

컴포넌트 기반의 체계적인 재공학 프로세스

차 정 은[†] · 김 철 홍^{††} · 양 영 종^{†††}

요 약

소프트웨어(S/W) 재공학은 S/W의 생명주기의 연장을 통한 지속적인 비즈니스 가치 창출 및 궁극적인 S/W ROI(Return on Investment) 확대에 가장 효과적인 기술 중 하나이다. 그럼에도 불구하고, S/W 재공학은 비용 소모적이며, 그 효과 역시 미흡한 어려운 작업으로 인식되어 왔다. 사실, 빈번히 발생하는 유지보수 요구에 대해 레거시 시스템들을 일치성 없이 그때그때 확장, 수정함으로써, 기존 시스템 본연의 의도를 상실 시켜 S/W시스템 아키텍처가 존재하지 않는 난잡한 시스템으로 전환시키는 경우가 종종 발생하고 있다. 더욱이 급격히 변하는 시스템 환경과 복잡 다양해지는 고객의 요구를 충족시킬 수 있는 새로운 S/W 시스템들을 매번 적시에(Time-to-Market) 제공하기는 거의 불가능하다. 따라서, 새로운 IT 기술의 출현과 비즈니스 정보 모델의 다양한 변경, 시스템 처리 로직의 급격한 복잡성 증가 등의 변화에 적절히 대처하기 위해서는 조직의 주요 자산으로서 레거시 시스템의 활용을 극대화할 수 있는 체계적인 재공학이 요구된다. 그러므로 본 논문에서는 레거시 시스템들의 가치를 극대화할 수 있는 체계적인 재공학 방법론 제공을 목적으로, 재공학의 초기 계획 단계에서부터 역공학 과정과 컴포넌트 변환 단계를 포함하는 재공학 프로세스와 그에 따른 구체적인 작업과 기법 및 산출물들을 명시한 레거시 시스템의 컴포넌트화 프로세스인 마르미-RE를 제안하고 간단한 사례적용 과정을 제시한다.

키워드 : 레거시 시스템, 재공학 프로세스, 역공학, 컴포넌트 변환

Component-Based Systematic Reengineering Process

Cha, Jung-Eun[†] · Kim, Chul Hong^{††} · Yang, Young-Jong^{†††}

ABSTRACT

Software(S/W) reengineering is one of the effective technologies to produce a business worth and expand the S/W ROI continuously. In spite of, S/W reengineering has been recognized a cost-consumptive works with inefficient productivity. In fact, we have used to transform to confusion system with destructive system architecture by extending and updating legacy system in a temporary expedients. Moreover it is impossible to provide the time-market products for coping with rapid changeable system environment and meeting to complicated customer's requirements. Therefore, we need a systematic reengineering methodology to fulfill the changeable environment, as appearance of new IT techniques, various alteration of business information model, and increment of business logic. Legacy systems can be utilized as the core property in business organization through reengineering methodology.

In this paper, we target to establish the reengineering process, proposed MaRMI-RE consisting of initial Planning phase, reverse engineering and component transformation phase. To describe the MaRMI-RE, we presented the concrete tasks and techniques and artifacts per individual phase in process, and the case study is showed briefly.

Key Words : Legacy System, Reengineering Process, Component Transformation

1. 연구 배경

레거시 시스템(Legacy System)은 과거 그 당시에 설정된 목표를 위해 과거의 기술을 이용하여 개발되었지만 현재에도 여전히 중요한 비즈니스 로직을 포함하고 있어 다른 시스템으로 대체하기에는 위험 부담이 큰 S/W 시스템을 말한다[1]. 급변하는 IT 기술의 출현과 비즈니스 정보 모델의 다

양한 변경, 시스템 처리 로직의 복잡성 증가는 기존 시스템들의 잦은 유지보수에서 발생하는 기술적, 비용적 한계를 드러내고 있다[2].

레거시 시스템은 조직의 핵심 가치를 지닌 재사용 자산으로 활용할 수 있는 의미 있는 자산이다. 따라서, 레거시 시스템의 가치를 새로운 기술적, 환경적 요구에 맞도록 지속적으로 확장, 발전시키기 위해서는 체계적인 아키텍처를 갖는 새로운 타겟 시스템으로의 재공학이 필요하다. 이를 통해서만 시스템의 이해성과 재사용성을 높이며, 융통성 있는 유지보수 구조를 구축할 수 있어 향후의 시스템 진화 모델을 수용할 수 있다. 특히, 개인 및 조직의 정보 공유 매체일

[†] 정 회 원 : ETRI 임베디드 S/W연구단 S/W공학팀 선임연구원

^{††} 정 회 원 : 임베디드 S/W연구단 S/W공학팀

^{†††} 정 회 원 : ETRI 임베디드 S/W연구단 S/W공학팀 팀장(책임연구원)

논문접수 : 2005년 9월 16일, 심사완료 : 2005년 12월 14일

뿐만 아니라 업무 처리를 위한 핵심 기술로 인터넷이 보편화 되고 이미 개발 완료된 실행 가능한 독립 부품인 컴포넌트 기반 개발(CBD : Component Based Development)이 지배적인 S/W 개발 패러다임으로 확대되어감에 따라, 더 좋은 설계 구조와 아키텍처를 갖는 확장성 있는 시스템으로의 재공학에 대한 당위성은 더욱 강화되고 있다. 대부분의 레거시 시스템들은 표준화와 개방성의 결여나 변경의 어려움, 분산 아키텍처 결여 등의 원인으로 새로운 기술을 수용하거나 복잡한 비즈니스 요구를 확장 시키기에는 많은 문제점을 가지고 있다[3]. 따라서, 대규모 레거시 시스템을 컴포넌트 기반 시스템으로 체계적으로 변환하고 통합하기 위한 프로세스와 기법을 제공할 수 있는 재공학 프로세스가 절실히 필요하다.

본 논문에서는 컴포넌트 기반 시스템으로의 체계적인 변환과 통합을 수행하기 위한 프로세스와 기법을 제공하는 마르미-RE에 대해 설명한다. 현재, 마르미-RE가 객관적인 근거가 마련된 절차와 기법을 가지지 못한 것은 사실이나, 5개의 재공학 관련 기관들이 직접 협력하여 실사용자들의 요구에 맞는 프로세스를 도출하고 이를 다수 업체의 사례 적용을 통해 결과물들을 생성하였다. 따라서 증명화된 근거를 갖지 못할 지라도, 재공학 전문가들의 실제적인 노하우가 반영되어 그 활용에 대한 타당성을 확보하고 있다고 사료된다.

마르미-RE는 레거시 시스템의 AS-IS 모델을 타겟 시스템의 TO-BE 모델로 변환하기 위한 컴포넌트 기반의 재공학 프로세스로, 계획, 역공학, 컴포넌트 그리고 인도 단계로 구성된다. 현재, 1.0 버전이 완성되어진 상태이며, 실제 기업의 재공학 프로젝트에 사례 적용이 이루어진 상태이다. 본 논문 전체에 걸쳐 마르미-RE의 개념 및 구성하고 있는 작업 절차에 대하여 요약하여 서술한다.

