

레거시 시스템의 소프트웨어 컴포넌트화를 위한 재공학 프레임워크

박옥자^U 유철중 장옥배
전북대학교 컴퓨터학과
oipark@cs.chonbuk.ac.kr

A Reengineering Framework for Software Componentization on Legacy System

Oak-Cha Park^U Cheol-Jung Yoo Oak-Bae Chang
Dept. of Computer Science, Chonbuk National University

요 약

최근 시스템의 컴포넌트화 경향은 인터넷 환경이 보편화되면서 플러그앤플레이(plug and play) 형태로 조립하여 재사용하려는 기업의 요구에 가장 핵심적인 패러다임으로 등장하고 있다. 따라서, 기업은 기존의 레거시 시스템을 컴포넌트화하여 유지보수뿐만 아니라 새로운 시스템 도입에 있어서도 유연성 및 확장성을 용이하게 하기 위한 재공학 솔루션을 필요로 하고 있다. 따라서, 본 논문에서는 레거시 시스템용 소프트웨어 컴포넌트화하기 위한 재공학 프레임워크를 제안하였다. 이 프레임워크는 컴포넌트 개발 방법론에서 필요한 절차와 Robert C. Seacord와 John Robert가 제안한 레거시 시스템 현대화 방법론을 통합하여 제시하였으며, 프레임워크의 각 단계는 레거시 시스템의 분석 단계, 컴포넌트화 방법 결정 단계, 마지막으로 컴포넌트로의 변환 단계로 이루어져 있다. 각 단계에서는 다시 세부 절차와 이에 필요한 제반사항을 기술하여 레거시 시스템을 소프트웨어 컴포넌트화하는데 있어서 필요한 절차와 가이드라인을 상세하게 제공하고자 한다.

1. 서론

최근 소프트웨어를 부품화시켜 재사용하기 위해 대두되고 있는 컴포넌트 기반 개발 방법론(Component-Based Development)은 차후 21세기 소프트웨어 산업을 이끌어갈 핵심적인 패러다임으로 등장하고 있다. 특히, 언젠, 어디서나, 누구나 필요한 정보를 쉽게 얻을 수 있는 인터넷 환경이 보편화되면서 플러그앤플레이(Plug and Play) 형태로 조립하여 사용할 수 있음에 따라 컴포넌트 기반 소프트웨어 개발 추세는 더욱 가속화되고 있다[1]. 따라서, 기업에서도 기존의 레거시 시스템(legacy System)을 컴포넌트화(componentization)하여 유지보수뿐만 아니라 유연성 및 확장성에 있어서도 현재의 소프트웨어 발전 경향에 뒤처지지 않기 위한 재공학 솔루션을 필요로 하고 있다. 하지만, 이미 구축되어 노후한 시스템을 컴포넌트화하기 위해서는 몇 가지 문제점이 남아있다. 먼저 레거시 시스템을 컴포넌트화하는 과정에 있어서 선행적으로 수행해야 할 단계별 절차와 가이드라인이 명확하게 정의되어 있지 않아 어려움을 주고 있다[2]. 또한, 이상적인 컴포넌트를 구축에 필요한 정의가 명확하게 기술되어 있지 않아 컴포넌트화하는데 있어서 필요한 정보 추출과 이를 기반으로 이루어지는 래핑(wrapping)방법도 모호하다.

따라서, 본 논문에서는 레거시 시스템을 컴포넌트화하기 위한 단계별 가이드라인과 정형화된 명세를 기반으로 한

프레임워크를 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 레거시 시스템을 새로운 환경에 맞게 변환시키는 소프트웨어 재공학에 대해 간략히 알아보고, 3장에서는 레거시 시스템을 컴포넌트화하기 위해 본 논문에서 제안한 프레임워크에 대해 설명하며, 마지막 4장에서는 결론 및 향후연구에 대해 기술한다.

