

레거시 시스템의 재사용을 위한 RUP 기반의 효율적 재공학 프로세스

(An Efficient Reengineering Process based on RUP for
Reusing Legacy Systems)

최 일 우 [†] 류 성 열 ^{**}

(Il Woo Choi) (Sung Yul Rhew)

요 약 1980년대 이후, 소프트웨어 위기에 대응하기 위한 다양한 소프트웨어 공학적 기법들이 출현하기 시작하였고, 현재 소프트웨어의 질적 향상 및 생산성 향상을 피하기 위한 개발 지침으로 소프트웨어 컴포넌트 개발 프로세스를 구축하는데 관심이 집중되어지고 있다. 그러나 대부분의 방법론들이 앞으로 구축할 새로운 시스템의 재사용성 확보에 치중할 뿐, 실질적으로 기존 레거시 시스템의 자원을 재사용하여 새로운 시스템을 구축하는 방법은 효율적으로 지원하지 못하고 있는 실정이다.

본 논문에서는 레거시 시스템의 효율적 재사용을 위한 재공학 프로세스(Reengineering Process)로 RUP를 확장, 커스터마이징하여 구성한 RUP+re를 제시한다. RUP+re은 크게 역공학 워크플로우(Reverse Engineering Workflow), 재공학 워크플로우(Re-Engineering Workflow), 순공학 프로세스(Forward Engineering Process)로 구성되어진다. RUP+re의 각 워크플로우와 세부시스템을 제공하고 실사례 연구를 통하여 실질적으로 프로세스를 검증함으로써 레거시 시스템 자원의 재사용을 위한 소프트웨어 재공학 프로세스 구축에 효율적인 지침을 제공한다.

키워드 : 소프트웨어 개발 방법론, 재사용, 재공학, 역공학, 프로세스 커스터마이징, 알유피

Abstract The various software engineering techniques have been come out in order to cope with the software crisis since 1980's. The software Engineering has focused on the process of software component development. The process products a development 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 actually support efficiently how we reuse legacy system's resources and construct a new system.

In this paper, we present an efficient Reengineering Process for reusing legacy systems, RUP+re, which extends and customizes RUP. RUP+re consists of Reverse Engineering Workflow, Re-Engineering Workflow and Forward Engineering Process.

We describe RUP+re's workflows with their own detail steps and verify them practically through the case study. So, we provide efficient guidelines to the software Reengineering process construction which reuses the resources of a legacy system.

Key words : S/W Development Methodology, Reuse, Reengineering, Reverse Engineering, Process Customizing, RUP

1. 서 론

빠르게 변화하는 환경을 만족시키기 위해 정보시스템

은 점점 복잡해지고, 서비스 공급자나 개발자들이 사용자 요구에 맞는 시스템을 개발, 공급하기가 어려워지게 되었다. 이런 소프트웨어 개발의 한계성을 극복하기 위해 새롭게 대두된 것이 소프트웨어 개발 방법론(S/W Development Methodology)이다. 현재 소프트웨어 산업계에서는 새로운 응용 시스템 개발에 집중을 하면서 품질향상과 개발기간 단축 등 사용자의 욕구 충족을 위

[†] 학생회원 : 송실대학교 컴퓨터학과

lucifer@selab.soongsil.ac.kr

^{**} 종신회원 : 송실대학교 전자계산원 원장

syrhew@computing.soongsil.ac.kr

논문접수 : 2002년 4월 15일

심사완료 : 2002년 6월 27일

한 효율적인 소프트웨어 개발 프로세스(S/W Development Process)를 만드는데 치중하고 있다. 특히 재사용을 통한 생산성 향상에 치중한 컴포넌트 개발 방법론(Component Development Methodology)에 관심이 모아지고 있다.

기업들은 새롭고 다양한 기술적 요구사항들을 시스템에 반영하기를 원하며 이를 위한 정확한 분석 작업을 필요로 한다. 그러나 이러한 새로운 기술적 요구사항들을 시스템에 반영하기 위하여 기존의 자원을 모두 포기하고 새로운 시스템을 구축하는 것은 매우 어려운 일이다. 대표적인 소프트웨어 개발 프로세스인 Rational Unified Process(RUP)를 비롯하여 컴포넌트를 지원하는 Catalysis, UML Components 등의 여러 방법론들이 있으나 새로운 시스템의 재사용성 확보에 치중할 뿐, 실질적으로 기존의 레거시 시스템의 자원을 활용하여 새로운 시스템으로 재개발하는 방법 혹은 재사용 가능한 컴포넌트를 추출하는 방법 등은 명확히 제시되지 못하고 있다.

기 개발되어진 시스템은 실질적인 운용을 통하여 이미 검증 되었을 뿐만 아니라 유사 문제영역(Domain)의 어플리케이션 혹은 컴포넌트 개발에 필요한 문제영역 지식(Domain Knowledge)이 이미 확보되어 있는 상태이다[1]. 그러나 레거시 시스템 자원을 재사용 할 수 있는 실용적인 방법론의 부재로 기존의 자원들을 재사용하지 못하고 유사 문제 영역의 시스템을 구축하기 위하여 시스템을 완전 교체(Replacement)하고 있는 것이 현실이다[2].

레거시 시스템 자원을 재사용하여 소프트웨어 개발 생산성을 향상 시키는 방법 중 하나는 효율적인 역공학(Reverse Engineering)과 재공학(Reengineering)기법을 소프트웨어 개발 프로세스의 한 부분으로 통합하는 것이다.

