

## Eclipse 플랫폼을 이용한 컴포넌트 기반 개발 (CBD) 기법

김성안<sup>o</sup>, 라현정, 김수동

승실대학교 대학원 컴퓨터학과

{sakim<sup>o</sup>, hjla}@otlab.ssu.ac.kr, sdkim@ssu.ac.kr

### A Technique for Component-Based Development (CBD) using Eclipse Platform

Seong An Kim, Hyun Jung La and Soo Dong Kim

Dept. of Computer Science, Soongsil University

#### 요 약

컴포넌트 기반 개발 (CBD) 기술은 재사용을 통해 개발 노력과 시간을 줄여주는 기술로서 학계에서 활발한 연구가 이루어지고 있으며, 산업계의 기본 개발 패러다임으로 자리잡고 있다. 한편, 개방형 소프트웨어로 개발된 도구 및 어플리케이션 개발 통합 환경인 Eclipse 플랫폼이 각광 받고 있으며 Eclipse 플랫폼은 플러그인-조립이라는 형태로 구성된다. Eclipse 플러그인은 큰 재사용 단위를 가지고 조립을 통해 구성되는 CBD의 컴포넌트와 비슷한 특성을 갖는다. 기존의 부족한 특정 플랫폼 기반의 컴포넌트 설계 및 구현 지침에 관한 연구를 Eclipse 플랫폼 기반의 CBD 기법을 연구함으로써 해결할 수 있을 것이다. 따라서, Eclipse의 플러그인에 기반한 CBD 개발 기법에 대한 연구가 필요하다. 본 논문에서는 CBD 기술과 Eclipse 플랫폼에 대해서 알아보고 CBD 플랫폼과 컴포넌트의 요구사항을 분석하고 이러한 요구사항들이 어떠한 기법을 통하여 Eclipse 플랫폼에 적용 가능한지에 대해 알아본다. 끝으로 기존의 CBD 플랫폼에서의 개발과 비교하여 Eclipse 플랫폼을 사용함으로써 얻을 수 있는 장점에 대해서 알아본다.

## 1. 서론

컴포넌트 기반 개발 (CBD) 기술은 재사용을 통해 개발 노력과 시간을 줄여주는 기술로서 학계에서 활발한 연구가 이루어지고 있으며, 산업계의 기본 개발 패러다임으로 자리잡고 있다 [1]. 한편, Eclipse는 개방형 소프트웨어로 개발된 도구 및 어플리케이션 개발 통합 환경으로서 크게 각광받고 있다. Eclipse는 재사용 단위가 큰 플러그인의 조립을 통해 어플리케이션을 생성한다 [2]. 이러한 Eclipse 플러그인의 특성은 컴포넌트의 특성과 비슷하다. 그러므로, 기존의 CBD 관련 연구를 활용하여 Eclipse 플랫폼 기반의 플러그인 개발에 적용하는 것이 가능하다. 또, Eclipse 플랫폼 기반의 CBD 기법을 연구함으로써 기존의 부족한 특정 플랫폼에 적용한 상세 설계 및 구현 지침에 관한 연구를 할 수 있을 것이다. 따라서, 개방형 플랫폼인 Eclipse의 플러그인에 기반한 컴포넌트 개발 기법에 대한 연구가 필요하다.

본 논문에서는 CBD 플랫폼과 컴포넌트의 요구사항을 알아보고 이러한 특성을 Eclipse 플랫폼과 플러그인에 적용시키는 기법을 알아보고 이를 통해 얻을 수 있는 장점에 대해 알아본다.

## 2. 기반 연구

### 2.1. 컴포넌트 기반 개발 (CBD) 기술 [3]

CBD 기술은 재사용 가능한 컴포넌트를 조립하여 효율적으로 소프트웨어를 개발함으로써 개발 노력과 상품화 시간을 줄여주는 기술이다. CBD는 크게 컴포넌트 개발 단

계와 컴포넌트 조립을 통해 어플리케이션을 생성하는 컴포넌트 기반 소프트웨어 개발 단계로 나뉜다.

컴포넌트 개발 단계에서는 컴포넌트가 사용되는 어플리케이션에 따라 변하지 않는 공통성과 어플리케이션에 따라 특화되어야 하는 가변성을 식별하여 컴포넌트를 개발한다. 컴포넌트 개발 단계에서 작성된 컴포넌트는 CBD에서 재사용의 기본 단위이고 객체보다 상대적으로 큰 기능을 제공한다. 컴포넌트는 인터페이스와 구현을 분리함으로써 내부 구현을 숨기고 이를 통해 동일한 인터페이스를 가진 다른 컴포넌트로 교체 가능한 특성을 가진다. 컴포넌트 기반 소프트웨어 개발 단계에서는 어플리케이션에 알맞은 컴포넌트를 식별하고 이를 조립하는 과정으로써 식별된 컴포넌트는 커스터마이제이션 기법을 이용하여 어플리케이션에 맞게 특화 한다.