2. 기본 개념

2.1 선행 연구

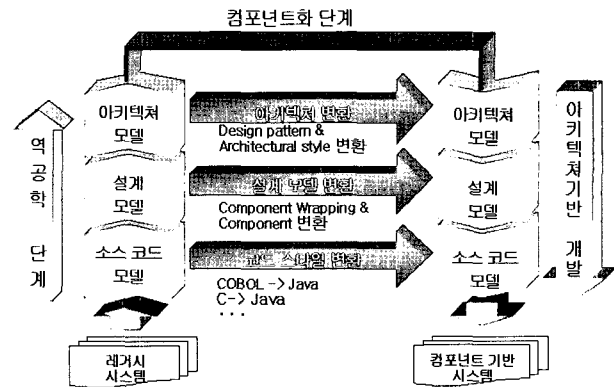
널리 참조되는 재공학 방법론으로 CMU SEI의 CORUM II(Common Object-based Reengineering Unified Model II)[4]가 있다. 이 방법은 아키텍처와 코드 기반 재공학 도구들을 통합하는 프레임워크 구축을 목적으로 재공학과 순공학의 통합 모형을 제시한다. 그러나 코드 중심의 재공학 프레임워크 개발에 초점을 뒀으로써 제한된 개발 환경을 가지며, 레거시 시스템의 분석에서 컴포넌트 구현에 이르는 체계적인 재공학 프로세스를 제시하지 못한다. 또한 Georgia Institute of Technology에서 Mosaic 웹 브라우저에 새로운 요구사항 반영을 통해 레거시 시스템 개선을 위해 개발한 MORALE(Mission Oriented Architecture Legacy Evolution)[5]은 기술적 요소보다 조직의 미션(Mission)에 중점을 두어 레거시 시스템의 프로세스 향상을 목표로 한다. 따라서 재공학 실무자가 재공학 프로젝트 수행에 필요한 구체적인 작업의 절차나 기법들이 제시가 미흡하다.

마르미-RE와 같이 아키텍처 중심의 재공학에 관한 연구

로 [6-9] 등이 있다. 그러나 이들 연구들도 개발자의 설계 결정이 코드로 매핑하는 동안 발생하는 정보 손실 및 변형을 체계화된 작업 절차나 산출물의 연계를 통해 극복하지 못하고 있어 정확한 재공학 비전과 전략을 준비하지 않는 한 재공학 프로젝트를 완성하기가 거의 불가능하다. 특히, 재공학자들이 조직의 의도에 맞게 재공학 절차와 기법을 선택하거나 반복적으로 수행할 수 있는 체계와 표준 지침을 갖추지 못하고 있어, 중요한 결정적 선택에서는 개발자의 주관적인 의사 결정에 의존하고 있는 실정이다. 뿐만 아니라, 레거시 시스템의 핵심 요소에 대한 공통 자원 확보하거나, 방법론 차원에서 재공학의 절차와 기법을 구체적으로 정의하고자 하는 노력은 미흡하여 많은 기업들이 재공학 프로젝트를 추진함에 있어서 시행 착오를 겪고 있다. 최근에는 패턴, 프레임워크, 컴포넌트 등을 포함하는 아키텍처 기반의 재공학 기술을 통해 이러한 한계점을 극복하고자 시도하고 있으나 본 논문에서와 같이 재공학을 위한 구체적인 절차나 기법들의 제시가 특정 영역에 매우 제한적이다[10]. 또한 전형적인 재공학 지원 도구 및 기법들은 대부분 소스 코드에 기반을 두고 데이터 흐름이나 제어 흐름을 분석하여 매트릭을 제공하는 정적 분석 및 재문서화 기법의 연구에 치중하였다. 따라서, 재공학의 전략적 계획을 확립하고 타겟 시스템에 적합한 아키텍처를 확보할 수 있는 체계적인 전개 방식을 전혀 지원하지 못한다. 최근 상업화된 국내의 재공학 도구인 RESORT[11], LegaMod[12] 등은 레거시 시스템의 코드를 분석하여 여러 가지 관점의 정보를 생성하고, 새로운 요구에 맞는 언어로 변환하거나 To-Be 모델 도출을 지원하는 기능을 제공한다. 그러나 이들 역시 코드 변환 중심의 제한된 언어와 To-Be 모델만을 지원함으로써 다양한 변환을 지원할 수 있는 기법과 프로세스를 제공하지 못한다.

2.2 기본 모형

마르미-RE는 앞에서 설명한 CORUM II의 “Horse Shoes” 메타 모델에 근간을 두고 개발되었다. 즉, 레거시 시스템의 정보를 분석하고 이해하여 보다 가치 있는 의미로 추상화하는 활동들로 구성되는 역공학 프로세스와, 각 추상화 수준에 따라 정보의 형상들을 동일 수준의 다른 형상으로 변



(그림 1) 마르미-RE의 개념 모델

환하는 컴포넌트화 프로세스, 그리고 새로운 컴포넌트 기반 시스템으로 순공학적 전개를 수행하는 아키텍처 기반 개발 프로세스로 구성되었다.

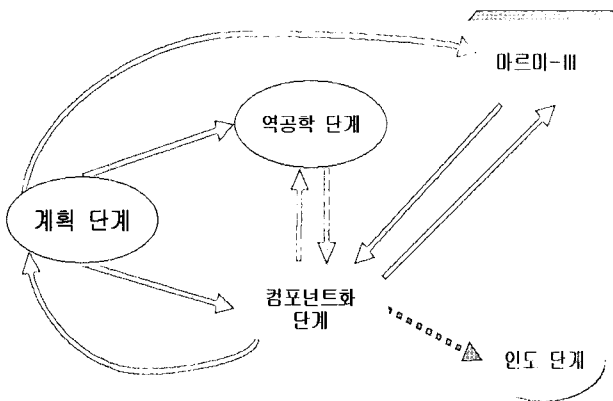
(그림 1)는 본 프로세스의 개념 모델을 보여준다. 역공학 단계는 최하위 수준의 소스 모델로부터 기능 모델, 아키텍처 모델로 추상화된 정보를 복구한다. 소스 모델은 원시 코드를 분석하여 텍스트 및 AST 정보를 생성하여 레거시 시스템의 코드 패턴을 식별한 것이다. 프로세스 모델은 추상화된 설계 정보인 시스템과 데이터의 구조도 및 단일 로직을 수행하는 워크플로우이며, 아키텍처 모델은 기능 모델을 그룹핑하고 필터링하는 과정을 거쳐 서브시스템과 그들간의 인터페이스 관계를 표현한다.

컴포넌트화 단계는 역공학의 각 수준에서 조직내의 역량에 따른 변환 프로세스를 지원한다. 가장 낮은 단계의 변환인 코드 스타일 변환은 단순히 프로그램 언어간의 변환을 지원하며, 기능 모델 변환은 랩핑과 DB 스키마 변환이 전형적인 사례이며, 아키텍처 변환은 복구된 레거시 시스템 아키텍처를 새로운 아키텍처로 변환하는 프로세스이다. 특히 컴포넌트 기반 개발은 마르미-III와 같은 순공학 방법론으로 각 변환 단계에서 마르미-III와 연계될 수 있는 방안을 제시한다.

마르미-RE는 기존의 방법론과 같이 순차적이거나, 동기화된 전개 프로세스가 아니라, 병렬적이고 선택적으로 재공학 프로세스를 맞춤 구성할 수 있도록 정의 하였다. CBD 지향의 아키텍처 기반 개발 프로세스를 제공함으로써, 다갯 아키텍처 중심으로 계속적인 확장과 조립, 맞춤을 지원하는 프로세스이다.