본 논문에서는 이러한 필요성을 충족하고 문제점을 해결하기 위한 방안으로 역공학, 재공학 방법을 소프트웨어 개발 프로세스에 통합하여 기 개발되어진 레거시 시스템의 자원을 이용, 어플리케이션 개발을 지원하는 효율적인 재공학 프로세스(Reengineering Process)와 그 세부 절차를 제시한다. 또한 사례연구를 통하여 제시된 재공학 프로세스가 레거시 시스템 자원의 재사용에 기반 한 소프트웨어 개발에 효과적으로 적용될 수 있음을 보였다.

2. 관련연구

2.1 재공학(Reengineering)

소프트웨어 역공학(Reverse Engineering)이란, 자동화 도구(Case Tool), 수작업 등을 통하여 소프트웨어의

물리적 계층(Physical Layer)의 표현(예, 원시코드)을 소프트웨어 데이터 요소와 프로세스 등을 설명해 주는 명세 계층(Specification Layer)의 표현으로 재구성하는 과정을 말한다[1].

소프트웨어 재공학(Reengineering)이란, 자동화 도구, 수작업 등을 통하여 기존의 소프트웨어 시스템의 기능 및 디자인을 수정, 성능 향상 및 유지 보수를 꾀하는 과정을 말한다[3].

일반적으로 재공학 프로세스(Reengineering Process)는 역공학과 순공학(Forward Engineering)을 포함하고 있다. 그림1의 수행절차를 살펴보면 역공학, 분석, 재구성, 변환 등의 4단계를 거치게 되어 있다. 역공학과 분석 단계를 통하여 소프트웨어의 명세와 그들 간의 관계를 재구축하기 위한 활동을 수행하고, 재구성(Restructuring)을 통하여 소프트웨어의 외부행위에는 변환을 미치지 않고 소프트웨어의 내부 기능이나 구조를 변경하고[4], 변환(Transition)단계를 통하여 언어, 운영 체제, 아키텍처 등이 상이한 환경으로 전환하거나 기능 및 기술력을 향상하는 활동을 수행한다[5].

일반적으로 재공학 목적에 따른 방법으로는 시스템의 버그 수정 및 외부적 행위의 변경 없이 개선을 수행하는 유지보수(Maintenance), 유지보수 보다 광범위한 변경을 수행 하며 기존 시스템의 주요 부분을 그대로 재사용하는 현대화(Modernization), 레거시 시스템의 유지보수 및 현대화가 비용적인 측면이나 기술적인 측면에 부적합하여 시스템을 새로 재개발하는 교체(Replacement)등이 있다[6]. 또한 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)등 다양하게 연구되고 있다[3].

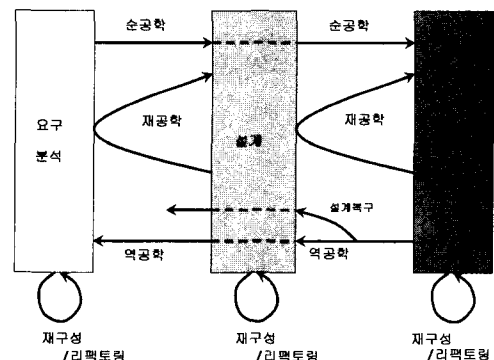


그림 1 역공학, 순공학, 재공학 간의 관계

재공학 프로세스의 역공학과 재공학 전략은 크게 기술적 관점과 비즈니스적 관점에 따라 구분한다[6].

기술적 관점은 레거시 시스템의 현재 상태와 관련되는 산출물을 정의, 추출하는 방법이다. 산출물을 역공학 및 재공학 하기 위해 적용하고자 하는 목표 수준(예, 소스 레벨, 디자인 레벨, 아키텍처 레벨, 프로세스 레벨 등) 및 기술(예, 화이트 박스, 블랙박스, 랩핑, 스크린 스크래핑, 미들웨어, EAI 등)과 관련하여 적용 가능한 자동화 도구(Case Tool) 등이 주요 변수가 된다[3,6].

비즈니스적 관점은 재공학 할 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 현대화할 것인지, 유지보수 할 것인지, 새로운 시스템으로 교체할 것인지 등의 결정을 필요로 하며 효율적인 재공학 프로세스는 이러한 두 가지 관점을 모두 충족시켜야 한다[3,6]. 그러나 현재까지 연구 되어진 재공학 프로세스들은 객체지향이나 컴포넌트 등의 새로운 기술을 지원 할 수 있는 역공학 방법 등이 정의되어 있지 않고 역공학, 분석, 재공학, 변환 등의 재공학 프로세스 단계간의 효율적인 연계 방법이 부족하다.

2.2 소프트웨어 개발 프로세스와 프로세스 커스터마이징(Process Customizing)

소프트웨어 개발 프로세스는 사용자의 요구사항을 소프트웨어 시스템으로 변환시키기 위한 일련의 활동이다. 현재까지 다양한 종류의 소프트웨어 개발 프로세스와 방법론들이 개발, 정의 되었으며 또한 기술 변화에 따라 많은 변화를 거듭하며 상호 보완, 발전되어 왔다. 국외의 대표적인 방법론으로는 RUP, Catalysis, Advisor 등이 있으며 국내의 대표적인 방법론으로는 한국 전자통신 연구원의 마르미III를 들 수 있다.

Rational사의 RUP는 유스케이스 기반(Use-case Driven), 아키텍처 중심(Architecture-Centeric), 반복 및 점증(Iterative & Incremental)에 의한 개발을 특징으로 들 수 있으며 산업계 표준으로 자리 잡고 있는 모델링 언어인 UML(Unified Modeling Language)을 기본으로 지원한다. RUP는 소프트웨어 개발 프로세스(Development Process)뿐만 아니라 프로세스 조직의 관리 프로세스(Management Process)도 지원한다. RUP는 다양한 유형의 소프트웨어 시스템, 응용영역, 조직체, 능력 수준 및 다양한 프로젝트 규모를 위하여 커스터마이징(Customizing) 될 수 있는 일반적인 프레임워크(Framework)이다. 여기서 커스터마이징 이란, 프로세스를 수정하여 이를 채택하는 조직체의 요구 및 목적,

제약사항등에 맞도록 프로세스를 개조하여 구성(Configuration)하고 구현(Implement)하여 사용하는 것을 말한다[7].

