로 진행될수록 추상화 수준이 낮아진다

<표 6> 재공학 프레임워크의 열 요소와 설명

항 목	내 용
Scope	계획자 관점. 각 요소별 개략적인 기능 범주를 식별하고 최종적으로 구현될 체계가 수행 할 기능, 규모 및 타 체계와의 관련성 정립. 추상화의 정도가 가장 크다.
Business Model	기능을 정의하고 포함하는 엔티티, 프로세스 및 그들간의 상호 연계성 정립.
System Model	상, 하위 모델을 근거로 구축되어지는 논리적 개념 표현 모델.
Technology Model	특정 구현 기술과 관련된 구현 기술 종속적 모델.
Detailed Representations	구축자 관점. 전체 시스템 개발명세의 특정 분야 및 세부 제약사항까지를 상세히 정의한다.

행의 요소는 관점에 대한 구성요소를 포함하고 있다.

프레임워크에서 요소들 사이의 상호관계와 정보교환은 일반적으로 수평/수직 방향으로 이루어지며 프레임워크의 구성은 조각(Sliver)라 정의되는 각 셀간의 연계에 의하여 표현된다. 조각은 하나 또는 여러 셀의 일부분을 연상하는 용어로 정의되며 전체 프레임워크의 셀을 몇 개의 수평, 수직적 조각으로 나타내는 것은 전체 재공학 프로세스 범위의 정의를 잃지 않고 이해 가능한 몇 개의 부분들로 시스템을 분할하는 방법을 제공한다[14].

이 역공학 프레임워크의 Data와 Function/UI 행의 요소들은 역공학을 통하여 추출된 산출물들이 일관성을 유지하며 각 요소들로 구성되어진다. Process Network, People/Work, Time/Tech Profile, Motivation 행의 요소들은 Data와 Function/UI 행의 역공학 산출물과 새로운 요구사항을 반영하여 진행되는 개발 프로세스를 함께 고려하여 구성되어진다.

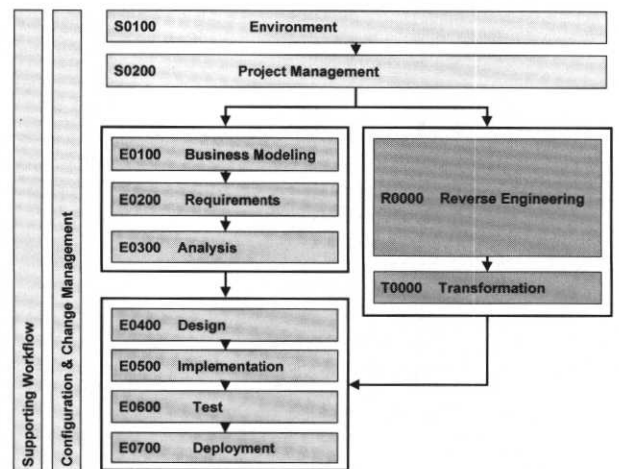
본 재공학 프레임워크는 역공학 산출물을 새로운 요구사항에 수용하는 가능성 및 방법, 기법등을 효율적으로 판단하기 위하여 프레임워크를 구축하는 각 셀을 구성 하므로서 레거시 시스템의 현 상황과 새로운 요구사항을 명확히 파악, 재공학을 통하여 재사용하기에 적합한 재사용 단위와

기법등을 판단한다. T0000을 통하여 검증, 정제되어진 산출물들은 수준에 따라 진화 워크플로우의 특정 단계(Phase)에 배치되어 재사용됨으로서 개발 기간을 단축, 효율적인 개발 수행을 지원한다.

3.3 진화 워크플로우(Evolution Workflow : E0000)

본 논문에서 제시하는 재공학 프로세스의 진화 워크플로우는 일반적인 어플리케이션을 개발 혹은 기 개발되어진 재사용 단위(예, Function, Module, Class) 및 상업용(COTS) 컴포넌트를 수용하여 시스템을 개발하는 컴포넌트 기반 개발을 목적으로 한다.