마르미-RE프로세스 전개는 (그림 2)와 같은 기본 모델에 기초한다.

즉, 계획 단계에서 설립된 재공학 전략과 그에 따른 프로세스에 따라 역공학 단계를 거친 후 컴포넌트 단계로 진행하거나, 바로 컴포넌트 단계로 진행 한 후, 필요 시 필요한 정보들을 역공학 단계를 수행함으로써 획득한다. 또한 컴포넌트 단계와 인도 단계는 마르미-III의 활동과 작업들을 참조로 하여 필요한 산출물들을 생산하며 순공학적 전개를 진행 시킨다.



(그림 2) 마르미-RE의 기본 프로세스 전개

3. 상세 절차 및 주요 산출물

3.1 프로세스 메타 모델

재공학 방법론은 사용자(즉, 재공학 방법론 개발자 및 활용을 원하는 고객) 요구 및 환경적 조건 분석이 매우 중요하며 재공학의 목표와 전략 등은 고객의 상황에 따라 다르게 적용되므로, 지속적인 유지보수 및 진화 계획에 효과적으로 대처할 수 있어야 한다. 따라서, 사용자 요구가 확정될 때까지 사용자들 간에 의사를 전달할 수 있는 절차가 확보되어야 하며, 환경적, 기능적 변화를 수용할 수 있는 피드백 및 반복적인 단계 진행의 보장은 필수적이다. 또한 재공학의 기술적 전략 선택에서 사용자 특성이 충분히 반영된 최상의 결정이 가능하도록 가능성 있는 후보들을 제시하고 선택할 수 있는 절차와 기법 등을 제공해야 한다.

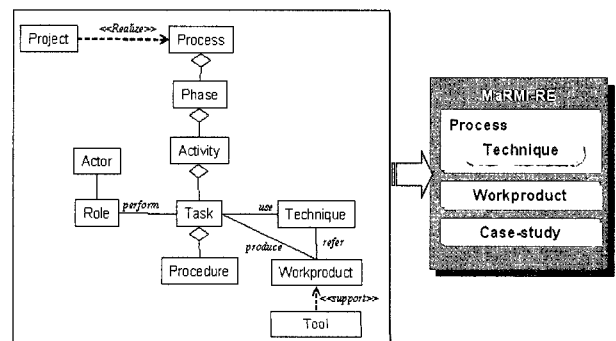
(그림 3)은 마르미-RE를 구성하는 프로세스 요소들의 메타 모델과 이를 중심으로 만들어진 마르미-RE의 결과물을 나타낸다. 재공학 프로젝트(Project)는 프로세스(Process)를 통해 실체화될 수 있으며, 프로세스는 재공학 공정의 논리적 구분인 여러 개의 단계(Phase)를 포함한다. 또한 단계는 재공학의 특정 목적 달성을 위해 체계화된 집합체인 여러 활동(Activity)을 포함하고 있으며 각 활동은 선택적인 작업(Task)을 포함한다. 작업은 다시 더욱 상세한 절차(Procedure)들과 해당 작업의 효과적 수행에 필요한 방법인 기법(Technique), 프로젝트 참가자들의 구체적인 역할(Role) 및 작업의 결과로 생산되는 산출물(Work product)로 구성된다[13].

3.2 전체 활동 구성

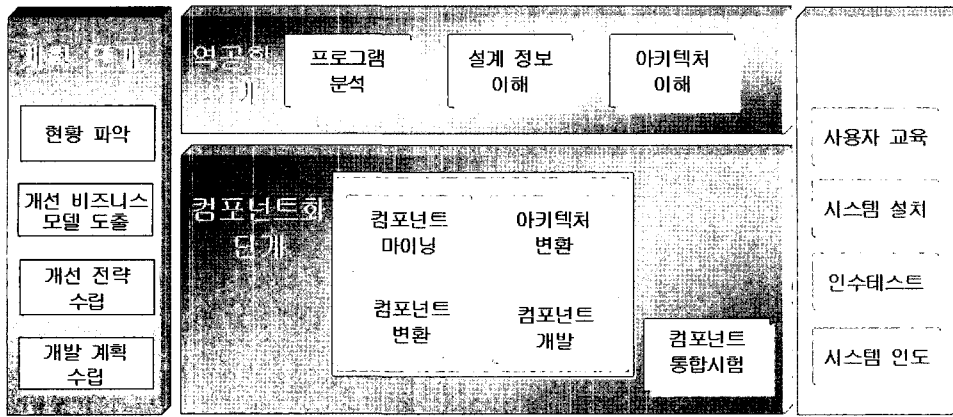
마르미-RE는 (그림 4)에서 보는 것과 같이 계획 단계, 역공학 단계, 컴포넌트화 단계, 인도 단계 등 4 단계로 구성되어 있으며, 각 단계는 메타 모델에 따라 세부 활동들과 그 활동들을 구성하는 다수들 개의 작업들로 구성된다[14, 15].

(1) 계획 단계

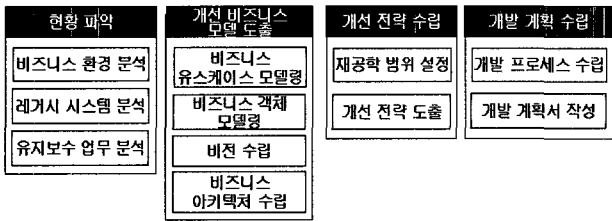
이 단계는 레거시 시스템의 전반적인 분석을 현상황의 문제점 및 재공학에 대한 기대 요구들을 파악하고, 기대되는 품질과 생산성 향상 여부에 대한 심도 있는 분석과 예측을



(그림 3) 마르미-RE의 메타 모델



(그림 4) 마르미-RE의 프로세스 구성도



(그림 5) 계획 단계의 활동과 작업

통해 효과적인 재공학 프로젝트 수행을 위한 프로젝트의 수행 목표와 범위 결정한다. 또한 컴포넌트화 전략과 적절한 개선 방향 및 전개 프로세스를 제시한다. 이러한 관점에서 본 단계는 (그림 5)와 같다.

4개의 활동과 11개의 작업으로 구성되며, 먼저 현황 파악 활동에서는 비즈니스 관점에서의 조직의 내부 이슈 및 문제점을 도출하고, 레거시 시스템의 기능을 업무 단위로 분석하며, 유지보수 상의 문제점이나 개선의 여지가 있는 부분들을 발견하여 개선 사항들을 협의한다.

두 번째 개선 비즈니스 모델 도출은 레거시 시스템의 업무에 대하여 이상적인 모델을 제시하고 이를 실현하기 위해 필요한 유스케이스의 객체를 모델링을 통해 이해 당사자들의 요구사항을 명확히 파악하며 나아가 프로젝트의 목표와 범위 및 비즈니스 유스케이스의 우선순위를 이해하고 수립함으로써 분산된 응용들의 구성 요소들 간의 관계를 정의하여 개선 아키텍처, 시스템 아키텍처, 소프트웨어 아키텍처 등의 3가지 관점에서 목표 아키텍처를 수립한다.

개선 전략 수립 활동은 개선 대상 업무의 비즈니스 가치 및 시스템 측면의 기술적 요소를 파악하여 재공학의 우선순위를 결정하고, 각 업무 단위별로 컴포넌트화를 위한 최적의 변환 전략을 수립한다.