Catalysis는 D'Souza에 의하여 제안되었으며 비즈니스 모델로부터 소스코드에 까지 이르는 전 과정의 추적성을 보장한다. 불명확한 모델이나 문서를 최대한 명확하도록 만드는 정확성, 컴포넌트 기반의 개발, 코드뿐만 아니라 분석이나 디자인의 산출물들까지 확대된 재사용 개념, 확장성, 유연한 프로세스를 주요 특징으로 하고 있다[8].

Advisor는 John Dodd에 의하여 제안 되었으며 Sterling Software사의 CASE Tool인 COOL시리즈의 소프트웨어 개발 및 컴포넌트 개발의 각 단계와 상호연계에 대한 가이드를 제공한다[9].

한국 전자 통신 연구원의 마르미III는 컴포넌트 개발(Component Development) 및 컴포넌트 기반 소프트웨어 개발(Component Based Software Development) 모두를 지원하는 프로세스로 세부절차와 지침, 산출물 등이 상세하게 정의되어 있는 것이 특징이다[10].

각각의 방법론들은 각 조직 및 개발 특성에 따라 커스터마이징 되어 사용되어지나 현재는 정형화된 적용사례가 미비 하며, RUP를 이용 특정영역(Specific Domain)으로 커스터마이징 되어진 경험 사례들만이 배포되어지고 있는 실정이다. 아직까지 다양한 조직과 프로젝트의 특성을 반영한 프로세스의 커스터마이징 경험 및 가이드라인이 부족하며 이에 관한 정형적인 방법이 필요하다[11].

2.3 재공학 방법론

기존 레거시 시스템의 재사용을 지원하는 재공학 방법론으로는 UIRich System Redevelopment Methodology(USRM), Mission Oriented Architecture Legacy Evolution(MORALE) 등을 들 수 있다.

각 방법론의 특징을 살펴보면 USRM은 계획, 분석, 포지셔닝, 변환의 4단계로 구성되었으며 UI, 데이터, 기능 위주의 재공학 접근방식을 제공한다[12]. MORALE은 분석, 설계, UI진화의 3단계로 구성되었으며 UI를 중심으로 한 재공학 접근방식을 취하고 있다[13]. 각 방법론들은 역공학을 통하여 기존 레거시 시스템의 자원을 추출 가능하다는 전제하에 구성되어 있으며 역공학의 세부적인 단계는 다루지 않고 재공학 기법에 중점을 두었다. 그러나 두 방법론 모두 세부단계의 액티비티와 산출물이 명확히 정의되지 않았으며 개발 라이프 싸이클의 전 단계를 지원하지 못한다.

현재 레거시 시스템의 자원을 이용한 재공학 기술은

레거시 시스템의 자원을 화이트 박스(White Box), 블랙 박스(Black Box), 랩핑(Wapping), 스크린 스크래핑(Screen Scraping), 미들웨어(Middleware), EAI(Enterprise Application Integration) 등[3], 다양한 접근법으로 연구되어지고 있으며 방법론 또한 소프트웨어 개발 라이프 싸이클의 전 단계를 지원하도록 계속 연구중에 있다.

3. 재공학을 위한 확장된 RUP+re의 구성

본 논문에서 제시하는 재공학 프로세스 RUP+re은 RUP를 확장하여 재공학을 수용할 수 있도록 커스터마이징 하여 구성 하였다.

RUP+re의 정의 및 구성은 다음과 같다.

RUP+re 정의)

- Process =Phase(*).
- Phase =Workflow(*)+Iteration(*).
- Workflow =Detailed Workflow(*).
- Detailed Workflow=Activity(*) + Sequence(*).
- (*) = 반복.

RUP+re 구성)

첫째, **역공학 워크플로우(Reverse Engineering Workflow: R0100)**: 기 개발된 원시코드 및 산출물을 분석하여 새로운 시스템을 개발하기 위해 산출물을 정의, 추출하는 워크플로우.

둘째, **재공학 워크플로우(Re-Engineering Workflow: R0200)**: 역공학 산출물을 재공학 목적에 따라 효율적으로 순공학에 수용시키는 워크플로우.

셋째, **RUP 기반 순공학 어플리케이션 개발 프로세스(Application Development Process)**: 순공학 어플리케이션 개발 프로세스. RUP를 기반으로 역공학, 재공학 워크플로우를 수용할 수 있도록 확장, 재정의.

역공학 워크플로우는 역공학을 통한 효율적인 산출물 정의 및 추출 방법을 제시한다. 재공학 워크플로우는 역공학 산출물을 근거로 재공학의 기술적, 비즈니스적 관점을 효율적으로 판단, 정의하며 역공학 산출물을 어플리케이션 개발 프로세스의 특정 단계(Phase)의 워크플로우로 연계하는 효율적 방법을 제시한다. 어플리케이션 개발 프로세스는 역공학, 재공학 워크플로우를 통하여 정의, 연계된 산출물을 순공학에 획득, 적용하여 재공학 목적에 알맞게 개발하는 방법을 제시한다.

다음 그림 2는 본 논문에서 제시하는 RUP+re의 전체적인 프로세스 구성도이다.