본 논문의 진화 워크플로우는 Rational사의 RUP를 기반으로 재공학을 수용할 수 있게 확장, 재정의 하였다. RUP는 일반적으로 언급되는 방법론(Methodology)과는 달리 조직의 특성과 프로젝트의 목표에 따라 프로세스 자체의 커스터마이징(Customizing)을 허용한다. 그러나 RUP는 레거시 시스템으로부터 재사용 단위를 추출하거나 추출한 재사용 단위를 획득, 적용, 공급에 적합한 프로세스는 아니며 이를 위한 가이드라인이 존재하지 않으므로 본 논문의 진화 워크플로우는 기존의 RUP를 기반으로 재공학을 위한 기존 재사용 단위의 획득, 적용, 공급에 관한 세부 워크플로우 및 액티비티를 각 워크플로우에 추가, 정의하여 구성하였다.



(그림 7) 프로세스 RUP+re의 워크플로우 구성도

(그림 7)과 같이 진화 워크플로우는 재공학 프로세스의 일부분으로 적용되어진다. 유지보수 및 기존의 시스템의 자원

<표 7> 재공학 프레임워크의 행 요소와 설명

항 목	Data	Function/UI	Process Network	People/Work	Time/Tech Profile	Motivation/Objectives
내 용	도메인의 혹은 역공학을 통한 산출물에 있어 업무, 데이터 관련점의 대상목록	도메인의 혹은 역공학 산출물에 있어 업무, 프로세스 관점의 대상목록	역공학 산출물과 순공학과와의 연계 관점의 해당 단계 및 워크플로우 대상목록	역공학 산출물 추출과 순공학 접목 및 개발을 위하여 수반되는 작업, 작업자	각 추상화 수준에서 적용 가능한 재공학 기법 및 기반 기술 목록	항목별 재공학 목표 및 제반 세부 사항



〈표 8〉 RUP+re 재공학 프로세스의 워크플로우별 산출물

워크플로우	산 출 물				
Environment	Development Case.				
Project Management	Software Development Plan, Iteration Plan.				
Reverse Engineering	Legacy System	[ ]는 산출물의 고유 번호, ( )는 현재 산출물을 추출하기 위하여 참조된 산출물의 번호이다.			최종 산출물세트의 분기 워크플로우
	Reverse Plan[0] Manual[1] Executable Code[2] Data Base[3] Source Code[4] Domain Knowledge[5]	UI Structure Report[6] (1, 2) UI Function Report[7] (1, 2, 4) Data Dictionary[8] (2, 3, 4) File Description[9] (4) Structure Chart[10] (4) Class Diagram[11] (4)	Data Flow Diagram[12] (6, 7, 8, 9)	Business Process Use Case[16] (12, 13) Scenario[17] (12, 13, 15) Use Case[18] (16, 17) System Inventory[19]	[16], Business
			Event Flow Description [13] (6, 7, 9) Entity Relation Diagram[14] (3, 4, 8) Sequence Diagram [15] (6, 7, 9, 11)		[17, 18], Requirement
					[6, 7], [8, 9], [11, 12], [13, 14], [15], Analysis/Design
	[4], Implementation				
Transformation	Re Eng Plan, ReEngineering Framework.				
Business	Supplementary Business Specification, Business Use Case, Business Object Model.				
Requirement	Vision Document, Supplementary Specification, Use Case, Use Case Specification.				
Analysis/Design	Analysis & Design Model.				
Implements	Implementation Model.				
Test	Test Plan.				
Deployment	Deployment Plan.				
C&C Management	Configuration & Change Management Plan.				

을 활용하여 개발하는 경우 역공학, 변환, 진화 워크플로우와 연계되어 진행되어지는 재공학(R0100, T0100, E0100~E0700) 경로를 통하여 프로세스가 전개 되어진다. 기존자원의 재사용 불가능한 재개발의 경우는 일반적인 어플리케이션 개발(E0100~E0700) 경로를 통하여 프로세스가 전개 되어진다.

재공학의 경우 역공학 워크플로우(R0000)가 전개되어짐과 동시에 새로운 요구사항이 반영되어진 시스템의 개발(E0100~E0500)이 시작되어진다. 이 개발은 역공학의 산출물의 추상화 수준과 같은 수준까지 전개되어지고 역공학 산출물의 추상화 정도에 따라 가변적이다. 이러한 역공학 워크플로우와 새로운 시스템의 요구사항이 반영된 분석/설계 단계의 산출물들을 바탕으로 변환 워크플로우(T0000)를 통하여 역공학 산출물과 새로운 요구사항이 반영된 시스템 개발 산출물과 통합되어 개발이 전개되어진다. 즉 역공학 가능성 여부와 재공학 가능성 여부를 판별하여 재공학을 수행하거나 유지보수 혹은 재개발이 수행되어진다.

