

# 레거시 시스템으로부터

## EJB 컴포넌트 추출을 위한 분석 기법

송문섭<sup>0</sup> 정효택 양영중  
한국전자통신연구원

{sirius, htjung, yangj}@etri.re.kr

### The Analysis Method for Extraction of EJB Component from Legacy System

Moon-Sub Song<sup>0</sup> Hyo-Taeg Jung Young-Jong Yang  
Electronics and Telecommunications Research Institute

#### 요 약

최근 분산 환경에서 최적의 소프트웨어 개발 방법으로 각광받는 컴포넌트 기반 개발은 재사용성과 생산성 등의 이점으로 인하여 급변하는 IT분야의 시스템 구축에 적합한 방법으로 인식되고 있다. 그러나 새로운 시스템의 컴포넌트 기반 개발이 아닌 경우, 즉 기존 레거시 시스템을 컴포넌트 기반 시스템으로 변환하는 방법에 대한 연구는 미비한 상태이다. 본 논문에서는 기존 레거시 시스템에서 컴포넌트 기반 시스템으로 변환하기 위해 필요한 컴포넌트 추출을 위한 레거시 시스템 분석 기법을 제안하고, 이를 웹 어플리케이션에 적용하여 Servlet 프로그램에서 EJB 컴포넌트 추출 시 필요한 레거시 시스템 분석기로 서블릿 코드 분석기(Servlet Code Analyzer)에 대해 설명하였다.

#### 1. 서 론

웹과 인터넷 기술의 발달로 인해 전산 환경이 분산 환경으로 변하고 있으며, 소프트웨어 개발 기술에 있어서도 컴포넌트화가 새로운 패러다임으로 자리잡고 있다.[1] 이러한 컴포넌트 기반 시스템에서는 어플리케이션 개발을 기본적으로 컴포넌트의 조합으로 보고 있기 때문에 단위 컴포넌트 개발은 컴포넌트 프레임워크와 더불어 컴포넌트 소프트웨어 개발에 있어서 중요한 분야이다.[2] 그러나 단위 컴포넌트 개발은 주로 각각의 산업 분야별로 새로 만들고 있을 뿐 기존의 레거시 시스템의 재사용은 이루어지고 있지 않다. 따라서 레거시 시스템을 래핑(Wrapping)이나 변환(Transforming) 등을 통해 컴포넌트 기반 시스템으로 변환할 필요성이 대두되고 있다.[3] 이러한 변환 이외에도 새로운 컴포넌트 기반 시스템을 구축 시 필요한 컴포넌트를 미리 레거시 시스템으로부터 추출할 필요도 있다. 본 논문에서는 레거시 시스템 변환이나 컴포넌트 추출 시 필요한 레거시 시스템 분석 기법을 제안하고 제안한 기법을 웹 프로그램에 적용하여 서블릿을 EJB 컴포넌트화하기 위해 필요한 서블릿 코드 분석기(Servlet Code Analyzer)에 대해 설명하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 레거시 시스템 현대화와 컴포넌트 기반 시스템(Component Based System)에 대해 설명하고 3장에서는 컴포넌트 추출을 위한 레거시 시스템 분석 기법을 제안한다. 4장에서는 제안한 기법을 적용하여 만든 서블릿 코드 분석기에 대해 설명하고 5장에서는 결론 및 향후 연구 방향을 제시한다.

#### 2. 관련 연구

##### 2.1 레거시 시스템 현대화(Legacy System Modernization)

급변하는 IT 분야에서 컴퓨팅 환경이 점차 개방화, 분산화됨에 따라 기존의 레거시 시스템을 새로운 시스템으로의 변환하고자 하는 레거시 시스템 현대화 문제가 발생하게 된다. 이

는 1990년대 말의 Y2K 문제와는 달리 시스템의 유지, 보수 측면에서 치명적이지는 않지만 장기적으로는 시스템의 변환이 이루어져야한다. 컴포넌트 기반 개발 방법이 소프트웨어의 재사용이라면 레거시 시스템 현대화는 시스템의 재사용이라고 생각할 수 있으며, 재개발(Redevelopment) 방법과 비교하면 많은 개발비용을 절약 할 수 있다. 일반적으로 레거시 시스템 현대화는 레거시 시스템의 프로그램과 데이터의 현대화(Modernization)를 의미한다. 재공학 측면에서 레거시 프로그램의 현대화는 역공학으로 연구되고 있으며, 데이터의 현대화는 데이터베이스의 래핑기법 등을 이용하여 다른 시스템 또는 새로운 시스템이 데이터를 액세스할 수 있도록 하는 연구가 활발히 이루어지고 있다.[4]