역공학 워크플로우(Reverse Engineering Workflow: R0100)는 이미 개발된 시스템의 원시코드 및 기존 개발

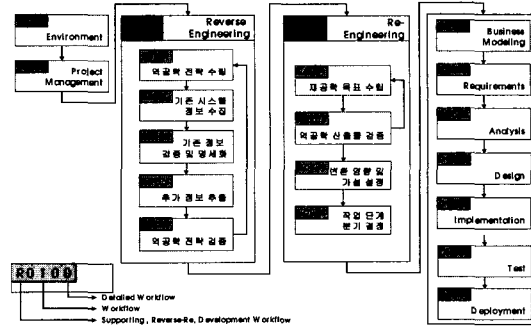


그림 2 RUP+re의 각 워크플로우와 역공학, 재공학 워크플로우의 세부 워크플로우

산출물을 입력받아 역공학을 통하여 다양한 형태의 재사용 가능한 시스템 분석, 설계 산출물을 생성, 다음 단계의 재공학 워크플로우의 입력물로 제공하는데 그 목적이 있다.

즉, R0100은 프로젝트 목적에 따라 역공학 전략을 수립, 기존의 시스템 자원을 최대한 활용하여 기본 산출물을 추출, 산출물간의 관계 및 일관성 정의를 통하여 상위 수준의 추상화를 제공하는 이차 산출물을 추출하는 단계로 역공학을 통한 효율적 산출물 추출 방법을 제시한다.

재공학 워크플로우(Re-Engineering Workflow: R2000)는 기술적인 측면과 비즈니스적인 측면에 있어서 재공학 방법 및 가능성 등을 판단하고, 역공학 워크플로우를 통하여 추출된 산출물을 어플리케이션 개발 프로세스의 특정 단계(Phase)의 워크플로우로 분기, 재공학을 위한 역공학과 순공학 프로세스와의 효율적 연계방법을 제시한다.

R0100을 통한 효율적인 산출물 추출 방법 및 R0200을 통한 순공학 어플리케이션 개발 프로세스의 효율적인 연계를 통하여 재공학 프로세스의 효율을 극대화 할 수 있다. 본 장에서는 RUP+re의 특징을 살펴보고 다음 장에서는 프로세스를 활용한 재공학 사례연구를 통하여 프로세스의 효율성 검증 및 문제점을 살펴본다.

3.1 RUP+re의 역공학 워크플로우(Reverse Engineering Workflow: R0100)

역공학 워크플로우(R0100)의 세부 워크플로우 및 목적은 다음과 같다.

R0101, 전체 프로젝트의 목적과 재공학의 목표에 부합하는 역공학 전략 수립.

역공학 전략에 따라 추출하는 산출물의 종류 및 방법 정의.

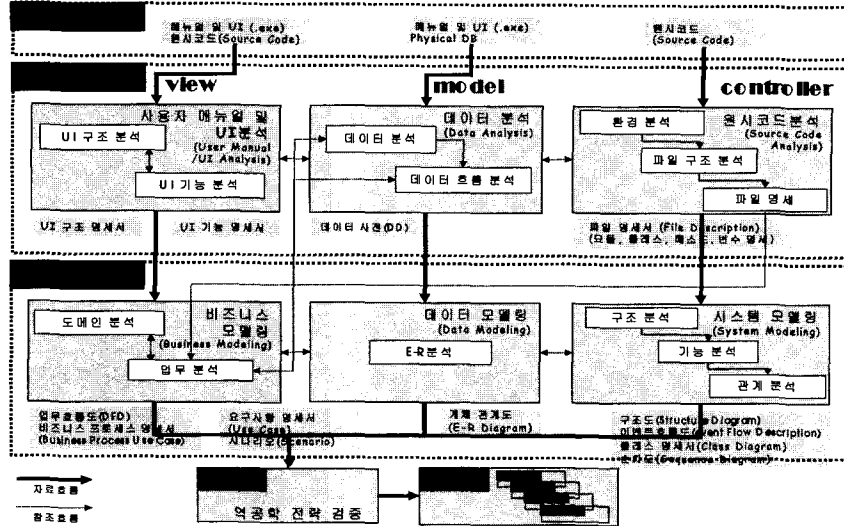


그림 3 MVC 모델을 기반으로 분류한 R0103, R0104간 액티비티와 산출물의 연관도

- R0102, 현 레거시 시스템 자원 수집, 상호 일관성 검증.
- R0103, 수집, 검증된 산출물을 MVC모델 기반으로 분류, 일차 분석을 통한 명세화.
- R0104, R0103을 통해 일차 분석, 추출된 산출물간의 상호 관계 추적.
재공학 목적에 적합한 추상화된 이차 산출물 추출.
- R0105, 역공학을 통한 최종 산출물이 프로젝트의 목적과 재공학 목표와 적합성 정의.
산출물과 목표간의 영향은 R0101단계로 피드백(Feedback).

다음 그림 3은 R0103과 R0104단계의 액티비티 및 산출물간의 관계를 도식화 한 것이며 표1을 통하여 R0103과 R0104단계의 세부 액티비티와 산출물들의 세부 설명을 제시하였다.

R0100단계의 역공학 기법은 효율적 산출물 추출을 위하여 기존의 레거시 시스템의 정보를 MVC모델[14]을 기반으로 분리하여 분석을 수행한다. Model관점은 레거시 시스템의 데이터 정보, View관점은 레거시 시스템의 UI와 행위 분석, Controller관점은 레거시 시스템의 기능 및 로직 분석을 그 목적으로 한다. 이러한 서로 다른 관점에 의하여 추출된 산출물들은 상호 연관성에 의하여 또 다른 관점의 혹은 다양한 추상화 수준의 정보를 추출할 근거로 활용되어진다. 다음 그림4는 역공학 단계를 통하여 추출되는 산출물들과 추상화된 산출물로 전개되어지는 단계 및 산출물간의 참조 관계를 정의하였다.