이를 위해 (그림 6)과 (그림 7)과 같은 산출물을 작성한다. 비즈니스적 요소와 시스템적 요소에 대해 항목 별 상대적 가중치(W:Weight) 부여하고, 각 개선 업무들에 대해 할당 값(AV:Assignment Value)을 위험도, 기술적 난이도 등과 같이 재공학의 필요성을 낮추는 요인 및 유지보수 비용 등 재공학의 필요성을 높이는 긍정적인 요인들에 대해서는

각각 (-5~0) 범위 값 및 (0~+5)의 값을 할당한다. 여기서 각 기술적 요소와 비즈니스 요소 사이의 가중치 비율은 재공학을 통해 획득하고자 하는 조직의 목표에 따라 융통성 있게 조정한다. 그러나, 동일 재공학 프로젝트를 통해 획득하고자 하는 조직의 비전을 단일화시킴으로써 재공학의 위험요소를 최소화하며 점증적인 변환 프로세스의 효과를 극대화시킬 수 있다. 따라서 하나의 재공학 프로젝트는 하나의 최종 목적을 설정하고 여기에 맞게 공정한 재공학 업무 선정을 위해 고정된 요소 가중치를 할당하는 것이 바람직하다.

이어 추정값(EV: Estimated Value)을 계산하여 업무 별 평가치 및 평균 평가치를 산출하여 업무 별 평가치가 평균 평가치 보다 큰 경우, 재공학 대상으로 식별한다. 추정값은 조직의 비전에 따른 요소별 가중치와 업무의 특성에 따른 할당 값에 따라 결정되며 업무별 재공학 우선 순위 비교를 위해 백분율(%)로 나타낸다. 즉, 추정값은 각 업무에 재공학 요구를 비교하기 위한 수치로서, 그 값이 높을수록 재공학의 필요성이 높음을 표현하고 낮을수록 재공학의 필요성이 적음을 의미하는 수단으로 이용된다.

요소	비즈니스적 요소(Bn)			시스템적 요소(Sn)			합계	우선 순위	
	가치	위험	확장성	품질	유지보수 비용	기술적 난이도			
업무	30%	10%	10%	20%	20%	10%	100%		
A	할당값(AV)	+2	-1	+2	+1	-1	-3	40%	X
	추정값(EV)	60%	-10%	20%	20%	-20%	-30%		
B	할당값	5	-3	3	4	-1	-4	170%	1
	추정값	150%	-30%	30%	80%	-20%	-40%		

(그림 6) 재공학 우선 순위 결정의 예

예제에서 A, B 두 업무에 대한 평균 평가치가 (210/2=105%)이므로, 업무 별 평가치가 105% 보다 큰 B업무를 재공학 대상으로 결정되며, 재공학 대상으로 선정된 업무들 가운데, 업무 별 평가치가 큰 우선 순위를 가지는 B업무가 1순위가 된다.

다음은 재공학 대상 선정을 위해 적용한 공식이다

$$\text{추정값(EV)} = \text{요소 별 가중치(W)} * \text{할당 값(AV)}$$

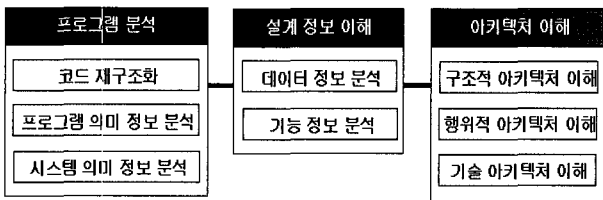
$$\text{업무 별 평가치} = \sum_{i=1}^n EV_i$$

$$\text{평균 평가치} = \frac{\sum EV_i}{\text{총업무의수}}$$

또한 재공학 전략을 결정하기 위해서는 (그림 7)의 예제에서 보는 것과 같이 업무에 대해 변환(T) 및 랩핑(W)의 기준 항목을 작성하고, 각 항목 별 할당값(1~5 사이)을 결정하고 그 값을 (T)와 (W)에 대해 각각 합하고 비교함으로써 완성한다.

비즈니스 유스케이스 별 고려 사항	가중치 (%)	A업무		B업무	
		할당값 (0-5)	평가치	할당값 (0-5)	평가치
T1	해당 업무에 대한 지속적 요구 사항의 변경과 추가가 일어날 가능성이 있는가	20	4	80	
T2	해당 업무의 방대한 처리로 인해 지속적 시스템 투자가 발생하는가	30	3	90	
T3	동일한 업무를 다양한 클라이언트 플랫폼으로 서비스할 필요가 있는가	5	3	15	
T4	해당 업무를 구현한 모듈의 품질이 만족스럽지 못한가	5	3	15	
T5	해당 업무를 구현한 모듈의 유지 보수 기술을 보유한 인력은 풍부한가	10	2	20	
T6	해당 업무는 비즈니스 적으로 가치가 높고 앞으로 가치가 지속성을 갖는가	30	4	120	
변환 전략 총평가치		100%	340%		
W1	비즈니스적으로 위험도가 높은가	20	3	60	
W2	요구 사항의 변경 및 추가로 인해 거의 영향을 받지 않는가	10	2	20	
W3	업무를 구현한 모듈의 규모가 너무 크고 관련 모듈들과의 결합도로 인하여 문법이 어려운가	30	4	120	
W4	기존 모듈의 구현 내용을 쉽게 이해할 수 없거나 이해를 지원할 인력 및 산출물이 없는가	10	4	40	
W5	내부 처리 로직이 복잡한가	15	3	45	
W6	이관해야 하는 데이터의 양이 방대하며 범위를 명확하게 한정 지을 수 없는가	15	3	45	
랩핑 전략 총평가치		100%	330%		
컴포넌트화 전략		T(변환)			

(그림 7) 개선 전략 수립의 예



(그림 8) 역공학 단계의 활동과 작업

마지막 활동인 개발계획 활동은 사용자의 요구 및 개발의 특성에 따라, 선택하고, 조정하여 최적의 개발 프로세스를 수립하고, 프로젝트 기간 내에 효율적으로 목표를 달성하기 위해 업무 목록, 업무 수행 방법 등을 구체화한 개발 계획서를 작성한다.

(2) 역공학 단계

레거시 시스템의 산출물들의 분석을 통해, 레거시 시스템의 주요 기능 및 구성 요소를 파악하며 소스 코드 분석과 설계 정보 모델링을 통해 레거시 시스템의 정적, 동적, 행위 정보 복구한다. 즉, 코드의 의미 분석 결과들을 설계 정보로 추상화 하는 모델링 작업이 수행함으로써 아키텍처 정보를 이해하고 추상화 시킴으로써, 컴포넌트화 단계의 활동들을 위한 준비 작업을 수행한다.

(그림 8)에서 보는 것과 같이 3개의 활동과 8개의 작업으로 구성된다.

프로그램 분석 활동은 레거시 시스템의 전형적인 역공학 프로세스를 나타낸다. 따라서, 코드 재구조화와 원시 코드 분석을 통해 레거시 프로그램의 구문 정보 및 의미 정보를 시스템 수준과 단위 프로그램 수준에서 분석하고 추출하며, 레거시 코드 패턴을 분석한다. 또한 레거시 코드로부터 서브루틴 호출 및 제어 그래프, 변수 관계도, 시스템 자원 그래프, 프로그램 호출 그래프와 화면 흐름 그래프 등을 산출해냄으로써 분석된 정보들을 정규화 한다. 이 작업에서는 자동화된 도구의 사용을 통해 효율성을 높을 수 있다.