〈표 8〉은 재공학 프로세스 RUP+re 통하여 정의되는 주요 산출물이다. RUP의 주요 산출물들과 일관성을 최대한 유지 시켰으며 R0000을 통하여 정의되는 산출물들은 T0000의 결과에 따라 각각의 해당 워크플로우로 분기, 특정 단계(Phase)의 하나의 산출물 세트(Artifacts set)를 이룬다. 이 산출물 세트를 기준으로 개발 목적에 따라 점진적이며 반복적으로 개발되어진다.

4. 사례 연구 및 평가

본 논문에서 제시하는 재공학 프로세스인 RUP+re의 효율성을 검증하기 위하여 다음과 같은 서로 다른 특징을 가지는 두 가지 사례연구를 수행하였다.

대상 프로그램의 특징은 다음과 같다.

〈표 9〉 사례 연구 대상 프로그램의 특징

	사례 A	사례 B
도메인	중소형 병원관리 프로그램	중소 기업형 ERP 프로그램
프로그램 이름	ADAMS-H	e-ERP
Source	Visual Basic	Visual C++
LOC	약 70,000 라인	약 140,000 라인
문제점	개발 문서 없음	개발 문서 없음
요구사항	<ul style="list-style-type: none"> <li>유지보수를 위한 개발 문서의 추출</li> <li>기능 개선 및 기능 향상을 위한 모듈의 추가 Web 환경으로의 전환</li> </ul>	<ul style="list-style-type: none"> <li>유지보수를 위한 개발 문서의 추출</li> <li>C/S 환경에서 EJB 컴포넌트 기반 환경으로의 전환</li> </ul>
비 고	<ul style="list-style-type: none"> <li>Visual Basic에서 권장하는 방식의 코딩 룰을 적용하지 않았다.</li> </ul>	<ul style="list-style-type: none"> <li>UI 설계를 표준 MFC 컨트롤을 사용하지 않았음.</li> <li>DB를 파일 시스템을 사용하였다.</li> </ul>

사례 연구의 대상 업체는 개발관련 문서가 거의 전무하므로 유지보수에 큰 어려움을 겪고 있었으며 일부 개발자

를 제외하고 도메인에 관한 전반적인 이해도 또한 낮은 상황이었다. 요구사항은 유지보수에 필요한 전반적인 개발관련 문서의 추출 및 새로운 기능 및 기술을 적용한 재개발을 수행하는 것이었다. 실 사례는 A를 위주로 제시하였다.

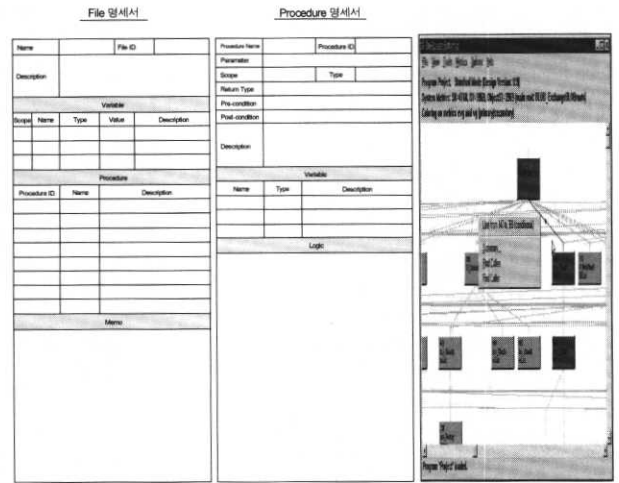
4.1 워크플로우별 적용

4.1.1 R0000(Workflow : Reverse Engineering)

● R0300(Detailed Workflow : 기존 정보 검증 및 명세화)  
R0300-Activity 1 : 매뉴얼 및 UI 분석

UI 구조 분석 및 UI 기능 분석은 사용자 매뉴얼 및 UI 실행을 통하여 각 UI 간의 구조 파악, 네비게이션 트리 작성. 각 UI의 기능 파악, 각 기능에 따른 이벤트의 흐름을 기술한다. Visual Basic 언어의 특성상 UI 관련 구조와 기능 분석은 수월하게 진행되었다.

