

순환공학(Round-Trip Engineering) 을 지원하는 클래스 다이어그램 설계

정양재, 이우진, 신규상
한국전자통신연구원
e-mail : {cornor, woojin, gsshin}@etri.re.kr

Design Of a Class Diagram Editor Supporting Round-Trip Engineering

Yang-Jae Jeong , Woo-Jin Lee, Gyu-Sang Shin
Electronics Telecommunications Research Institute

요 약

소프트웨어 위기를 극복하기 위해 재사용성, 유지보수성, 적시성 관점에서 컴포넌트 기반 방법론이 크게 각광받고 있다. 또한 기존의 코드를 이용하여 컴포넌트를 만드는 방법, 컴포넌트를 이용하여 새로운 컴포넌트를 만드는 방법 등 여러 컴포넌트 기술이 개발되고 있다. 본 논문에서는 컴포넌트의 개발 시간을 단축하기 위한 컴포넌트 생성 지원 도구 개발에 대해 간단히 설명한 후 설계와 구현을 동시에 할 수 있는 컴포넌트 클래스 다이어그램에 대해 기술한다. 설계와 구현을 동시에 지원함으로써 개발기간을 크게 단축할 수 있고 설계와 구현의 동일성을 보장한다. 또한 컴포넌트 클래스 다이어그램은 EJB 컴포넌트에 전개까지 지원한다.

1. 서론

소프트웨어 위기를 극복하기 위해 소프트웨어의 재사용, 유지보수, 개발 시간 단축이 필요하다. 이를 해결하기 위한 방법으로 컴포넌트 기반 방법론이 크게 각광 받고 있다. 컴포넌트 기술을 이용하여 기존의 코드를 크게 변경하지 않아도 재사용할 수 있는 방법과 컴포넌트를 조립하여 새로운 컴포넌트, 또는 새로운 어플리케이션을 만드는 방법들이 연구되고 있다. 하지만, 보다 빠른 컴포넌트 보급을 위해서는 컴포넌트 재사용에 앞서 새로운 컴포넌트를 보다 쉽고 빠르게 만들 수 있는 컴포넌트 개발 방법이 필요하다.

컴포넌트 개발을 위해 여러 곳에서 컴포넌트 개발 지원 도구를 만들었다. Computer Associate 사의 Cool:Joe, Compuware 사의 Uniface, Select 사의

SCF(Software Component Factory), TogetherSoft 사의 Together 등이 대표적인 컴포넌트 개발 지원 도구들이다. 이러한 도구들은 대부분 특정 컴포넌트 플랫폼 아키텍처만을 지원하고 있으며 몇몇 도구는 설계와 구현을 완전히 분리하여 설계와 구현의 솔기없는(seamless) 연결을 지원하지 못한다[1].

본 논문에서는 컴포넌트 기반의 소프트웨어의 개발 시간의 단축을 위해 설계, 구현, 전개의 모든 과정을 지원하는 컴포넌트 생성 지원 도구에 대해 기술하며, 특히 설계로부터 구현 코드를 얻는 순공학, 구현 코드로부터 설계를 얻는 역공학을 동시에 지원할 수 있는 클래스 다이어그램에 대해 기술한다. 클래스 다이어그램은 순환공학(Round-Trip Engineering)을 지원하며 설계와 구현 코드를 각각 변경했을 때 다른 곳을 즉시 변경하는 SRE(Simultaneous Round-trip Engineering)를 지

원한다.

2. 컴포넌트 생성 지원 도구

컴포넌트 생성 지원 도구는 도메인 정보를 기술하는 영역 모델러, 컴포넌트 다이어그램 및 컴포넌트 별 EJB(Enterprise Java Beans)[2] 설계를 지원하는 컴포넌트 모델러, EJB 소스 코드 생성, 전개, 시험을 지원하는 코드 생성기, 자바 시스템에서 컴포넌트를 추출하는 응용 컴포넌트 추출기, 데이터베이스와 관련된 기능 등을 제공하는 데이터베이스 연계기, 통합 GUI, 그리고 각 서브시스템에 골고루 포함되어 있는 개념적인 서브 시스템인 모델정보 관리기의 총 7 개의 서브 시스템으로 구성되어 있다[3].