두 번째로 설계 정보 이해 활동은 앞 단계의 프로그램 분석 정보들을 토대로 기능적인 단위 프로세스들을 식별하고 이들간의 제어 흐름과, 관련된 테이블과의 데이터 흐름을 명확히 하며, 다음 활동인 아키텍처 이해를 위한 시스템 설계 정보를 제공한다. 이는 레거시 시스템에 대한 설계 정보를 모델링 하고 구조적인 다이어그램으로 추상화 시킴으로써 더 높은 이해를 획득하기 위한 것이다. 이를 위해 개체 관계도, 데이터베이스 스키마 및 계획 단계에서 이상적인 모델 상에서 추출된 유스케이스와 실제 레거시 시스템의 유스케이스 사이의 대응표 등이 산출물로 생성된다.

아키텍처 이해 활동은 레거시 시스템을 구성하고 있는 모듈들을 보다 큰 단위의 서브 시스템 단위로 추상화 하여 이들 간의 의존 관계가 어떻게 구성되어 있는가를 파악하는 구조적 아키텍처와, 구조적 아키텍처를 구성하는 서브 시스템들을 바탕으로 서브 시스템들 간의 호출 관계가 어떻게 이루어지는 지를 이해하기 위해 행위적 아키텍처를 생성한다, 그리고 레거시 시스템을 구성하는 하드웨어 환경과 각각의 서브 시스템들을 구현하기 위해 사용된 기술 등을 파악하는 작업들을 수행한다.



(그림 9) 컴포넌트화 단계의 활동 및 작업 구성도

(3) 컴포넌트화 단계

이 단계는 레거시 시스템에서 역공학 과정을 통해 추출된

정보를 바탕으로 의미적으로 연관성이 높은 시스템 구성 요소(entities)들을 컴포넌트화 하기 위해 관련성 높은 부분들을 그룹핑 하여 컴포넌트 후보로 식별한다. 또한 레거시 시스템의 재공학 방법과 이를 성공적으로 수행하기 위한 전략을 결정하고, 추출된 재사용 요소들의 컴포넌트화 실행을 위해 S/W 및 컴포넌트, 시스템 아키텍처를 정의한다. 그리고 추출된 컴포넌트의 인터페이스를 식별하고, 컴포넌트 내부의 정적 구조 및 동적구조를 생성하고, 시스템 아키텍처를 바탕으로 목표 시스템을 위한 컴포넌트를 설계, 구현하고 테스트 한다.

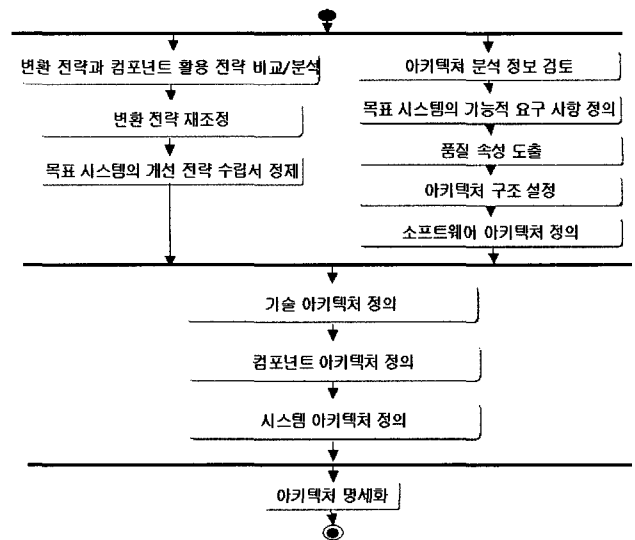
(그림 9)은 컴포넌트화 단계를 구성하는 활동과 작업들을 나타낸다.

컴포넌트 마이닝 활동은 레거시 시스템을 비즈니스 기능을 수행하는 단위에 따라 여러 부분으로 나누고 이들을 각각 컴포넌트로 대응시켜 그것들을 파악하고 추출한다. 이를 위해 레거시 시스템에서 하나의 비즈니스 기능을 수행하는 단위를 추출하기 위해 역공학 과정에서 추출된 시스템 정보를 바탕으로 의미적으로 연관성이 높은 시스템 구성 요소(elements)들을 그룹핑하고 이것을 컴포넌트 후보로 인식하고, 추출된 컴포넌트 후보를 컴포넌트 활용 전략에 따라 평가하여 새로운 시스템에서 활용하기 위한 방안을 수립한다.

(그림 10)은 컴포넌트 마이닝 활동에 대한 작업 흐름을 요약한 것이다.

아키텍처 변환 활동에서는 레거시 시스템을 재공학 하기 위한 방법과 이를 성공적으로 수행하기 위한 전략을 확인하고, 추출된 재사용 요소들의 컴포넌트화 기법들을 확정한다. 이를 위해 재공학 요구를 분석하여, 재공학 시스템이 목표로 하는 새로운 환경을 정의한다. 또한, 재공학 시스템의 S/W 아키텍처를 재모델링 하고, 상호작용 모델링을 통해 비즈니스 컴포넌트들에 대한 컴포넌트 아키텍처를 설계하며, 기술적 아키텍처를 포함하는 시스템 아키텍처를 정의한다.

아키텍처 변환 활동을 구성하는 작업들의 흐름은 (그림 11)에 요약되어 있다.

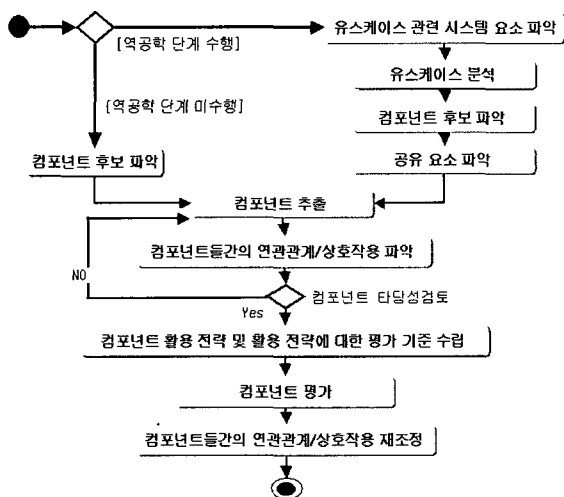


(그림 11) 아키텍처 변환 활동에 대한 작업 흐름

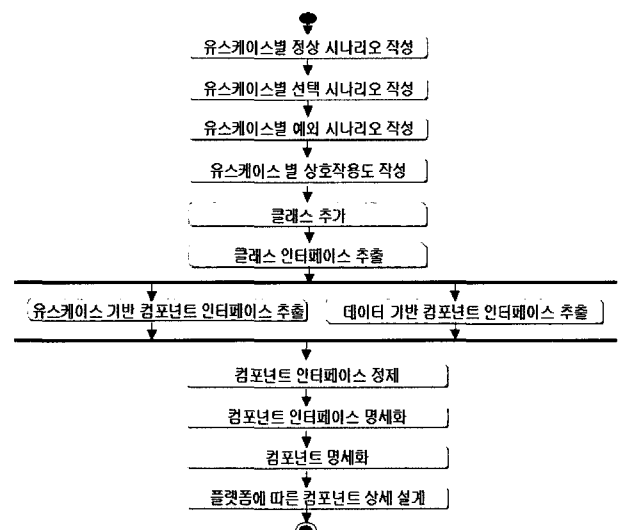
컴포넌트 변환 활동에서는 추출된 컴포넌트의 인터페이스를 식별하고, 컴포넌트 내부 구조를 설계하고, 컴포넌트 내부 클래스들 간의 동적인 메시지 흐름 정보를 기반으로 컴포넌트 인터페이스의 오퍼레이션들을 식별한다. 또한 이렇게 추출된 설계 정보들을 가지고 컴포넌트를 구현 기술 플랫폼에 맞도록 매핑하여 구현하고, 구현된 컴포넌트 각각에 대한 단위 시험을 시행한다.