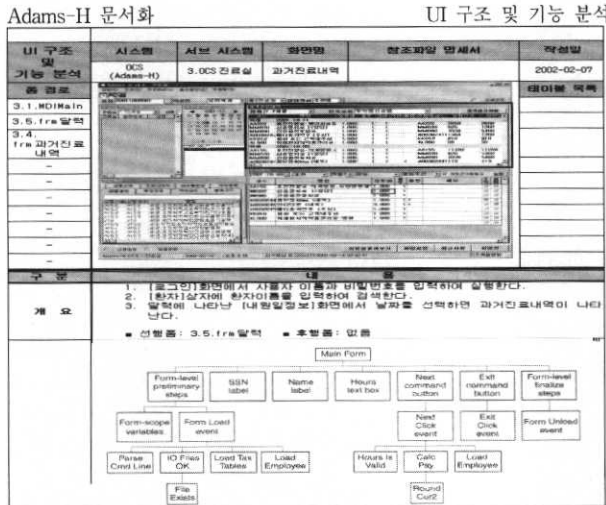
비 구조화된 모듈을 분할 및 기능별로 집단화한다.



(그림 9) File, Module Description Template, Structure Chart(McCave의Battlemap)

● R0400(Detailed Workflow : 추가 정보 추출)  
R0400-Activity 1 : 비즈니스 모델링

기초적인 도메인 지식, UI 기능, 구조 보고서, 데이터 사전, 이벤트 흐름 등의 전 단계의 산출물을 근거로 도메인을 명확히 정의하고 비즈니스 프로세스를 추출, 변환되어지는 데이터를 추적 DFD로 작성하였다.



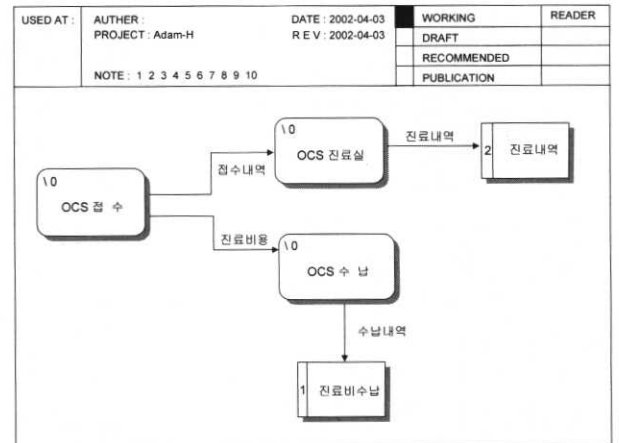
(그림 8) UI 기능, 구조 명세서

● R0300-Activity 2 : 데이터 분석

데이터 분석 및 데이터 흐름 분석은 사용자 매뉴얼 및 UI를 분석하여 입력과 출력에 관계 되어진 데이터들을 우선적으로 추출, 데이터 사전을 구축하였다. 또한 UI 기능의 이벤트에 따른 변환흐름을 분석하였다.

● R0300-Activity 3 : 원시 코드 분석

자동화 역공학 툴인 McCabe를 이용하여 기본적인 파일 정보 분석 및 구조 분석 과정을 수행하기 위하여 한글로 처리된 변수를 영문으로 수작업 변경 하였고 변경된 원시 코드의 입력을 통하여 1차적인 구조도를 추출해 내었다. 1차로 추출한 구조도를 바탕으로 파일명세서를 구성할 파일의 정보, 모듈간 호출관계 분석, 모듈간 파라미터 분석 등, 세부 사항을 분석함과 동시에 필요에 의해 슬라이싱 및 클러스터링을 통하여 주어진 원시코드에 대해 기능별로 분석 가능한 모듈로 클러스터링 하였다. 그리고 수작업에 의해



(그림 10) 추출되어진 DFD(Data Flow Diagram)

● R0400-Activity 2 : 데이터 모델링

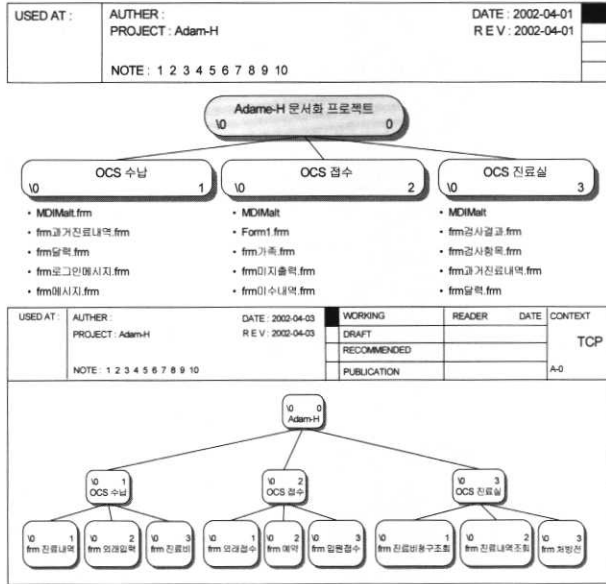
업체의 물리적인 DataBase를 Case Tool을 이용하여 E-R Diagram을 추출 하였다.