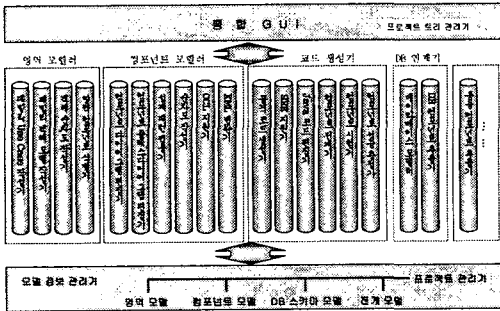


그림 1. 컴포넌트 생성 지원 도구의 블록도

- 영역 모델러 : 영역 모델러는 대상 어플리케이션이 속하는 어플리케이션이 속하는 어플리케이션 패밀리의 공통 요소들을 모델링하기 위해 패밀리에 존재하는 기능적인 단위인 Use Case들을 기술하는 Use Case 다이어그램과 패밀리에 존재하는 클래스들을 정의하는 패밀리를 클래스 다이어그램을 기술하는 기능이 있으며, 이러한 정보를 바탕으로 후보 컴포넌트들을 식별하는 기능을 제공한다.
- 컴포넌트 모델러 : 컴포넌트 모델러는 영역 분석 정보를 바탕으로 컴포넌트를 정의하고 각 컴포넌트의 인터페이스를 명확히 기술하며 또한 컴포넌트의 상세 설계를 위해 내부 클래스들을 정의할 수 있는 도구이다. UML(Unified Modeling Language)[4]의 컴포넌트 다이어그램, 클래스 다이어그램, 순차도, 설계 패턴, OCL, XML 변환 기능이 제공되며, 다양한 다이어그램을 효율적으로 사용하거나 다이어그램들 간의 산출물 공유를 원활히 할 수 있는 모델을 관리한다
- 코드 생성기 : 코드 생성기는 컴포넌트 다이어그

램의 각 컴포넌트 별로 내부 비즈니스 로직을 구현하고 컴포넌트들을 패키징하여 전개, 시험할 수 있는 환경을 제공한다. 그리고 상세 설계와 소스 코드의 일관성을 유지하기 위해 SRE 기능을 제공한다.

- 응용 컴포넌트 추출기 : 응용 컴포넌트 추출기는 기존의 자바 언어로 된 프로그램으로부터 컴포넌트를 식별하여 컴포넌트 다이어그램을 생성하고 EJB로 변환한다. 컴포넌트를 식별하기 위해서 자바 코드 분석기를 통해서 소스 코드를 분석하여 클래스 다이어그램을 생성하고, 생성된 클래스 다이어그램과 기타 분석 정보들을 바탕으로 하여 컴포넌트 후보를 제시한다. 사용자는 컴포넌트 후보 중에서 적절한 컴포넌트를 선택하여 EJB 컴포넌트로 변환한다. 변환된 컴포넌트는 컴포넌트 모델러에서 수정 및 편집할 수 있도록 컴포넌트 다이어그램 및 인터페이스 정의 형식으로 만들어져 컴포넌트 모델러에 전달된다.
- 데이터베이스 연계기 : 데이터베이스 연계기는 데이터베이스의 테이블을 생성, 관리할 수 있는 환경을 제공한다. 사용자는 데이터베이스 연계기를 통해 쉽게 DB 테이블을 생성할 수 있고 컴포넌트 설계 시에는 DB 테이블로부터 손쉽게 엔티티 빈(Entity Bean)을 생성할 수 있는 기능을 제공한다.
- 통합 GUI : 통합 GUI는 사용자가 쉽게 도구를 이용할 수 있는 인터페이스를 제공하며 사용자에게 의해 발생하는 이벤트를 처리하여 해당되는 서비스가 제공되도록 하는 그래픽 인터페이스이다.
- 모델정보 관리기 : 모델정보 관리기는 Use Case 모델, 패밀리를 객체모델을 관리하는 도메인 모델 관리기, DB 스키마 모델을 관리하는 스키마 모델 관리기, 컴포넌트 다이어그램과 각 컴포넌트를 관리하는 컴포넌트 모델 관리기, 컴포넌트 전개기에 의해 만들어진 전개 파일들을 관리하는 전개 모델 관리기로 구성된다.