### 2.2. Eclipse 플랫폼의 개요 [4]

Eclipse는 개방형 소프트웨어로서 확장 가능한 개발 플랫폼이다. Eclipse 플랫폼을 구성하고 있는 대부분은 플러그인이다. 그러므로 플러그인의 조립을 통해 Eclipse 플랫폼을 확장해 새로운 어플리케이션을 생성하는 것이 가능하다. Eclipse 플랫폼은 크게 통합 개발 환경과 플러그인 개발 환경으로 구성된다.

Eclipse 플랫폼의 통합개발 환경은 기본적으로 자바언어 개발 툴 키트 (JDT)을 포함하고 있으며 C/C++ 언어 개발 툴 키트 (CDT) 및 다른 언어에 대한 추가환경이 플러그인으로 제공되고 있다. 플러그인 개발환경 (PDE)은 Eclipse 플랫폼의 기능을 확장할 수 있는 플러그인을 개발하도록 특별한 형태의 개발환경이다. PDE는 플러그인 개발을 위한 각종 도구를 지원한다.

### 3. CBD 플랫폼과 컴포넌트의 요구사항

#### 3.1. CBD 플랫폼으로서의 요구사항

본 절에서는 CBD 플랫폼으로서 가져야 하는 요구사항이 무엇인지 알아보고 이를 설명한다.

표 1. CBD 플랫폼 요구사항

CBD 플랫폼 요구사항	설명
컴포넌트 개발 명세	개별적인 컴포넌트 개발을 위한 명세
상호교환 명세	컴포넌트와 컴포넌트 사이의 메시지 전송 메커니즘
조립 명세	컴포넌트와 컴포넌트의 조립이 어떻게 이루어져야 하는지에 대한 명세
커스터마이제이션 메커니즘 명세	수정 없이 컴포넌트의 기능을 확장할 수 있는 방법에 대한 명세

표 1에서 요약하였듯이 CBD 플랫폼은 개별적인 컴포넌트의 개발 시 준수해야 하는 명세를 제시하여야 한다. 예를 들면, 썬 마이크로시스템즈의 EJB에서 '지속성을 관리하는 엔티티빈의 경우 반드시 javax.ejb.EntityBean 인터페이스를 구현해야 한다.'와 같이 컴포넌트를 개발단계에서 준수해야 하는 명세를 제공하여야 한다. 또, CBD 플랫폼은 컴포넌트와 컴포넌트 간의 메시지 교환이 이루어지는 방식을 명세해야 한다. 이를 통해, 컴포넌트 간의 조립 시 컴포넌트 간의 메시지 교환이 이루어진다. 컴포넌트 조립은 컴포넌트를 이용하여 어플리케이션을 생성하는 과정에서 사용되는 기법으로 컴포넌트와 컴포넌트의 조립이 어떻게 이루어져야 하는지에 대한 명세가 제시되어야 한다. 또한, 컴포넌트는 조립과정에서 어플리케이션에 특화된 정보를 받아들이 수 있어야 한다. 이러한 어플리케이션에 따라 변화하는 성질을 가변성이라고 한다. 가변성은 크게 제한적인 선택적 요소를 가지는 닫힌 경우와 받아들일 수 있는 값이 제한적이지 않은 열린 경우가 있다 [5]. 커스터마이제이션 메커니즘 명세에서는 컴포넌트의 수정 없이 닫힌 경우와 열린 경우의 가변성을 모두 설정 할 수 있는 방식이 제시 되어야 한다.

현재의 J2EE나 .NET과 같은 CBD 플랫폼에서는 커스터마이제이션 메커니즘에 대한 정의가 되어 있지 않다.

#### 3.2. 컴포넌트 요구사항

본 절에서는 컴포넌트 개발 단계에서 만들어지는 컴포넌트가 가져야 하는 요구사항이 무엇인지 알아보고 이를 설명한다.

표 2. 컴포넌트 요구사항

컴포넌트의 요구사항	설명
인터페이스와 구현의 분리	인터페이스와 구현을 분리하여 내부 구현을 숨긴다.
대체성	컴포넌트는 동일한 인터페이스를 제공하는 다른 컴포넌트로 대체 가능하다.
큰 기능성	컴포넌트는 객체보다 큰 기능성을 가진다.

표 2에서 요약하였듯이 컴포넌트는 인터페이스와 구현을 분리함으로써 내부 구현을 숨길 수 있다. 내부 구현을 숨김으로써 내부 구현의 변경이 외부에 미치는 영향을 없앨 수 있고 이러한 특성을 이용하여 컴포넌트를 인터페이스가 같은 다른 컴포넌트로 대체 가능하다. 또, 컴포넌트는 내부의 워크플로우를 숨기고 외부에 객체보다 큰 기능성을 가지는 인터페이스만을 공개할 수 있다. 따라서, 사용자는 객체 단위의 기능성이 아닌 컴포넌트가 제공하는 커다란 기능성에 초점을 맞출 수 있다.