##### 2.2 컴포넌트 기반 시스템

컴포넌트 기반 시스템이 기존의 다른 시스템과 구별되는 가장 큰 특징은 교체가능성이다. 즉 어플리케이션 사용에 영향을 주지 않고 다른 버전의 컴포넌트나, 동일한 서비스나 기능을 제공하는 다른 컴포넌트로 교체할 수 있다는 것이다. 여기서 다른 컴포넌트로 대체한다는 것은 그 전 컴포넌트보다 새로운 기능 또는 향상된 기능을 추가할 수도 있다는 것이다. 또 다른 컴포넌트 기반 시스템의 특징은 인터페이스의 사용이다. 컴포넌트 기반 시스템에서 컴포넌트의 접근은 물리적인 구현과는 상관없이 설정된 인터페이스를 통해서만 가능하다.

컴포넌트 기반 시스템에서의 어플리케이션의 개발은 기존 시스템이 계획, 설계, 분석, 구현, 테스트 등의 순차적인 방법으로 소프트웨어를 개발했던 것과는 달리 일반적으로 이해, 준비, 조립, 구현, 관리의 단계를 거쳐 개발이 이루어진다.[5] 이해 단계에서는 어플리케이션의 요구사항을 이해하고 컴포넌트와 인터페이스의 형태로 표현된 업무 및 어플리케이션 아키텍처를 만든다. 준비 단계에서는 투명한 인터페이스를 가지는 개별 컴포넌트를 제작한다. 조립 단계에서는 인터페이스를 통한 컴포넌

트의 조립 또는 통합으로 어플리케이션을 생성한다. 구현 단계에서는 컴포넌트와 어플리케이션을 전개하고 실행한다. 관리 단계에서는 프로세스 및 인벤토리를 관리한다. 컴포넌트 기반 개발(Component Based Development)은 기존의 개발 방법에 비해 개발과 유지보수가 용이하며 컴포넌트의 재사용성으로 인하여 개발 생산성과 품질이 향상된다. 이러한 컴포넌트 기반 개발의 장점과 더불어 인터넷의 발달, IT 분야의 표준화 등은 컴포넌트 기반 시스템으로의 전환을 더욱 가속시키고 있다.[6]

### 3. 레거시 시스템 분석 기법

레거시 시스템 현대화의 방법으로는 크게 변환 기법과 래핑 기법으로 나뉘 볼 수 있으며, 레거시 시스템의 특성에 맞춰 다양한 기법을 사용한다. 본 논문에서는 그 중에서 레거시 시스템의 특정 모듈을 EJB 컴포넌트로 변환하기 위해 필요한 레거시 시스템 분석 기법을 제안하고자 한다. 제안하는 레거시 시스템 분석 기법은 크게 5단계로 나뉜다. 5단계는 다음과 같다.

#### 3.1 분석 정보 요구 사항 정의 단계

레거시 시스템에서 EJB 컴포넌트로 변환하기 위해 EJB 컴포넌트 추출에 필요한 정보, EJB 컴포넌트 생성에 필요한 정보 등을 정의하는 단계이다. 요구된 분석 정보는 레거시 시스템에서 추출 가능한지를 먼저 판단하여 모듈별로 정의한다. 다음 그림은 서블릿을 EJB 컴포넌트로 변환하기 위한 분석 정보 요구 사항 정의 단계를 UML로 표현한 예이다.

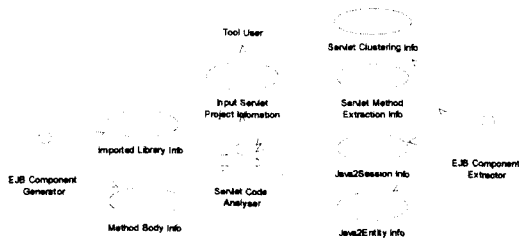


그림 1 분석기 요구 사항 Use Case 예

#### 3.2 분석 정보 DB 생성 단계

분석 정보 요구 사항 정의 단계에서 정의된 분석 정보를 DB Schema를 이용해 데이터베이스를 생성하는 단계로 실제 소스 코드로부터 추출된 분석 정보는 여기에 저장된다.

#### 3.3 분석 정보 추출 문법 작성 단계

분석 정보 추출 문법 작성 단계에서는 요구된 분석 정보를 레거시 프로그램 소스에서 추출하기 위해 레거시 프로그램 문법 작성한다. 즉 분석 정보 추출에 필요한 문법만을 따로 정의해서 작성하는 단계이다.