3. 컴포넌트 클래스 다이어그램 편집기

컴포넌트 클래스 다이어그램 편집기는 각 컴포넌트 구성하고 있는 클래스들의 정적인 구조를 표현하기 위한 편집기로서 컴포넌트 다이어그램에 기술된 컴포넌트의 요구사항에 맞게 클래스들을 정의하고 그들 간의 관계를 기술한다. 컴포넌트 다이어그램에 기술된 컴포넌트의 인터페이스를 컴포넌트 클래스 다이어그램에 표현하기 위해 대표 세션 빈(Session Bean)을 사용한다. 세션 빈의 내용이 변경되면 자동으로 컴포넌

트의 인터페이스가 변경되고 반대의 경우도 마찬가지이다. 컴포넌트 클래스 다이어그램 편집기는 EJB 기반의 빈 클래스들을 정의하고 SRE 를 지원하여 컴포넌트 클래스 다이어그램의 클래스와 파일 시스템의 자바 파일이 일대일 대응되도록 한다.

모델 생성이벤트를 받아 뷰를 생성한다.

컴포넌트 클래스 다이어그램에서 하나의 노테이션을 생성하기 위해 복잡한 방법을 사용하지만 SRE 지원기와 같이 컴포넌트 클래스 다이어그램과 연결된 다른 블록들이 논리적인 클래스 모델로만 통합되면 되는 장점을 갖는다. SRE 지원기에 의해 클래스 모델이 생성되면 클래스 모델 생성 이벤트에 의해 뷰 모델과 뷰가 이벤트에 반응하여 새로운 클래스 노테이션을 화면에 표현한다.

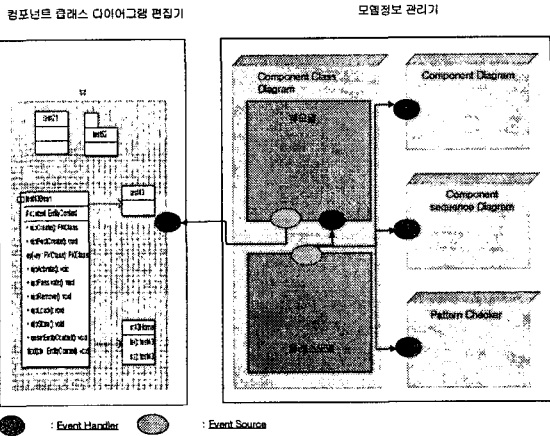


그림 2. 컴포넌트 클래스 다이어그램 구조

컴포넌트 클래스 다이어그램 편집기가 일반적인 클래스 다이어그램 편집기와 다른 점은 SRE 를 지원하기 위해 다이어그램의 클래스 모델과 실제 자바 파일의 클래스 내용이 동일하게 유지 된다는 것이다. 컴포넌트 클래스 다이어그램의 클래스 내용이 변경되면 자바 파일의 내용이 변경되고, 자바 파일이 변경되면 변경 내용이 클래스 모델에 반영되어야 한다. 이를 구현하기 위해 컴포넌트 클래스 다이어그램은 그림 2.와 같이 3 계층으로 이루어진다. 뷰는 다이어그램 편집기에 표시되는 부분으로 뷰 모델과 클래스 모델을 화면에 보여주는 역할을 담당한다. 뷰 모델은 뷰 정보를 저장하고 클래스 모델과 뷰를 서로 연결하는 역할을 담당한다. 클래스 모델은 다이어그램과는 별도로 클래스의 이름, 필드, 메소드 등의 논리적 정보만을 저장한다.