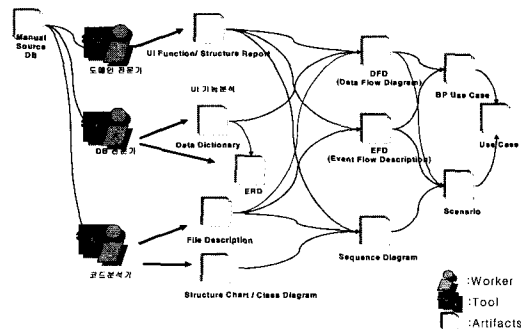


그림 4 역공학 산출물간의 연관도

3.2 RUP+re의 재공학 워크플로우(Re-Engineering Workflow: R0200)

재공학 워크플로우(R0200)의 세부 워크플로우 및 목적은 다음과 같다.

- R0201, 역공학을 통하여 추출된 산출물을 기반으로 재공학의 목표 및 재사용 수준.
재공학 기법 정의.
- R0202, 역공학 산출물이 재공학의 목표에 적합한 형태와 수준을 가지고 있는가 검증.
역공학 산출물이 재공학 목표에 부적합한 정보 수준이라면 재공학의 방법을 수정.
R0201단계로 피드백.
- R0203, 현 산출물을 활용하여 재공학을 수행할 경우 현 산출물들과 새로운 요구사항이 반영된

표 1 R0103과 R0104의 액티비티 및 산출물 설명

R0103 R0104	내용 및 목적	
사용자 매뉴얼 및 UI분석	(1) UI 구조 분석 사용자 매뉴얼 및 UI실행을 통하여 각 UI 요소 간의 구조를 파악, UI구조도 작성. (2) UI 기능 분석 사용자 매뉴얼 및 UI실행을 통하여 각 UI 요소의 기능을 파악, 구조와 각 기능에 따른 이벤트를 네비게이션 트리 로 기술.	
	입력물: Manual, Executable Source, UI.	출력물: UI Structure/Function Description.
비즈니스 모델링	(1)도메인 분석 전 단계의 UI 기능/구조 분석, 데이터 흐름 및 도메인 지식을 기반으로 문제 영역 및 도메인 지식 정의. (2)업무 분석 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(Module, Class, Function, Variable), DD. 출력물: DFD, Business Process Use Case, Use Case, Scenario.	
데이터 분석	(1)데이터 분석 UI를 통한 데이터 분석 : 사용자 매뉴얼 및 UI를 분석, 입력과 출력에 관계되어진 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. (2)데이터 흐름 분석 UI, 원시코드, Data Base를 통하여 데이터의 입력, 변환, 출력 과정 정의.	
	입력물: Manual, Executable Source, UI, Physical DB.	출력물: DD(Data Dictionary).
데이터 모델링	(1)ER(Entity-Relation) 분석 전 단계의 산출물을 기반으로 ERD를 추출, 물리적인 데이터베이스에서 E-R 추출.	
	입력물: Physical DB, Domain Knowledge.	출력물: ERD(Entity Relation Diagram).
원시코드 분석	(1)환경 분석 대상 원시소스 파일의 전체적인 개발 환경(언어, 개발 툴, 아키텍처, etc) 분석. (2)파일 구조 분석 대상 원시소스 파일의 전체적인 구조 파악. 각 파일간의 구조, 관계 정의 (3)파일 명세 파일단위, 모듈단위, 클래스단위, 함수단위, 변수 등의 정보를 케이스틀을 이용하여 명세화 한다.	
	입력물: Source Code.	출력물: File Description (Module, Class, Function, Variable).
시스템 모델링	(1)구조분석 원시코드 구조 분석 : 대상 원시코드를 Case Tool을 이용 분석. 원시코드 입력을 통해 출력되는 모듈, 클래스 구 조도를 파악. 인터페이스 분석 : 모듈(클래스)간 Calling / Called 호출관계 및 파라미터 이동 관계 분석. 구조와 호출관계를 중심으로 상호 인터페이스 분석. 다이어그램 정의 : 전단계의 산출물 정제, 명확한 구조도(Structure Chart), 클래스 다이어그램 (Class Diagram) 명세화. (2)기능 분석 모듈(클래스) 기능 분석 : 각 모듈(클래스) 내부의 함수 단위의 기능을 분석. 단계적 분할 및 집단화 : 소스 분석을 통하여 모듈의 기능 정리, 분석. 원시코드 분석 테이블을 작성, 유사 기능으로 분할 및 집단화. (3)관계분석 이벤트 시퀀스 정의 : 전단계의 산출물들과 명세화한 구조도를 기반으로 기능에 관한 이벤트의 흐름을 명세.	
	입력물: Source Code, File Description(Module, Class, Function, Variable). 출력물: File Description, Structure Chart, Event Flow Description, Class Diagram, Sequence Diagram.	

새로 생성된 산출물간의 일관성 및 변환의 영
향을 예측, 설정.

R0204, 측정된 산출물 세트(Artifacts Set)의 척도를
기준으로 순공학 단계(Phase)를 최소화하기
위한 분기 결정.

R0200을 통하여 검증, 정제되어진 산출물들은 수준에
따라 순공학 어플리케이션 개발 프로세스의 특정 단계
(Phase)를 생략, 개발 기간을 단축 가능케 함으로서 효
율적인 개발 수행을 지원한다.