● R0400-Activity 3 : 시스템 모델링

상위 단계를 통한 산출물을 이용하여 모듈, 클래스간의 구조 및 관계를 명세화 하는 단계이다. 시스템의 분석대상을 개략적인 부분에서 최대한 상세화해 보여주는 것이며, (그림 12)에서는 기능 모델에 대한 구조를 계층적으로 표현

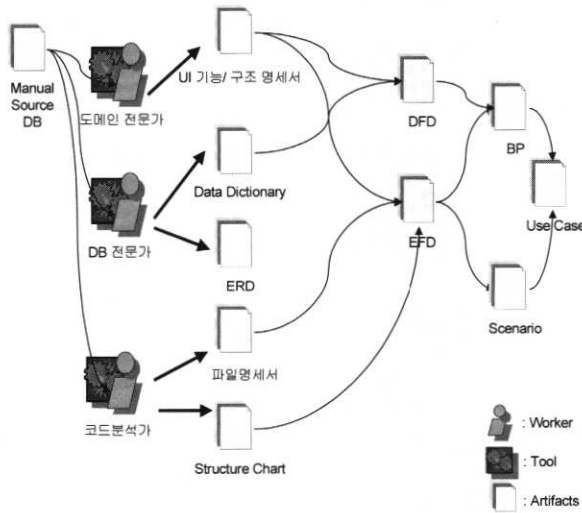
하여 전체적인 모듈 기능 관계를 표현해 준다.

가능한 부분은 복잡도, 응집도 등의 해당 매트릭을 적용하여 슬라이싱 및 클러스터링을 수행 하였다. 주어진 원시 코드에 대해 기능별로 재사용 가능한 모듈로 클러스터링 및 수작업에 의해 비 구조화된 모듈을 재구성하였다.



(그림 11) Structure Chart

사례 A의 R0000를 통하여 추출된 산출물과 각 산출물간의 연관 관계는 (그림 12)와 같다.

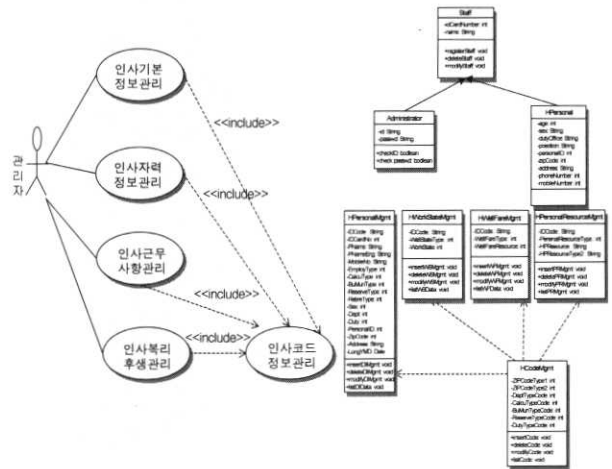


(그림 12) R0000을 통하여 실제 추출된 사례 A의 산출물의 연관도

4.1.2 E0100-E0200(Workflow : Evolution)

역공학 워크플로우가 진행됨과 동시에 새로운 시스템의 요구사항을 명확하게 파악하기 위하여 시스템 분석을 수행 한다. 진화 워크플로우에서 선행되어지는 시스템 분석의 수

준은 일반적으로 역공학 워크플로우의 최상위 산출물의 레벨과 같게 수행되어진다. 본 사례에서는 역공학을 통하여 구조도와 도메인의 유스케이스 까지 추출 가능하였고 새로운 시스템에 수용되어질 부분을 전환 워크플로우에서 판단 하게 된다.