4. 컴포넌트 클래스 다이어그램 모델과 SRE 지원기

SRE 지원기는 컴포넌트 클래스 다이어그램과 EJB 소스 코드의 일관성을 유지시키는 기능을 제공한다. SRE 기능은 크게 소스 코드 변경을 컴포넌트 클래스 다이어그램에 반영하는 기능과 클래스 모델 변경을 소스 코드에 반영하는 기능을 갖는다. 만약 소스 코드의 변경이 있었다면 해당 소스 코드를 분석하여 변경된 클래스 정보를 얻은 후에 모델정보 관리기의 API 를 이용하여 해당 클래스 정보를 변경한다. 컴포넌트 클래스 다이어그램이 변경된 경우에는 변경된 클래스 모델에 해당하는 소스 코드를 분석하여 클래스 모델의 내용에 따라 소스 코드를 변경한다. 그림 3.는 SRE 지원기의 구조를 보여준다.

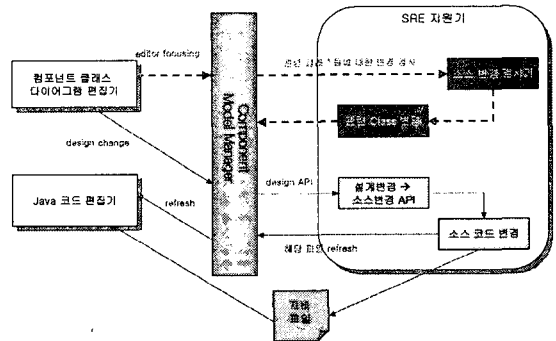


그림 2. SRE 지원기의 구조

컴포넌트 클래스 다이어그램에서 하나의 노테이션을 화면에 보이는 과정은 다음과 같다. 첫째, 편집기에서 노테이션 생성 명령을 내린다. 둘째, 편집기에서 뷰 모델에 노테이션 생성 명령을 내린다. 셋째, 뷰 모델에서 클래스 모델에 클래스 생성 명령을 내린다. 넷째, 클래스 모델 생성 후 이벤트를 발생한다. 다섯째, 뷰 모델에서 클래스 생성 이벤트를 받아 뷰 모델을 생성 후 이벤트를 발생한다. 여섯째, 에디터에서 뷰

컴포넌트 클래스 다이어그램에서 클래스 모델들은 패키지 모델 내에 생성된다. 패키지 모델 역시 클래스 모델의 일종이므로 패키지 모델들은 트리 구조를 갖게 된다. 즉, 컴포넌트 클래스 다이어그램의 클래스 모델은 그림 4.와 같이 패키지에 의한 트리 구조를 갖게 된다. SRE 지원기는 파일 시스템의 파일들을 분석하여 클래스 정보를 추출한다. 추출된 클래스 정보는 파일 시스템의 구조에 따라 (그림 3)과 같이 트리 구조를 갖는다.

컴포넌트 클래스 다이어그램의 클래스 모델과 SRE 지원기의 클래스 모델은 각각 별도의 클래스 모델을 트리 구조로 관리한다. 하지만 두개의 모델은 서로 동일한 구조와 동일한 내용을 유지해야 된다. 컴포넌트 클래스 다이어그램은 뷰 또는 다른 블록들의 요구에 의해 계속해서 모델이 변경된다. SRE 지원기 역시 도구와는 별도로 사용자에 의해 소스 코드가 변경될 수 있으므로 계속해서 모델이 변경된다. (그림 4)는 각 모델에 변경요구가 있을 경우 자신의 모델과 타 블록의 모델을 변경하는 순서를 표현한 그림이다. 컴포넌트 클래스 다이어그램은 ClassModelManger 가 모든 클래스들을 관리한다. 클래스가 생성, 삭제, 변경되면 이벤트가 발생되고 이 이벤트는 트리 상의 상위 노드로 전달된다. 결국 모든 트리의 변경은 루트 노드인 ClassModelManager 에 전달된다. ClassModelManager 는 전달된 변경 내용에 따라 SRE 지원기에 변경요구를 한다. 이 때 변경요구는 두 블록 사이의 약속된 프로토콜에 의해 이루어진다. SRE 지원기의 모델 역시 유사한 방법으로 자신의 모델을 변경하고 타 블록의 모델 변경을 요구한다. SRE 지원기는 자신이 관리하는 디렉토리를 계속해서 검사한다. 이때 변경된 파일, 또는 디렉토리를 발견하면 SRE 지원기의 모델을 변경하고 정해진 프로토콜에 의해 ClassModelManager 에 모델 변경을 요구한다.