3.3 RUP+re의 어플리케이션 개발 프로세스(Application Development Process)

본 논문에서 제시하는 재공학 프로세스 중 R0100,
R0200 워크플로우를 제외한 순공학 어플리케이션 개발
프로세스는 컴포넌트 개발을 고려하지 않은 일반적인
어플리케이션을 개발 혹은 기 개발되어진 재사용 단위
(예, Function, Module, Class) 및 상업용(COTS) 컴포
넌트를 수용하여 시스템을 개발하는 컴포넌트 기반 개
발을 목적으로 한다.

표 2 재공학 워크플로우의 세부 워크플로우와 액티비티

R0200	내용 및 목적	산출물
R0201 재공학 목표 수립	(1)재공학 목표 설정 역공학 산출물을 근거로 현대화(Modernization), 유지보수(Maintenance), 교체(Replace)등 재공학(Re Engineering) 수행 목표 및 수행 범위를 판단, 결정. (2)재공학 기법 설정 재공학이 가능하다면 재사용 수준 및 재공학 기법 정의. 역공학 산출물의 수준에 따라 소스 레벨, 디자인 레벨, 아키텍처 레벨 등의 다양한 재사용 수준 정의. 화이트박스(White Box), 블랙박스(Black Box) 혹은 래핑(Wrapping), 스크린 스크래핑(Screen Scraping), 마들웨어(Middleware), EAI(Enterprise Application Integration)등의 재공학 기법 정의.	재공학 계획서 (Re Engineering Plan)
R0202 역공학 산출물 검증	(1)산출물 정의 R0201의 목표 및 기법에 적합한 역공학 산출물 선별, 추출. 추가 산출물 정의. (2)산출물 검증 R0201의 목표 및 기법에 적합한 형태, 수준, 일관성 등을 검증. 목표에 부적합하다면 R0201 단계로 피드백.	
R0203 변환 영향 및 가설 설정	(1) 재공학 기법에 따른 변환 영향 요소 예측. 재공학 기법에 따른 시스템 전반에 미치는 영향 및 위험 요소를 파악, 기술. (2) 재공학 기법 결정에 따른 변환 가설 설정. 재공학 기법에 따른 시스템 및 산출물들의 변환 가설 및 위험을 추정.	
R0204 작업 단계 분기 결정	(1)분기 결정 R0201, R0202, R0203의 산출물을 근거로 어플리케이션 개발 프로세스의 특정 단계(Inception, Elaboration, Construction, Transition Phase)로 분기 판단. 재공학의 목표에 부합되는 산출물들은 특정 단계의 반복을 수행하는 산출물 세트(Artifacts Set)로 정의. 산출물 세트를 해당 워크플로우들로 분기 수행. 순공학의 단계를 최소화하는 것이 목표.	

본 논문의 어플리케이션 개발 프로세스는 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 어플리케이션 개발 프로세스는 기존의 RUP를 기반으로 재공학을 위한 기존 재사용 단위의 획득, 적용, 공급에 관한 세부 워크플로우 및 액티비티를 각 워크플로우에 추가, 정의하여 역공학, 재공학 워크플로우와의 효율적인 연계를 지원하였다.

그림 5와 같이 어플리케이션 개발 프로세스는 독립적으로 순공학에 적용되어지거나 재공학 프로세스의 일부 분으로 적용되어진다. 일반적인 순공학의 경우는 어플리케이션 개발(D0100, D0200, D0300, D0400, D0500, D0600, D0700) 경로를 통하여 프로세스가 전개 되어지고, 기존의 시스템의 자원을 활용하여 어플리케이션을 개발하는 경우 역공학, 재공학 워크플로우를 수행하고 진행되어지는 재공학(R0100, R0200, D0100, D0200, D0300, D0400, D0500, D0600, D0700) 경로를 통하여

프로세스가 전개 되어진다. 지원 워크플로우(Supporting Workflow)는 순공학과 재공학 모두를 지원한다.

표 3은 재공학 프로세스 RUP+re을 통하여 정의되는 주요 산출물이다. RUP의 주요 산출물들과 일관성을 최대한 유지 시켰으며 R0100을 통하여 정의되는 산출물들은 R0200의 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.

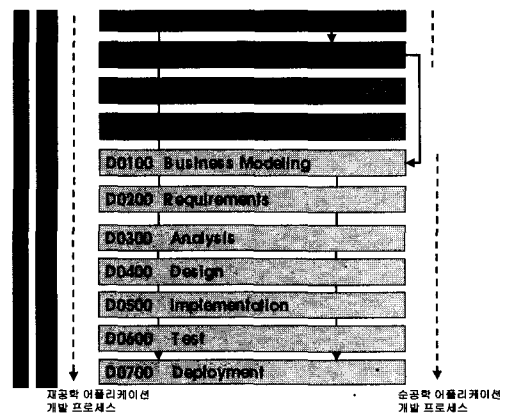


그림 5 역공학 프로세스

표 3 RUP+re 재공학 프로세스의 워크플로우별 산출물

워크플로우	Development Case.					
Environment	Development Case.					
Project Management	Software Development Plan, Iteration Plan.					
Legacy System	최종 산출물세트의 분기 워크플로우					
	Reverse Plan[0]	UI Structure Report[6] (1,2)	Data Flow Diagram[12] (6,7,8,9)	Business Process Use Case[16] (12,13)	[16].	Business
	Manual[1]	UI Function Report[7] (1,2,4)	Event Flow Description[13] (6,7,9)	Scenario[17] (12,13,15)	[17],[18].	Requirement
	Executable Code[2]	Data Dictionary[8] (2,3,4)	Entity Relation Diagram[14] (3,4,8)	Use Case[18] (16,17)	[6],[7],[8],[9],[11],[12],[13],[14],[15].	Analysis / Design
Data Base[3]	File Description[9] (4)	Sequence Diagram[15] (6,7,9,11)				
Source Code[4]	Structure Chart[10] (4)					
Domain Knowledge[5]	Class Diagram[11] (4)					
Re Engineering	Re Eng Plan.					
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.					
Configuration & Change Management	Configuration & Change Management Plan.					