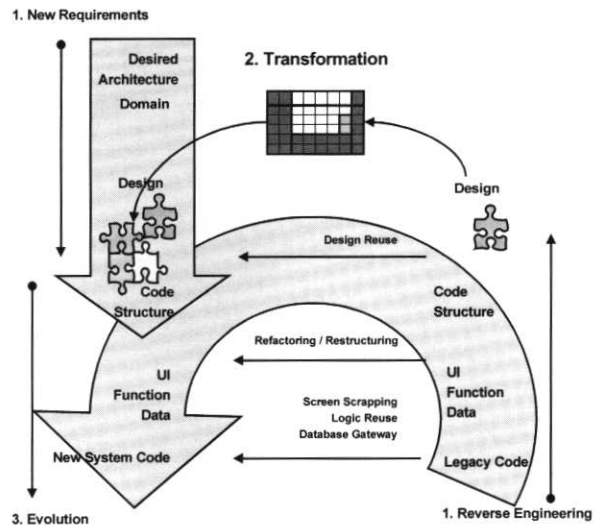


(그림 13) 새로운 시스템의 요구사항이 반영된 분석, 설계모델의 예

4.1.3 T0000(Workflow : Transformation)

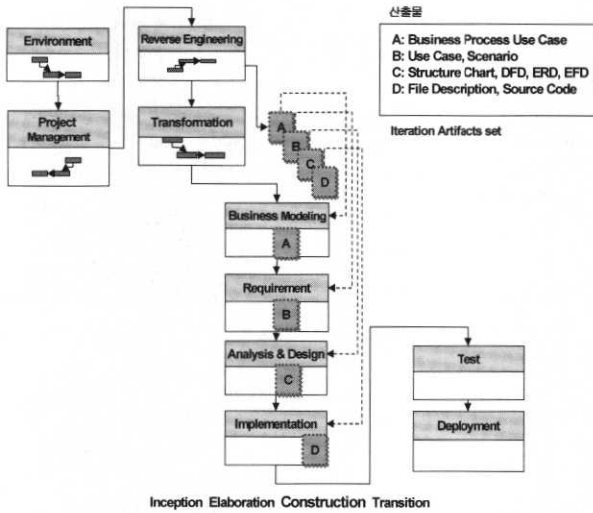
Detailed Workflow : T0100, T0200, T0300, T0400

전환 워크플로우의 목적은 재공학 가능성 및 재사용 수준 판단과 효율적인 기법설정에 그 목적을 두고 있다. 재공학 프레임워크를 구축, 현 시스템의 재공학 가능성 및 효율적인 재사용 기법을 수립 하는데 도움을 준다. 또한 이를 근거로 재사용 산출물의 진화 워크플로우로의 분기를 결정 짓는다.



(그림14) 변환 워크플로우의 수행

(그림 15)에서와 같이 각각의 산출물은 T0000의 T0100, T0200, T0300 세부 워크플로우를 통하여 화이트 박스형태의 재구성성을 통한 진화 워크플로우의 구현 단계(Construction Phase)의 산출물 세트(Artifacts set)으로 전환, 각각 해당 워크플로우로 수용되었다.



(그림 15) T0000을 통하여 진화 워크플로우로 분기한 재사용 산출물 세트

4.2 평 가

사례 연구를 통해 본 연구 결과를 정리해 보면 <표10>과 같다.

사례 A의 경우 재공학 요구사항을 반영하기 위하여 R0000 단계 산출물을 효과적으로 추출 가능하였고 추출된 산출물의 일관성 및 품질 수준이 재사용 요구사항을 만족하였다. T0000를 통하여 역공학 산출물을 검증, 재공학 가능성 판단, 순공학의 Inception, Elaboration 단계를 생략, Construction 단계의 각 워크플로우의 산출물 세트로 분기 하였다. 분기된 산출물 세트에 추가적인 요구사항을 직접 반영 가능하였고 요구사항이 반영된 산출물간의 일관성도 추적 가능하였다. C/S 환경에서 Web 환경으로 변환하기 위하여 Visual Basic 원시코드의 변환에 자동화 틀을 이용한 코드수준의 재사용이 90% 이상 가능하였다.