#### 3.4 Parser 설계 및 분석 단계

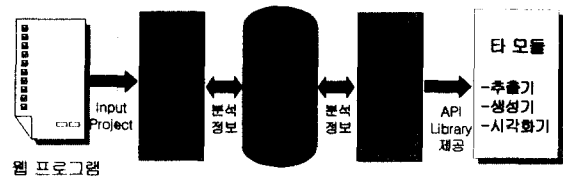
레거시 시스템에 맞게 각각의 파서(Parser)를 설계하고 설계된 파서를 이용해 입력된 레거시 프로그램과 분석 정보 추출 문법을 비교하여 요구된 분석 정보를 분석해낸다. 분석된 분석 정보는 분석 정보 DB에 저장된다.

### 3.5 API 라이브러리 구현 단계

타 모듈(추출기, 생성기 등)에서 요구된 분석 정보를 사용할 수 있도록 인터페이스 역할을 해 주는 단계이다. 모듈별로 API 라이브러리를 설계하고, 각 모듈이 실제 분석 정보 DB를 이용할 수 있도록 해준다.

### 4. 서블릿 코드 분석기

서블릿 코드 분석기는 3장에서 제안한 레거시 시스템에서 컴포넌트를 추출하기 위한 분석 기법을 웹 어플리케이션 프로그램(서블릿)에 적용한 도구이다. 전체적으로 레거시 시스템에서 EJB 컴포넌트를 추출하는 과정은 크게 세 단계로 나뉘 볼 수 있다. 첫 번째 단계에서는 레거시 시스템을 분석하여 컴포넌트 추출에 필요한 정보를 추출하며, 두 번째 단계에서는 추출된 분석 정보를 바탕으로 실제 EJB 컴포넌트를 추출한다. 마지막으로 추출된 EJB 컴포넌트를 실제 생성하는 단계를 거쳐 EJB 컴포넌트를 추출하게 된다. 서블릿 코드 분석기는 위의 첫 번째 단계를 수행하는 도구로 컴포넌트 추출 과정에서 서블릿 코드 분석기의 역할은 다음 그림 2와 같다.



서블릿 코드분석기(SCA)

그림 2 서블릿 코드 분석기(SCA) 전체 구조도

서블릿 코드 분석기는 세부적으로 분석 모듈과 분석 정보 생성 모듈로 나뉜다. 분석 모듈은 입력된 어플리케이션을 패턴 매칭 기법을 이용하여 타 모듈에서 요구한 정보를 분석하는 모듈이며, 분석 정보 생성 모듈은 분석된 결과를 타 모듈에서 분석 정보를 이용해서 작업을 수행할 수 있도록 API 라이브러리를 제공해 주는 모듈이다. 타 모듈로는 EJB 컴포넌트를 추출하는 추출기, 추출된 결과를 이용해서 실제 EJB 컴포넌트를 생성하는 생성기, 추출과 생성된 결과를 사용자에게 시각적으로 제공해 주는 시각화기 등이 있다.

#### 4.1 분석 모듈

다음 그림 3은 분석 모듈의 세부 구조도이다. 분석 모듈은 UI Package, Parsing Package, DB Package로 구성되어 있다. 먼저 분석기는 분석하고자 하는 웹 프로그램을 통합 인터페이스를 통해 입력받는다. 입력받은 웹 프로그램은 Inventory Menu를 거쳐 분석 요청을 받으면 임시 파일(Temp File)을 변환된다. 임시 파일은 입력 프로그램을 분석 정보 추출 문법과 비교하기 위해 분석에 필요 없는 소스를 제거한 파일이다. 작성된 임시 파일을 대상으로 각각의 파서들이 작성된 문법을 기반으로 파싱(Parsing) 작업을 하여 분석 정보 DB에 분석된 정보를 저장한다. 이때 여러 파서들이 필요한 이유는 웹 프로그램의 특성 때문이다. 즉 서블릿 파일의 구성이 HTML, SQL, Java 등의 다중언어로 구성되어 있기 때문이다.

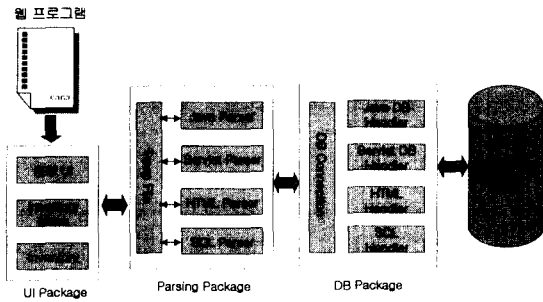


그림 3 분석 모듈 구조도

4.2 분석 정보 생성 모듈

다음 그림 4는 분석 정보 생성 모듈의 세부 구조도이다.