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.

	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 70,000라인	약 140,000라인
문제점	개발 문서 없음	개발 문서 없음
요구 사항	유지보수를 위한 개발 문서의 추출 기능 개선 및 기능 향상을 위한 모듈의 추가 Web환경으로의 전환	유지보수를 위한 개발 문서의 추출 C/S환경에서 EJB컴포넌트 기반 환경으로의 전환
비고	Visual Basic에서 권장하는 방식의 코딩 룰을 적용하지 않았다.	UI설계를 표준 MFC컨트롤을 사용하지 않았음. DB를 파일시스템을 사용하였다.

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

• R0100 (Workflow:Reverse Engineering)

R0103 (Detailed Workflow:기존 정보 검증 및 명세화)

R0103-Activity1: 메뉴얼 및 UI 분석

UI 구조 분석 및 UI 기능 분석은 사용자 메뉴얼 및 UI실행을 통하여 각 UI간의 구조 파악, 네비게이션 트리 작성, 각 UI의 기능 파악, 각 기능에 따른 이벤트의 흐름을 기술한다. Visual Basic 언어의 특성상 UI관련 구조와 기능 분석은 수월하게 진행되었다.

R0103-Activity2: 데이터 분석

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

R103-Activity3: 원시 코드 분석

자동화 역공학 틀인 McCabe를 이용하여 기본적인

용하여 슬라이싱 및 클러스터링을 수행 하였다. 주어진 원시코드에 대해 기능별로 재사용 가능한 모듈로 클러스터링 및 수작업에 의해 비 구조화된 모듈을 재구성하였다.

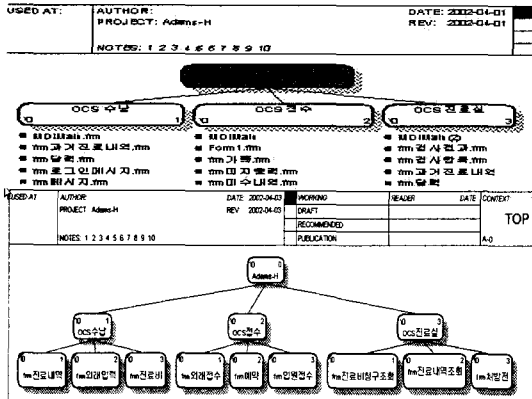


그림 9 Structure Chart

• R0200 (Workflow:Re Engineering)

Detailed Workflow:R0201, R0202, R0203, R0204

사례 A의 R0100를 통하여 추출된 산출물과 각 산출물간의 연관 관계는 그림 10과 같으며 각각의 산출물은 R0200의 R0201, R0202, R0203 세부 워크플로우를 통하여 화이트 박스형태의 재구성을 통한 현대화(Modernization)가 가능하였다, 그림 11에서와 같이 R0204를 통하여 어플리케이션 개발 워크플로우의 구현 단계(Construction Phase)의 산출물 세트(Artifacts set)으로 전환, 각각 해당 워크플로우로 분기되었다.

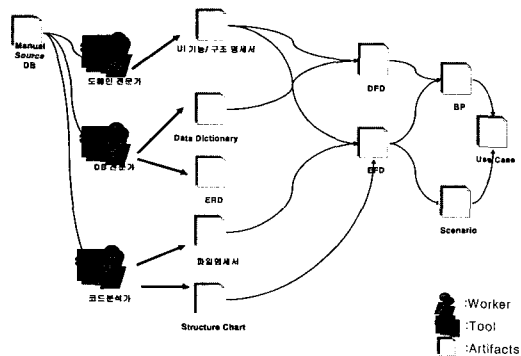


그림 10 R0100을 통하여 실제 추출된 사례A의 산출물의 연관도

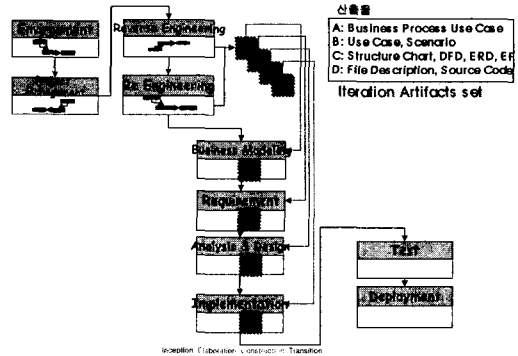


그림 11 R0200을 통하여 프로세스 단계 분기한 산출물 세트

4.2 평가

사례 연구를 통해 본 연구 결과를 정리해 보면 다음 표 4와 같다.

표 4의 사례A의 경우 재공학 요구사항을 반영하기 위하여 R0100단계 산출물을 효과적으로 추출 가능하였고 추출된 산출물의 일관성 및 품질 수준이 재사용 요구사항을 만족하였다. R0200를 통하여 역공학 산출물을 검증, 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 Web환경으로 변환하기 위하여 Visual Basic원시코드의 변환에 자동화 툴을 이용한 코드수준의 재사용이 90%이상 가능하였다.

사례B의 경우 개발환경의 문제로 인하여 R0100의 산출물간의 일관성을 유지할 수 없었으며 품질 수준이 재공학 목적을 충족시키지 못하고 R0200단계를 통해 Inception단계로 분기가 가능하였으나 추가적인 요구사항을 반영하기 위하여 도메인 분석이 재 수행되는 등 R0100단계의 역공학 산출물이 D0100, D0200, D0300를 통하여 정제(Refine) 및 재개발(Re-Development)되어야 하였다. 또한 EJB기반의 컴포넌트 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 R0100을 통하여 추출되는 산출물의 수준은 레거시 시스템의 현재 상태와 그것을 역공학 하기 위해 사용 가능한 자동화 도구가 주요 변수가 되었다.