● 연관성 단위 : 라인 추가, 삭제, 변경

컴포넌트 클래스 다이어그램 모델과 SRE 지원기의 모델 사이에 프로토콜로 해결할 수 없는 경우도 있다. 예를 들어, 클래스 다이어그램이 존재하지 않는 특정 디렉토리를 역공학을 통해 클래스 다이어그램을 생성할 수 있다. 또는 옵션에 의해 SRE 를 지원할 수도 있고 그렇지 않을 수도 있다. 이러한 경우는 트리 복사, 트리 비교 알고리즘을 적용하여 새롭게 모델을 생성한다.

5. 결론 및 향후 연구

본 논문에서는 순환공학을 지원하는 클래스 다이어그램의 설계를 기술하였다. 순환공학을 지원하는 클래스 다이어그램은 설계와 구현을 동시에 수행할 수 있도록 한다. 컴포넌트 클래스 다이어그램은 SRE 뿐 아니라 패턴분석기와 결합될 예정이다. 패턴 분석기와 클래스 다이어그램을 결합하면 특정 패턴에 따라 클래스를 생성할 수도 있고, 구현된 코드가 특정 패턴을 따르는 지도 검사할 수 있다. 현재 컴포넌트 클래스 다이어그램은 독립적으로 설계 및 프로토타입 구현이 완료된 상태이며 SRE 부분은 설계가 완료되어 프로토타입을 개발중이다. 컴포넌트 생성 지원 도구는 컴포넌트 식별, 모델링, 코드 생성, 전개 등을 지원하여 컴포넌트의 개발 전과정을 지원함으로써 컴포넌트 개발 비용을 줄일 수 있다.

[참고 문헌]

[1] 김민정, 조진희, 이우진, 신규상 “EJB 컴포넌트의 모델링 및 생성 지원 도구의 설계”, 한국정보과학회 추계학술대회논문집, 2000.
 [2] Sun, Enterprise JavaBeans Specification, Version 1.1, Sun Microsystems Inc, 1999.
 [3] ”컴포넌트 개발 지원 도구 상세설계서”, 한국전자통신연구원 2001.
 [4] OMG, “UML Specification Version 1.3”, Object Management Group, January 1999.

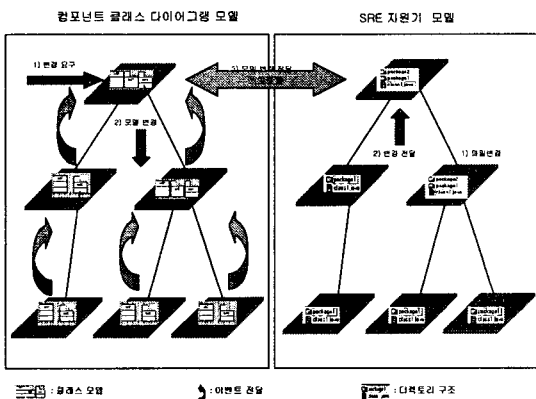


그림 4. 컴포넌트 클래스 다이어그램 모델과 SRE 지원기 모델의 동일성 보장

컴포넌트 다이어그램 모델과 SRE 지원기 사이의 프로토콜은 다음과 같다.

- 패키지 단위 : 패키지 추가, 삭제, 변경
- 클래스 단위 : 클래스 추가, 삭제, 변경
- 클래스 내부 단위 : 속성, 메소드 추가, 삭제, 변경