2. 소프트웨어 재공학

재공학은 유지보수 활동중에서 소프트웨어 미래의 유지보수성이나 신뢰성을 개선시키기 위해, 또는 미래의 성능향상을 위해 더 좋은 기능을 제공하면서 이루어지는 방법이다 [3]. [4]에서는 재공학 전략을 크게 기술적 관점과 비즈니스 관점에 따라 구분하였다. 기술적 관점은 레거시 시스템의 현재 상태와 그것을 재공학하기 위해 이용하는 도구가 주요 변수가 되고, 비즈니스 관점은 재공학을 할 경우의 비용과 그에 대한 효율성간의 상충관계(trade-offs)를 찾아서 타당성을 분석한 후 수행하는 것이다. 이와 같은 두 가지 관점을 레거시 시스템을 새로운 환경으로 변환시키는데 있어서 비용측면의 문제가 주요 쟁점이 될 수 있으므로 현대화(modernization)할 것인지, 새로운 시스템으로 교체(replacement)할 것인지 여부의 결정을 필요로 한다. 따라서, 우리는 레거시 시스템을 대상으로 할 경우 컴포넌트화

하기 위해 어떤 방법에 의해 재공학할 것인지를 선택하는 변수가 주요 쟁점이 될 수 있을 것이다. 이 논문에서 제안하는 프레임워크는 위와 같은 문제점에 대한 가이드라인을 마련하면서 상세한 절차를 기술하였다.

3. 레거시 시스템의 컴포넌트화를 위한 라이프사이클

본 논문에서 제안하는 재공학 프레임워크는 기존의 레거시 시스템을 컴포넌트화하기 위한 단계별 절차를 기술한 것이다. 이 모델은 Robert C. Seacord과 John Robert[3]이 제안한 현대화 기법 모델과 일반적인 컴포넌트 개발 방법론의 특징을 조합하여 구축한 프레임워크로서 세 단계로 이

루어진다. 1단계는 레거시 시스템 분석 단계이고, 2단계는 분석된 정보를 바탕으로 한 컴포넌트화 방법 결정 단계이며, 마지막으로 3단계는 컴포넌트로의 변환 단계이다. 이 세 단계는 단계적으로 수행되면서 유지보수 과정을 통해 순환적으로 진행되며 각 단계마다 각각의 상세 절차가 명시되어 있다. 그림 1은 각 단계별 특징을 간단히 도식화한 것이며, 표 1은 그림 1에 대한 단계별 상세 사항을 기술한 결과이다.

3.1 레거시 시스템 분석 단계

이 단계에서는 레거시 시스템을 분석하고 모듈화 가능한 범위를 선정하여 이를 컴포넌트화할 수 있도록 영역을 설정하는 단계이다. 이 영역 선정단계는 크게 비즈니스 로직부