전체 5개 작업으로 구성되는 본 활동의 작업 흐름은 (그림 12)와 같다.

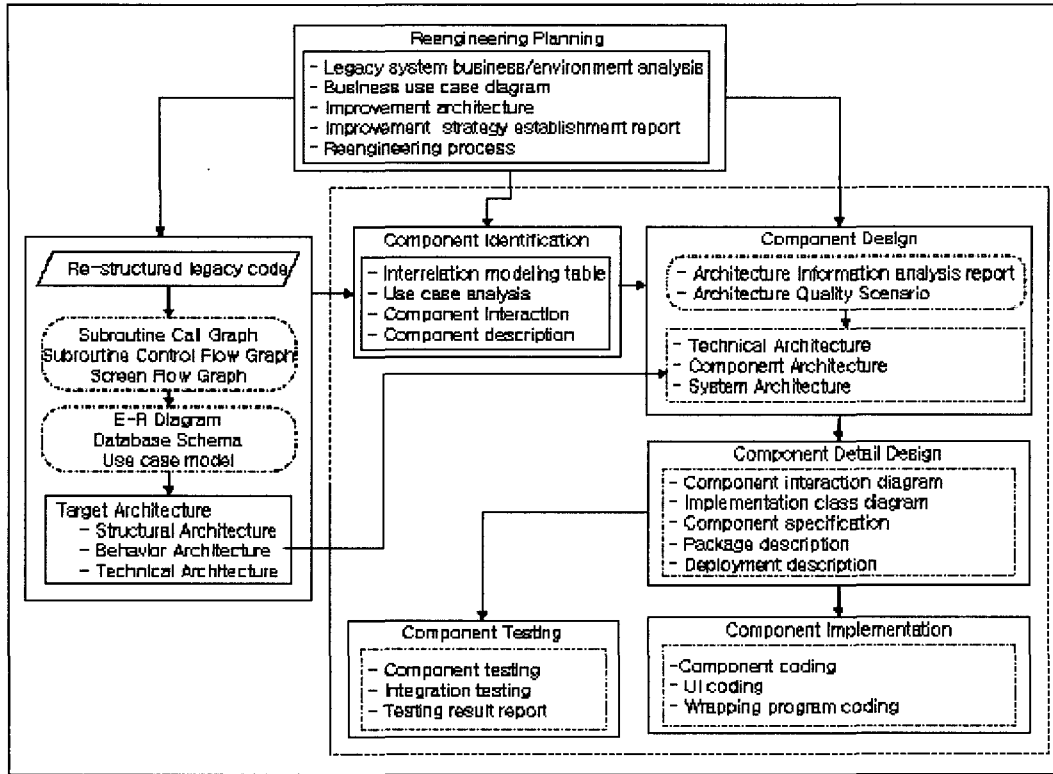
컴포넌트 개발 활동은 실제 타겟 환경의 요구 사항에 맞는 새로운 컴포넌트로 레거시 시스템의 분석 정보들을 변환시키는 작업들로 구성된다. 이를 위해 실제 구현할 화면에 대하여 UI 화면을 구현한 후 컴포넌트와 연동 시키며, 컴포넌트를 구성하는 여러 구성 요소(인터페이스, 빈 클래스, 내부 클래스 및 클래스 메소드, 주요 키 클래스)들을 적용 기



(그림 10) 컴포넌트 마이닝 활동에 대한 작업 흐름



(그림 12) 컴포넌트 변환 활동에 대한 작업 흐름



(그림 13) 마르미-RE 산출물 사이의 연관도

술과 표준에 맞게 구현한다. 또한, 개발된 컴포넌트들에 대해 각 컴포넌트 단위로 시험을 수행하며, 아울러 각 컴포넌트 내부의 클래스들의 시험도 수행한다.

컴포넌트 통합 시험 활동에서는 개발된 개별 컴포넌트들을 프로토타이핑 구축을 통해 통합함으로써, 레거시 시스템이 가진 전체 기능을 발휘 여부와 제약 사항들을 분석하고 점검한다. 이를 위해 변환 전략에 따라 추출된 컴포넌트들을 재공학 시스템 아키텍처 상에 배치하고 통합하여, 구현 컴포넌트들이 다른 컴포넌트들과 제대로 통신하는지를 평가한다. 또한 컴포넌트 아키텍처와 비즈니스 요구가 잘 정의되고 구현되었는지를 테스트한다. 특히 컴포넌트 변환 활동과 함께 마르미-III와의 연계 작업이 많은 부분으로, 프로세스 수행 상 많은 피드백이 발생한다.

설명된 MaRMI-RE를 적용하여 레거시 시스템의 재공학을 수행할 경우 생성될 산출물들 간의 관계성이 (그림 13)에 제시된다.

3.3 재공학 프로세스 수립

재공학자들이 실제로 마르미-RE를 적용하여 그들 자신의 프로젝트를 전개하기 위해서는 (그림 2)의 기본 프로세스를 경우에 따라 선택, 조정할 수 있는 시나리오 제공이 필요하다. 이를 위해 마르미-RE는 프로세스 시나리오 제시를 통해 재공학자들이 최적의 프로세스를 맞춤할 수 있도록 하였다.

다음의 <표 1>은 마르미-RE의 기본 프로세스를 맞춤하기 위한 개별적인 전개 시나리오를 간략히 요약한 것이다.

<표 1> 마르미-RE의 재공학 시나리오

유형	특징
1	<p>계획 → 역공학 → 컴포넌트화 → 인도</p> <p>역공학 단계의 모든 작업들이 완료된 후 컴포넌트화 단계로 진행할 수도 있으나, 선별된 역공학 작업만을 수행한 후 컴포넌트 단계로 진행하고, 필요에 따라 다시 역공학 단계의 활동과 작업들로 피드백 하여 작업 수행</p> <ul style="list-style-type: none"> - 레거시 시스템의 참조 자료(코드/설계서/운영서 등)가 풍부할 때 - 레거시 시스템 관련 작업자(개발자/운영자/유지보수자 등)로부터 레거시 시스템에 대한 정보를 충분히 획득할 수 있는 경우 - 레거시 프로그램의 코드 품질이 우수하거나 도구 지원이 가능한 경우 - 레거시 코드의 재사용이 전략적으로 필요한 경우 - 업무별 레거시 소스 코드의 모듈 분리가 가능한 경우 - 변환 전략이 폐기가 아니거나, 컴포넌트 변환 계획이 미진한 경우
2	<p>계획 → 컴포넌트화 → 역공학 → 컴포넌트화 → 인도</p> <p>컴포넌트화 단계로 먼저 진행한 후, 컴포넌트 작업 중 필요 정보들을 역공학 단계로 피드백 하여 획득하는 작업을 반복 수행하면서 변환</p> <ul style="list-style-type: none"> - 컴포넌트화를 위한 프로젝트의 비전이 명확한 경우 - 컴포넌트화의 구체적인 전략 및 레거시 관련 자원이 부족한 경우 - 레거시 시스템의 일정 부분에 대한 유지가 필요한 경우 - 레거시 시스템에 관련된 전문 작업자의 도움이 가능한 경우 - 인력 수준이 높으며 프로젝트 수행을 위한 기반이 안정된 경우