사례 B의 경우 개발환경의 문제로 인하여 R0000의 산출물간의 일관성을 유지할 수 없었으며 품질 수준이 재공학 목적을 충족시키지 못하고 T0000 단계를 통해 Inception 단계로 분기가 가능하였으나 추가적인 요구사항을 반영하기 위하여 도메인 분석이 재 수행되는 등 R0000 단계의 역공학 산출물이 E0100, E0200, E0300를 통하여 정제(Refine) 및 재개발(Re Development)되어야 하였다. 또한 EJB 기반의 컴

<표 10> 산출물 재사용 사례 결과표

입력물	최종 산출물	자동화 작업 도구	재사용 산출물 및 재사용율 평가		결론
사례 A	UI Structure/Function-Report. Data Dictionary. File Description. Structure Chart. DFD. EFD. ERD. Business Process - Use Case. Scenario. Use Case.	Together, Rose, 수작업. 수작업. McCabe, 수작업. McCabe. 수작업. McCabe. ERWin. 수작업. 수작업. 수작업.	Business Process Use Case.	최 적	역공학 워크플로우를 통하여 효율적인 재사용 산출물 추출 가능.  변환 워크플로우를 통하여 효율적인 재사용 목표 설정 및 산출물 분기 가능.
			Use Case. Scenario. ERD. Structure Chart. DFD.	최 적 최 적 최 적 최 적	
			Source Level Reuse Design Level Reuse	적 합 최 적	
			산출물 수준 및 일관성	최 적	
사례 B	UI Structure / Function-Report. Data Dictionary. File Description. DFD. Class Diagram. Sequence Diagram. ERD Business Process - Use Case. Scenario. Use Case.	Together, Rose, 수작업. 수작업. McCabe, 수작업. 수작업. Together, Rose. Together. 수작업. 수작업. 수작업. 수작업.	Business Process Use Case. Use Case. Scenario. ERD. Class Diagram. Sequence Diagram.	보 통 보 통 보 통 부적합 보 통 보 통	역공학, 변환 워크플로우를 통한 진화 워크플로우와의 효율적인 연계지원, 재사용 산출물을 통한 개발 프로세스의 특정 단계(Phase) 생략 가능.
			Source Level Reuse Design Level Reuse	불 가 부적합	
			산출물 수준 및 일관성	보 통	
			산출물 분기 단계	Inception	
재사용율 : 최적(수정 없이 전부 재사용가), 적합(약간의 수정 후 재사용가), 보통(일부 수정 및 정제 작업 후 재사용가), 부적합(일부 수정 및 정제 후 부분적으로 재사용가), 불가(재사용 할 수 없음)					

## 참고 문헌

포넌트 환경으로의 변환은 Visual C++ 원시코드의 Java 소스 수준의 재사용은 불가능한 하였다. 최종적으로 비즈니스 로직과 도메인 정보에 관한 디자인 수준의 일부분만이 재사용 가능하였다. 실질적으로 역공학 과정인 R0000을 통하여 추출되는 산출물의 수준은 레거시 시스템의 현재 상태와 그것을 역공학 하기 위해 사용 가능한 자동화 도구가 주요 변수가 되었다.

R0000 워크플로우의 문제점으로는 첫째, 개발된 원시코드 크기가 대규모인 경우 분석을 위하여 전체적인 구조를 수작업을 통해 단계적으로 분할해야 하는 문제점을 보여주었다. 둘째, 역공학을 통한 일차 추출 산출물들의 일관성이 자동적 추적되지 않기 때문에 추상화 단계의 수준이 높아 질수록 산출물간의 일관성을 검증하는데 어려움이 따랐다.

T0000 워크플로우의 문제점으로는 첫째, 재공학 할 것인지, 새로운 시스템으로 교체 할 것인지 여부의 결정을 하기 위한 정형적인 비용 산정 기법이 없어 휴리스틱하게 수행하였고 둘째, 재공학을 수행하기로 결정된 경우 산출물의 품질측정, 일관성 검증, 프로세스로 분기 결정에 정형적인 기법이 부족하였다.

## 5. 결 론

본 논문에서는 제시하는 RUP+re는 RUP를 확장, 커스터마이징 하여 레거시 시스템 진화를 통한 재공학 프로세스로 구성하였다. 제시하는 재공학 프로세스는 역공학, 변환, 진화 워크플로우로 구성된다.

역공학 워크플로우에서는 역공학을 통해 추출해야 하는 산출물 정의, 산출물간의 연관성 및 일관성을 통하여 추상화된 이차 산출물을 추출할 수 있는 효율적인 추출 방법을 제시하였다.

변환 워크플로우에서는 역공학 산출물을 통하여 재공학의 목표를 달성할 수 있는 판단 근거와 추출된 역공학 산출물을 활용한 순공학으로의 효율적인 프로세스 분기법을 제시하였다.