표 4 산출물 재사용 사례 결과표

입력물	최종 산출물	자동화 작업 도구	재사용 산출물 및 재사용을 평가		결론
사례 A Manual. Executable Code. Data Base. Source Code. Domain Knowledge.	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	최적 최적 최적 최적 최적 적합 최적 최적	역공학 워크플로우를 통하여 효율적인 재 사용 산출물 추출 가 능. 제공학 워크플로우를 통하여 효율적인 재 사용 목표 설정 및 산 출물 분기 가능. 역공학, 제공학 워크 플로우를 통한 순공 학과의 효율적인 연 계지원, 순공학의 특 정 단계(Phase) 생략 가능.
			Use Case.		
			Scenario.		
			ERD.		
			Structure Chart.		
			DFD.		
			Source Level Reuse		
			Design Level Reuse		
			산출물 수준 및 일관성		
			산출물 분기 단계		
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	보통 보통 보통 부적합 보통 보통 불가 부적합 보통	
			Use Case.		
			Scenario.		
			ERD.		
			Class Diagram.		
			Sequence Diagram.		
			Source Level Reuse		
			Design Level Reuse		
			산출물 수준 및 일관성		
			산출물 분기 단계		
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)					

R0100 워크플로우의 문제점으로는 첫째, 개발된 원시 코드 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 높아질수록 산출물간의 일관성을 검증하는데 어려움이 따랐다.

R0200 워크플로우의 문제점으로는 첫째, 제공학 할 것인지, 새로운 시스템으로 교체 할 것인지 여부의 결정을 하기 위한 정형적인 비용 산정 방법이 없어 휴리스틱하게 수행하였고 둘째, 제공학을 수행하기로 결정된 경우 산출물의 품질측정, 일관성 검증, 프로세스로 분기 결정에 정형적인 방법이 부족하였다.

5. 결론

본 논문에서는 제시하는 제공학 프로세스인 RUP+re는 역공학 워크플로우, 제공학 워크플로우, 순공학 어플리케이션 개발 프로세스로 구성, RUP를 제공학을 수용할 수 있게 확장, 커스터마이징 하여 수용하였다.

역공학 워크플로우에서는 역공학을 통해 추출해야 하는 산출물 정의, 산출물간의 연관성 및 일관성을 통하여

추상화된 이차 산출물을 추출할 수 있는 효율적인 추출 방법을 제시 하였다.

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

순공학 프로세스인 어플리케이션 개발 프로세스는 역공학, 제공학 워크플로우와의 연계를 위하여 RUP를 커스터마이징 하여 포함하였으므로 단순한 순공학 프로세스로 혹은 레거시 시스템 재사용을 위한 제공학 프로세스로 각각 지원 가능하다. 또한 기존의 기술들이 역공학, 순공학, 혹은 제공학 등 서로 분류되어 상호 연계성 및 효율성을 찾지 못하는데 반하여 역공학, 제공학 워크플로우를 순공학 워크플로우들과 일련의 제공학 프로세스로 통합, RUP+re를 통한 상호 기술간의 효율적 연계로 제공학 프로세스의 효율성을 최대화 하였다.

참고 문헌

[1] Rene R.Klsch, *Reverse Engineering: Why and How to Reverse Engineer Software*, Proceedings of the California Software Symposium, 1996.
[2] Pentead, R. Masiro, P.C. Cagnin, M.I. An

- Experiment of Legacy Code Segmentation of Improve Maintainability.* CSMR'99, Amsterdan. Proceedings. IEEE,p, 111-119, 1999.
- [3] SEI Reengineering Center, *Perspectives of Legacy System Reengineering*, 1995.
- [4] Dolly M. Neumann, *Evolution Process for Legacy System Transformation*, ACM, 1998.
- [5] Gerald C. Gannod, *PACKRAT A Software Re-engineering 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] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Software Development Process*. 1999. The Addison-Wesley Object Technology Series.
- [8] D'Souza and Wills, "Objects, Components, and Framework with UML," 1999.The Addison-Wesley Object Technology Series.
- [9] Sterling Software, "Advisor 2.0," 1998.
- [10] "마르미 방법론", 한국 전자 통신 연구원, at URL: <http://www.component.or.kr>
- [11] Louise Scott,Ross Jeffery, Lucila Carvaho, John D' Ambra, *Practical Software Process Improvement-The IMPACT Project*, IEEE, 2001.
- [12] at URL : <http://www.systemtransformation.com/TMethodology.htm>
- [13] 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, September 29-October 3, 1997, pp. 150-159
- [14] 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.



류 성 열

1997년 2월 아주대학교 컴퓨터학부(공학 박사). 1997년 3월 ~ 1998년 3월 George Mason University 교환교수. 1981년 3월 ~ 현재 숭실대학교 정보과학대학 컴퓨터 학부 교수. 1998년 3월 ~ 2001년 2월 숭실대학교 정보과학대학원 원장. 1998년 3월 ~ 현재 숭실대학교 전자계산원 원장. 관심분야는 리엔지니어링, 소프트웨어 유지보수, 소프트웨어 재사용, 소프트웨어 재공학/역공학



최 일 우

1995년 숭실대학교 전자계산학과 학사(공학사). 1997년 숭실대학교 컴퓨터학과 석사(공학석사). 1997년 ~ 현재 숭실대학교 컴퓨터학과 박사 과정. 관심분야는 개발 방법론, 유지보수, 역공학, 재공학, 재사용, 컴포넌트 기반 소프트웨어 공학