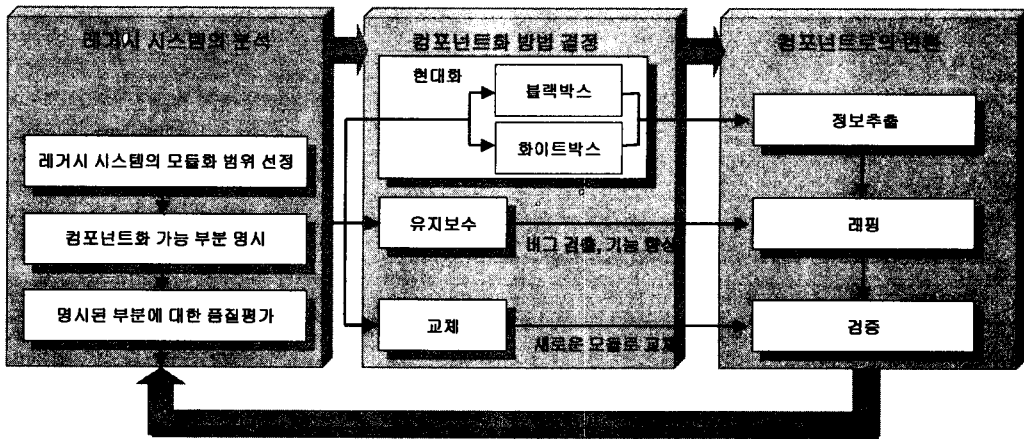


그림 1 레거시 시스템의 컴포넌트화를 위한 프레임워크

	1단계: 레거시 시스템의 분석 및 평가	2단계: 컴포넌트화 방법 선정	3단계: 컴포넌트 변환
역할	레거시 시스템을 컴포넌트화하기 위한 분석	1단계에서 얻어진 정보를 바탕으로 세 항목의 범주에 맞는 가이드라인을 마련하여 재공학 방법을 선정한다. 각 재공학 방법에는 일정한 가이드라인이 필요	2단계에서 정해진 절차에 의해 각각 필요한 정보를 추출하여 컴포넌트화하고 검증 유지보수나 교체된 컴포넌트를 조립 가능하게 함
세부 단계	가. 레거시 시스템의 컴포넌트화 할 범위 선정 - 비즈니스 로직 부분 - 데이터 처리부분 - 사용자 인터페이스 부분 나. 컴포넌트화 할 영역에 대한 상태 평가	범주 1: 현대화(modernization) 할 것인가? - 화이트박스(white-box) 현대화 - 블랙박스(black-box) 현대화 범주 2: 유지보수(maintenance)할 것인가? - 버그 점검, 간단한 구조 변경 범주 3: 교체(replacement)할 것인가?	가. 정보 추출 - 프로그램 슬라이싱 - 각 언어에 맞는 추출방법론 나. 래핑(wrapping) - 래핑 절차 및 저장 방법 연구 다. 검증
가이드라인	· 이상적인 컴포넌트 구축을 위해 필요한 시스템 분석 가이드라인 · 선정된 레거시 코드의 컴포넌트화 가능성을 평가하기 위한 별도의 수정된 매트릭스 필요	· 컴포넌트 변환 방법을 선택하기 위한 가이드라인과 제한조건을 정의하고 절차 명세 · 현대화 할 경우 블랙박스 현대화 할지 화이트박스 현대화 할지에 대한 상세 가이드라인 마련 · 각 방법에 대한 가이드라인 필요 · 컴포넌트 변환 후 이루어질 테스트 방법론 정의	· 컴포넌트 명세 필요 · 컴포넌트 아키텍처 필요 · 이를 기반으로 프로그램 슬라이싱을 위한 기본 가이드라인 필요 · 각 언어에 맞는 정보 추출 방법 · 각 언어에 맞는 래핑 방법 · 각 언어에 맞는 테스트 방법 마련

표 1 프레임워크의 단계별 상세 절차

분, 데이터 처리 부분, 사용자 인터페이스 부분으로 접근할 수 있다.

이 단계에서 이루어지는 분석 활동은 모듈화 가능한 부분을 분할할 때 필요한 가이드라인과 선정된 레거시 코드의 컴포넌트화 상태를 평가하기 위한 별도의 수정된 매트릭스가 필요하다.

3.2 컴포넌트화 방법 선정 단계

이 단계에서는 1단계에서 수행한 컴포넌트화할 모듈 범위를 선정하고 난 후의 재공학 방법을 선정하는 단계이다. 일반적으로 재공학 방법에는 유지보수, 교체 그리고 현대화 세 가지 범주로 이루어진다. 유지보수 과정은 작은 변경만으로도 시스템을 진화시키는 것으로 반복적이면서 증가적인 프로세스이다. 이 단계에서 이루어지는 활동은 버그 검출과 간단한 기능 향상을 수행하지만 시스템 자체의 구조적 변경은 가하지 않는다. 따라서 3단계로 이동했을 경우 간단한 수정만 가한 후 래핑단계로 들어가게 된다.

교체는 최근의 비즈니스 요구를 맞출 수 없는 레거시 시스템에 적절하다. 기존의 레거시 시스템이 현대화가 가능하지 않거나 유지보수에 대한 비용이 너무 많이 소요될 경우 과감히 교체 방법을 선택하는 것이 바람직하다. 또한, 유지보수 하는데 있어서 문서화되지 않거나 낱짜가 지났을 경우에도 교체방법이 적합하다. 교체는 선정된 영역의 모듈을 새롭게 컴포넌트화하게 되므로 다음 3단계에서의 정보 추출이나 래핑과정을 거치지 않고 곧바로 검증 단계에 적용된다.