진화 워크플로우는 역공학, 재공학 워크플로우와의 연계를 위하여 RUP를 커스터마이징 하여 포함하였으므로 단순한 개발 프로세스로 혹은 레거시 시스템 재사용을 위한 재공학 프로세스로 각각 지원 가능하다.

본 논문을 통하여 제시한 레거시 시스템의 진화를 위한 재공학 프로세스는 기존의 기술 및 프로세스들이 역공학, 순공학, 혹은 재공학 등으로 각각 분류되어 프로세스간의 상호 연계성 및 효율성을 찾지 못하는데 반하여 역공학, 재공학 워크플로우를 소프트웨어 진화를 기반으로 일련의 재공학 프로세스로 통합, 상호간의 효율적 연계로 레거시 시스템의 재사용 및 재공학 프로세스의 효율성을 최대화 하였다.

- [1] Rene R. Klsch, *Reverse Engineering : Why and How to Reverse Engineer Software*, Proceedings of the California Software Symposium, 1996.
- [2] Penteado, R. Masiro, P. C. Cagnin, M. I., *An Experiment of Legacy Code Segmentation of Improve Maintainability*, CSMR'99, Amsterdam, Proceedings, IEEE, pp.111-119, 1999.
- [3] SEI Reengineering Center, *Perspectives on Legacy System Reengineering DRAFT Version 0.3*, Carnegie Mellon University, Software Engineering Institute, 1995.
- [4] Dolly M. Neumann, *Evolution Process for Legacy System Transformation*, ACM, 1998.
- [5] Gerald C. Gannod, *PACKRAT A Software Reengineering Case Study*, IEEE, 1998.
- [6] Rober C. Seacord, John Robert, *A Survey of Legacy System Modernization Approaches*, Technical Note CMU/SEI-2000-TN-003, April, 2000.
- [7] RENAISSANCE Consortium, *RENAISSANCE version 3.1 : Method and tool support for the evolution and reengineering of legacy systems*, 1998.
- [8] Rick Kazman, Steven G. Woods, S. Jeromy Carriere, *Requirements for Integrating Software Architecture and Reengineering Models : CORUM II*, Working Conference on Reverse Engineering, 1998.
- [9] at URL : <http://www.systemtransformation.com/TMethodology.htm>.
- [10] Gregory Abowd, Ashok Goel, Dean F. Jerding, Michae McCracken, Melody Moore, J. William Murdock, ColiPotts, Spencer Rugaber and Linda Wills, "MORALE-Missio Oriented Architectural Legacy Evolution," *Proceedings International Conference on Software Maintenance '97*, Bari, Italy, pp.150-159, September~October, 1997.
- [11] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Software Development Process*, The Addison-Wesley Object Technology Series, 1999.
- [12] Ilwoo Choi, sungyul Rhew, *An Efficient Reengineering Process based on RUP for Reusing Legacy Systems*. Journal of KISS : Software and Applications, Vol.29, pp.619-630, October, 2002.
- [13] Gienn E. Krasner and Stephen T. Pope, *A Cookbook for using the Model-View-Controller User Interface Paradigm in Smalltalk-80*, Journal of Object-Oriented Programming, 1988.
- [14] J. A. Zachman, *A Framework for Information system Architecture*, IBM System Journal, Vol.38, pp.454-470, 1999, [www.zifa.com](http://www.zifa.com).



### 최 일 우

e-mail : lucifer@selab.ssu.ac.kr

1995년 숭실대학교 전자계산학과 학사  
(공학사)

1997년 숭실대학교 컴퓨터학과 석사  
(공학석사)

2002년 숭실대학교 컴퓨터학과 박사 수료

관심분야 : 소프트웨어 개발 프로세스, 소프트웨어 재공학/  
역공학, 재사용, CBD, CBSE,



### 류 성 열

e-mail : syrheew@comp.ssu.ac.kr

1997년 아주대학교 컴퓨터학부(공학박사)

1997년~1998년 George Mason

University 교환교수

1998년~2001년 숭실대학교 정보과학  
대학원 원장

1981년~현재 숭실대학교 정보과학대학 컴퓨터학부 교수

1998년~현재 숭실대학교 전자계산원 원장

관심분야 : 소프트웨어 유지보수, 소프트웨어 재사용, 소프트  
웨어 재공학/역공학