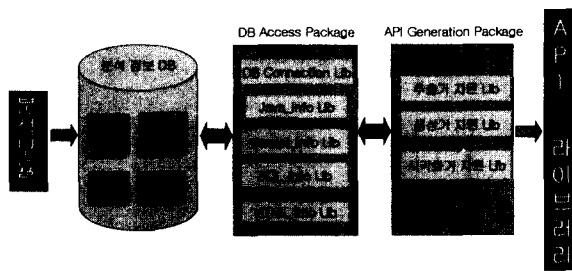


그림 4 분석 정보 생성 모듈 구조도

분석 정보 생성 모듈은 분석 모듈에서 저장한 분석 정보를 분석 정보 DB에서 가져다 API 라이브러리를 만드는 모듈이다. 분석 정보 생성 모듈의 구성은 분석 정보 DB를 접근하기 위해 필요한 DB Access Package와 실제 다른 도구에 제공할 API 라이브러리를 생성하는 API Generation Package로 구성되어 있다. 다음 그림 5는 API Generation Package의 구조도이다.

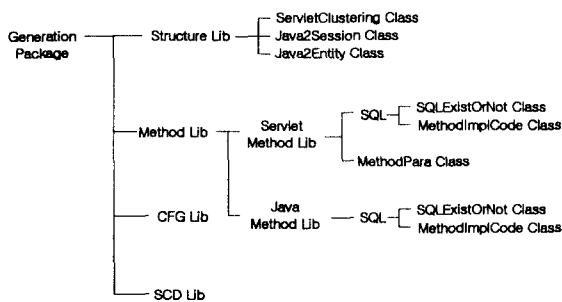


그림 5 API Generation Package 구조도

5. 결론 및 향후 연구 과제

본 논문에서는 레거시 시스템에서 EJB 컴포넌트를 추출하기 위한 분석 기법을 제안하였으며, 제안한 기법을 이용하여 웹 어플리케이션(서블릿)에서 EJB 컴포넌트를 추출하기 위해 필요한 서블릿 코드 분석기를 구현하였다. 아래 그림 6은 서블릿

코드 분석기(Servlet Code Analyzer)가 입력받은 파일을 정의된 요구사항에 맞춰 분석한 결과를 보여주는 화면이다.

현재까지의 작업 결과는 테스트한 예제에 대해서 요구한 정보를 모두 추출하였다. 그러나 많은 파일들에 대한 테스트를 하지 못하였고, 입력으로 사용한 소스가 한정되어 있어서 분석기의 기능이 검증되지 못한 상황이다.

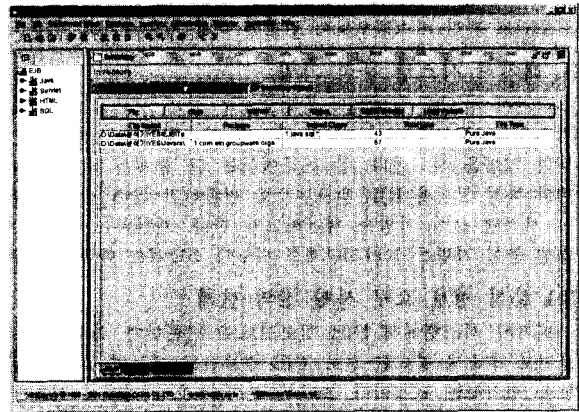


그림 6 서블릿 코드 분석기(Servlet Code Analyzer)

향후 연구 과제는 타 모듈(EJB 컴포넌트 추출기, 생성기)에서 분석된 정보를 API 라이브러리를 통해 사용할 수 있도록 하는 것과 분석된 정보에 대한 검증을 통해 결과에 대한 평가 방법에 관한 연구가 이루어져야 한다.

참고문헌

- [1] U. Assmann, R. Schmidt: Towards a Model For Composed Extensible Components. *Workshop Foundations of Component-Based Systems, Proceedings*, Zurich, Switzerland September 26, 1997.
- [2] Component Source Homepage, <http://www.componentsource.com>
- [3] Seng, Jia-Lang & Tsai, Wayne. "A Structure Transformation Approach for Legacy Information Systems - A Cash Receipts/Reimbursement Example," *Proceedings on the 32nd Hawaii International Conference on System Sciences*, 1999.
- [4] Weiderman, Nelson H. Bergey, John K. Smith, Dennis B. & Tilley, Scott R. *Approaches to Legacy System Evolution* (CMU/SEI-97-TR-014 ). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University. 1997.
- [5] R. Schmidt: Component-based systems, composite applications and workflow-management. *Workshop Foundations of Component-Based Systems, Proceedings*, Zurich, Switzerland September 26, 1997.
- [6] Stets, Robert J. Hunt, Galen C. Scott, Michael L. "Component-Based APIs for Versioning and Distributed Applications," *IEEE Computer*, 54-61, July 1999.