마지막으로 현대화는 유지보수보다는 더 광범위한 변경을 필요로 하면서 기존 시스템의 주요 부분은 그대로 보존하려고 할 때 유용하다. 현대화 방법에는 블랙박스 현대화와 화이트박스 현대화가 존재한다. 블랙박스 현대화는 시스템 내부 구조의 이해는 필요 없이 입력력 인터페이스의 이해를 통해 현대화시키는 방법이다. 이 방법은 보통 래핑을 기반으로 수행하는데 소프트웨어 가공물에 의해 유입된 인터페이스와 통합을 위해 실제 요구되는 인터페이스의 불일치를 제거하기 위해 사용되기도 한다. 따라서, 기존의 레거시 시스템의 내부 구조를 변경할 필요없이 컴포넌트화하기 위해서는 블랙박스 기법을 선택하여 래핑으로 전환하는 것이 바람직하다. 반면, 화이트박스 현대화 기법은 내부 시스템의 오퍼레이션의 이해를 중요시한다. 또한, 시스템의 컴포넌트와 그들간의 관계를 명세해주어야 하며 상위레벨의 추상화를 표현하게 된다. 프로그램 이해과정을 거친 다음 코드를 재구축하게 되는데 이때 응집도 있는 모듈로 분할되도록 하면서 컴포넌트로의 변환에 필요한 정보 추출을 위한 슬라이싱 기법이 필요하다[4].

3.3 컴포넌트로의 변환 단계

마지막 3단계인 컴포넌트로의 변환 단계는 이 프레임워크의 마지막 단계로서 2단계에서 설정된 재공학 방법의 가이드라인을 토대로하여 컴포넌트화한다. 먼저 2단계에서 블랙

박스 기법이나 화이트박스 기법을 선택하였을 경우 이 단계에서는 컴포넌트로 변환시키기 위해 어떤 정보를 추출해야 할지 결정한다. 이 추출단계에서는 가장 이상적인 컴포넌트 명세에 맞는 정보를 추출하기 위한 가이드라인과 추출 기법이 필요하다. 래핑 단계에서는 추출된 정보를 바탕으로 이상적인 컴포넌트 구조에 맞는 래핑을 수행하게 된다. 이 방법은 상대적으로 노력이 적게 들면서 컴포넌트 시스템 기반 접근이 가능하고 시스템의 많은 부분에 영향을 주지 않고 레거시 시스템 구조를 변경할 수 있기 때문에 레거시 시스템의 컴포넌트화 방법에서 많이 사용되는 방법이다. 하지만, 이와 같이 래핑을 마치게 되면 우리는 재공학 과정을 거친 컴포넌트를 검증하는 과정으로 기존의 레거시 시스템의 자료를 변경시키지 않고 안전하게 변환되었는지 테스트 해야 한다. 이 단계에서 필요한 활동은 정보 추출 방법과 래핑 전략을 보다 상세화시켜 기술해주어야 하고 변환된 컴포넌트의 테스트 전략을 명세하여야 한다.

지금까지 기술한 세 단계는 일반적인 컴포넌트 변환방법으로서 1단계에서 3단계까지 절차적으로 수행되지만 이미 변환된 컴포넌트 자체의 유지보수가 필요할 경우에도 적용 가능하도록 순환적으로 진행된다.

4. 결론 및 향후 연구

본 논문에서는 기존의 레거시 시스템을 컴포넌트로 변환시키기 위한 프레임워크를 제안하였다. 각 단계는 성공적인 컴포넌트화를 위해 각 단계별로 보다 상세화시켜 기술하였으며 이 절차에 따라 우리가 우선적으로 따라야할 가이드라인을 기술하였다. 이 가이드라인은 각 단계별 세부절차를 수행하기 위하여 우선적으로 필요한 간략한 절차이므로 우리는 이를 구체적으로 가이드라인을 마련하여 정형적으로 수행되도록 하여야 한다. 이 프레임워크는 단순한 소프트웨어 컴포넌트화뿐만 아니라 최근 대두되고 있는 엔터프라이즈 컴포넌트로의 변환에 있어서도 적용가능하도록 제시하였다.

향후 연구에서는 각 단계에서 이루어지는 역할을 보다 상세화시켜 컴포넌트화를 위해 필요한 정형화된 가이드라인을 마련하고자 한다.

참고 문헌

- [1] 김명준, 김재규, 양영중, 컴포넌트 산업 활성화 방안, 한국정보처리학회지, Vol. 7, No. 4, pp.3~9, July 2000.
- [2] Peter Herzum, Oliver Sims, Business Component Factory-A Comprehensive Overview of Component-Based Development for Enterprise. John Wiley & Sons. Inc. 2000.
- [3] 장옥배 외 5인 공역, 소프트웨어공학-이론과 실제, 한산출판사, 2000.
- [4] Rober C. Seacord, John Robert, "A Survey of Legacy System Modernization Approaches," Technical Note CMU/SEI-2000-TN-003, April 2000.