<표 1> 마르미-RE의 재공학 시나리오(계속)

3	
	<p>역공학 단계의 작업 없이 직접 컴포넌트화 단계로 진행한 후, 새롭게 요구되는 비즈니스 컴포넌트들을 생성 하기 위해 마르미-III의 필요 활동들을 수행하고, 그 결과들 컴포넌트 단계의 산물들과 통합하여 프로젝트를 진행</p> <ul style="list-style-type: none"> - 컴포넌트화 단계를 진행하기 위한 구체적인 방향이 수립되어진 경우 - 레거시의 이해가 충분하고 추가될 비즈니스가 뚜렷이 정의된 경우 - 레거시 시스템에 대해 활용할 가치 있는 자원이 없는 경우 - 컴포넌트화 단계를 진행할 인력 자원이 풍부한 경우
4	
	<p>계획 단계의 일차 분석에서, 레거시 시스템에 크게 영향을 받지 않으면서 새롭게 변화되어야 할 부분이 대부분인 경우, 먼저 마르미-III 작업에서 순공학 작업을 통해 필요한 컴포넌트들을 먼저 생성한 후, 컴포넌트 단계로 진행하여 재공학 프로젝트의 비전과 전략에 따른 컴포넌트화 작업 수행</p> <ul style="list-style-type: none"> - 레거시 시스템과 독립적인 새로운 비즈니스 추가가 많이 요구되는 경우 - 레거시 시스템에 대한 충분한 이해를 가지고 있는 경우 - 레거시 시스템에 대해 활용할 가치 있는 자원이 없는 경우 - 재공학 발주자를 비롯한 이해 당사자들 간의 의사 합의가 원활한 경우 - 시스템 아키텍처 및 분석가 등 프로젝트 참가 인력의 수준이 높은 경우
5	
	<ul style="list-style-type: none"> - [유형 1]과 [유형 3]이 결합된 유형으로, 타겟 시스템이 레거시 시스템의 범주 내에 포함된 것 이외의 다른 비즈니스가 필요할 때 이용 - 계획 단계에서 세워진 비전과 변환 전략에 따라 역공학 단계를 통해 레거시 시스템의 정보들을 수집하고, 컴포넌트 단계를 진행하는 중, 새롭게 추가되는 비즈니스들을 마르미-III 프로세스를 통해 생성한 후, 다시 컴포넌트화 단계로 복귀하여 컴포넌트 전략 하에서 기존의 컴포넌트들과 통합하는 과정으로 수행 <p>- [유형 1]과 [유형 3]의 사례가 혼합 경우에 적용</p>
6	
	<p>계획 단계를 수행한 후, 역공학 단계를 걸쳐 레거시 시스템의 자료로부터 추출될 컴포넌트의 정보를 획득하는 동시에, 마르미-III 작업을 통해 새롭게 추가되는 비즈니스에 대한 컴포넌트 생성이 병렬적으로 이루어진 후, 컴포넌트화 단계에서 이들을 통합하는 과정으로 진행</p> <ul style="list-style-type: none"> - 계획 단계에서 변환 전략 및 기법들을 명확히 결정할 수 있는 경우 - 결정된 전략들의 변경이 거의 발생하지 않는 경우 - 타겟 시스템이 일부는 레거시 시스템의 재사용을 극대화하고, 일부는 새로운 비즈니스 확장을 요구하는 경우 - 프로젝트 관리자 및 시스템 아키텍처 등, 상위 작업자들의 인력 수준이 높은 경우 - 시스템의 점진적 유지보수 및 전개를 수행할 경우

4. 사례 적용

마르미-RE는 전체 약 30만 라인의 124개 COBOL 프로그램으로 구성된 국내 모기업의 커피탈 영업 관리 시스템을 대상으로 사례 적용을 시행하였다. 이 시스템은 지방 3개의

커피탈 회사 합병으로 개발 환경 및 언어가 상이한 3가지 리스 관리 시스템을 독립적으로 운영 중이다. 재공학을 통한 개선 방향으로 업무 표준화, 언어 통일, 시스템 운영 환경 개선(웹 환경), 3개 회사 통합, 데이터베이스 표준화를 목표로 하고 있다.

타겟 시스템은 웹 환경 상에서 사용자가 접근, 관리할 수 있는 컴포넌트 기반 시스템으로 변경된 것으로, J2EE 기반 컴포넌트화를 위해 기존 코볼 프로그램의 비즈니스 로직 부분은 레거시 코드를 사용하며 UI 부분을 컴포넌트화하며 새로운 요구사항을 반영하며, 컴포넌트화 방법으로 랩핑을 주로 사용하였으며, 레거시 시스템 분석 내용 외의 별개의 새로운 요구사항은 변환을 도입하였다. 재공학을 통해 리스 관리 시스템에서 영업관리는 14개의 유스케이스로 식별하고 이들을 컴포넌트 후보로 선정하였다. 그리고 컴포넌트 후보를 분석하여 마르미-RE의 각 단계를 적용하여 컴포넌트화 하였으며, 새로운 요구사항의 컴포넌트는 순공학 방법의 컴포넌트화 방법론인 마르미-III를 적용하여 분석 설계하였다.

5. 프로세스 평가

효율성을 평가하기 위해 소프트웨어 프로세스의 정량적 실험과 같은 정규화된 분석 메소드[14, 15]는 적절한 평가 메트릭의 부족으로 매우 어렵다. 따라서 본 논문에서는 MaRMI-RE를 재공학의 기술적 요소에서 도출한 평가 항목을 기준으로 다른 재공학 프로세스와의 비교를 통해 그 유용성을 살펴보았다. 평가 항목들은 ISO/IEC 12207[16], ISO/IEC 15504[17]와 같은 소프트웨어 프로세스 표준에 근거한 것들이다.

MaRMI-RE의 특징은 <표 2>에 요약되어 있다. 먼저 프로세스 적용시 산출물간의 가독성을 높이기 위해서 UML 모델링을 사용하였다. 이는 모든 재공학 프로젝트의 전체 개발 과정에서 생성된 산출물들 간의 연관관계를 명확히 할 수 있어 개발자들이 이전 작업에 대한 이해를 높이고 특정한 상황에 맞도록 작업을 조절할 수 있도록 한다. 재공학 시행 여부 및 재공학 대상의 우선 순위 결정을 위한 구체적인 작업 절차를 제공한다. 재공학 전과정에 대한 구체적인

<표 2> 기존 재공학 프로세스와 MaRMI-RE의 비교

Criteria	CORUMII	MORALE	DARE	MaRMI-RE
도구 지원 여부	○	●	○	○
표기 방법	itself	itself	itself	UML based
프로세스 맞춤 기법	X	○	X	●
재공학 타당성 평가 절차	○	●	X	○
CBD 지원	X	X	X	●
품질 관리 절차	○	○	○	X
상세 지침	○	○	○	●
산출물 템플릿	X	X	○	●
아키텍처 변환 메소드	●	X	●	●

(● : 완벽히 지원, ○ : 부분적 지원, X : 미지원)

지침과 산출물의 템플리트를 제공함으로써 MaRMI-RE의 학습과 적용을 용이하게 한다.

6. 결론 및 향후 연구

변하는 IT 기술의 출현과 비즈니스 정보 모델의 다양한 변경, 시스템 처리 로직의 급격한 복잡성 증가는 레거시 시스템들의 잦은 유지보수에서 발생하는 기술적, 비용적 한계를 드러내고 있다. 사실, 대부분의 레거시 시스템들은 표준화와 개방성의 결여나 변경의 어려움, 분산 아키텍처 결여 등의 원인으로 새로운 기술들을 수용하고 복잡한 비즈니스 요구를 확장 시키기에는 많은 문제점을 가지고 있다. 특히, 기존의 재공학 방법론들은 사용자가 자기의 의도에 맞도록 재공학 절차와 기법을 선택하거나 반복적으로 수행할 수 있는 지원 체계와 표준 지침을 갖추지 못하고 있어, 중요한 결정적 선택에서는 사용자의 전문적인 식견에 의존하고 있는 실정이다.

본 논문에서는 레거시 시스템들의 자산 가치를 극대화 하기 위한 체계적인 방법론으로서 마르미-RE를 제안했다. 마르미-RE는 CORUM II의 "Horse Shoes" 메타 모델에 근간을 두고 개발된 아키텍처 기반의 컴포넌트화 프로세스로 컴포넌트 및 웹 서비스 시스템으로 변환하거나 랩핑하기 위한 절차와 기법을 제공한다. 계획 단계, 역공학 단계, 컴포넌트화 단계, 인도 단계의 4 개의 단계로 구성되어 있으며, 인도 단계를 제외한 3개 단계의 9개 활동, 29개 작업에 대한 절차와 기법, 산출물들이 완성되어 있는 상태다.

마르미-RE는 기존의 방법론과 같이 순차적이거나, 동기화된 전개 프로세스가 아니라, 병렬적이고 선택적인 재공학 프로세스를 맞춤 구성할 수 있도록 정의 하였다. 현재 비즈니스 영역에서 잘 운영되는 레거시 시스템을 대상으로 정의된 프로세스를 적용시켜 그 장단점을 평가하고 있다. 향후, 마르미-III와의 연계성을 정립하여 순공학과 재공학이 통합된 완전한 라이프 사이클 지원 프로세스를 완성하고, 관련 도구에 의한 지원 체계 구축에 대한 연구를 추진함으로써 마르미-RE의 활용성과 신뢰성을 극대화 시킬 계획이다.

참 고 문 헌

[1] Dolly M, Neumann, "Evolution Process for Legacy System Transformation", IEEE Technical Applications Conference, Washington, pp.57-62, November, 1996,
 [2] William Ulrich, Legacy Systems: "Transformation Strategies", Prentice Hall, 2002
 [3] Nelson Weideman, Dennis Smith, Scott Tilley, "Approaches to Legacy System Evolution", CMU/SEI-97-TR-014, 1997
 [4] Rick Kazman, Steven G. Woods, S. Jeromy Carriere, "Requirements for Integrating Software Architecture and Reengineering Models: CORUM II", Fifth Working Conference on Reverse Engineering, Honolulu, Hawaii,

pp.154-163, Oct., 1998.
 [5] Abowd G. Goel A. Jerding D.F., McCracken M., Moore M., Murdock J.W., Potts C., Rugaber S., Wills L., "MORALE. Mission Oriented Architectural Legacy Evolution" International Conference on Software Maintenance, Bari, ITALY, pp.150-159, Oct., 1997.
 [6] G Antoniol, R. Fiutem, Cristoforetti, "Design Pattern Recovery in Object Oriented Software," 6th Workshop on Program Comprehension, Ischia, Italy, June, 1998, pp.153-163
 [7] Christian Kramer, Lutz Prechelt, "Design Recovery by Automated Search for Structural Design Patterns in Object-Oriented Software", 3rd Working Conference on Reverse Engineering(WCRE), Monterey, CA, pp.208-215, November, 1996.
 [8] Rudolf K. Keller, Reinhard Schauer, Sébastien Robitaille, and Patrick Pagé, "Pattern-Based Reverse-Engineering of Design Components", Conference on Software Engineering (ICSE), Los Angeles, CA, pp.226-235, May, 1999.
 [9] William C. Chu, Chih-Wei Lu, Chih-Peng Shiu, Xudong He, "Pattern Based Software Reengineering : A Case Study", Journal of Software Maintenance: Research and Practice, Vol.12, No.3, pp.300-308, May/June, 2000.
 [10] Peter Herzum, Oliver Sims, Business Component Factory : A Comprehensive Overview of CBD for the Enterprise, OMG press, December, 1999.
 [11] RESORT 사용자 매뉴얼, soft4soft, 2004 <http://www.soft4soft.com>
 [12] LegaMod 사용자 매뉴얼, (주)케미스, 2004, <http://www.camis.co.kr>
 [13] 차정은 외, "레거시 시스템의 재공학 방법론을 위한 메타 모델 정의" 제5회 한국 소프트웨어공학 학술대회, 2003. 2
 [14] 김철홍 외, "Legacy 시스템의 컴포넌트화를 위한 재공학 방법론", 한국 SI학회지, Vol.2, No.1, 2003. 5.
 [15] 한국전자통신연구원, 마르미-RE Ver1.0 매뉴얼, 2004.2
 [16] ISO 12207: Information Technology-Software Life Cycle Processes, Int'l Organization for Standardization, 1995.
 [17] ISO 15504: Information Technology-Software Process Assessment (Part 1 to 9), Int'l Organization for Standardization. 1998.



차 정 은

e-mail : mary2743@etri.re.kr

1995년 효성여대 전자계산학과(이학사)

1997년 대구효성가톨릭대학교 대학원 전자계산학(이학석사)

2001년 대구효성가톨릭대학교 대학원 전자계산학(이학박사)

2001년~현재 ETRI 임베디드 S/W연구단 S/W공학팀 선임연구원
 관심분야: 컴포넌트기반 개발, 임베디드 소프트웨어, 재사용, 설계 패턴, 재공학



김 철 흥

e-mail : kch@etri.re.kr

1982년 충남 문과대학교

1993년 성균관대학교 대학원 정보공학과
(이학석사)

현 재 ETRI 임베디드 S/W연구단 S/W공
학팀

관심분야: 임베디드 소프트웨어 재사용, 프로덕트라인, 재사용
프레임워크



양 영 종

e-mail : yangyj@etri.re.kr

1979년 서울대학교 사범대학 과학교육과

1995년 영국 레스터대학교 대학원 전산학
과(이학석사)

현 재 ETRI 임베디드 S/W연구단 S/W공
학팀 팀장(책임연구원)

관심분야: 임베디드 시스템 프레임워크, 프로덕트 라인, 컴포넌
트기반 개발