현재의 J2EE나 .NET과 같은 CBD 플랫폼은 위의 요구사항을 다 만족하지만 가변성에 대한 정의가 되어 있지 않기 때문에 재사용 가능한 큰 기능성의 지원에는 한계가 있다.

### 4. Eclipse 플랫폼에서의 CBD 기법

#### 4.1. Eclipse 플랫폼에서의 CBD 플랫폼 요구사항

본 절에서는 3.1 절에서 알아본 CBD 플랫폼의 요구사항이 어떠한 기법을 통해 Eclipse 플랫폼의 특성으로 반영할 수 있는지 알아본다.

표 3. CBD 플랫폼의 요구사항과 Eclipse 플랫폼에서의 표현

CBD 플랫폼	Eclipse 플랫폼에서의 활용방법
컴포넌트 개발 명세	플러그인 개발 명세
상호교환 표준	확장과 자바 코드를 통한 상호교환
조립 표준	매니페스트 파일을 통한 조립
커스터마이제이션 메커니즘	확장점 제공을 통한 커스터마이제이션

컴포넌트 개발 명세는 Eclipse 플랫폼의 플러그인을 개발 시 지켜야 하는 플러그인 개발 명세로 볼 수 있다. 'Eclipse 플러그인은 반드시 plugin.xml이라는 매니페스트 파일에 선언되어야 하며 반드시 org.eclipse.runtime.core.Plugin 클래스를 상속 받은 클래스를 포함한다.'와 같은 플러그인을 체계적인 방식으로 개발할 수 있는 표준 명세가 제시된다.

상호교환은 확장점과 자바 코드를 통해서 이루어 질 수 있다. 매니페스트(manifest) 파일에 의존되는 플러그인의 확장을 정의하고 확장한 플러그인의 공개 인터페이스를 호출해 플러그인 간의 상호교환을 할 수 있다.

그림 1에서 볼 수 있듯이 Eclipse 플랫폼은 플러그인의 조립을 통해 구성된다. 플러그인을 통한 CBD 기법을 적용한 어플리케이션 개발 시 불필요한 플러그인은 플랫폼을 구성하는 요소라 해도 제외하는 것이 가능하다. 플러그인의 조립을 통한 어플리케이션 개발 시 아키텍처에 대한 결정이 이루어져야 한다.

커스터마이제이션 메커니즘은 플러그인 제작 시 확장점을 제공하고 확장점에 어플리케이션 특화된 정보를 가진 플러그인을 채움으로써 열린 가변성을 설정 할 수 있다. Eclipse 플랫폼은 플러그인 기법을 통해 열린 가변성을 위한 표준 처리 메커니즘을 제공하고 있다.

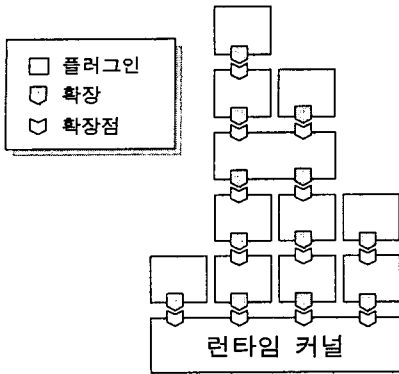


그림 1. Eclipse 플랫폼 구조

#### 4.2. Eclipse 플러그인의 컴포넌트 요구사항

본 절에서는 3.2 절에서 알아본 컴포넌트의 요구사항이 어떠한 기법을 통해 Eclipse 플러그인의 특성으로 반영 할 수 있는지 알아본다.

표 4. 컴포넌트 요구사항과 Eclipse 플러그인

컴포넌트	Eclipse 플러그인의 활용방법
인터페이스와 구현의 분리	선언과 구현의 분리
대체성	선언과 구현의 분리, 약한 결합, 고유한 아이디와 버전 정보 사용
큰 기능성	프래그먼트, 플러그인, 휘쳐 지원

표 4에서 요약 하였듯이 컴포넌트가 가져야 할 요구사항은 Eclipse 플러그인으로 표현 가능하다.

인터페이스와 구현의 분리는 Eclipse 플랫폼의 매니페스트 파일을 통한 선언과 자바언어로의 구현으로 표현이 가능하다.

컴포넌트가 가져야 하는 대체성은 Eclipse 플러그인의 선언과 구현의 분리와 매니페스트 파일을 통한 약한 결합을 통해서 가능하다. 선언과 구현을 분리함으로써 구현의 변화가 전체 시스템에 미치는 영향을 줄일 수 있고 매니페스트 파일을 통한 약한 결합을 통해 플러그인과의 의존성을 줄일 수 있다. 또, 고유한 아이디와 버전 정보를 가짐으로 매니페스트 파일에서 이의 수정을 통한 대체가 가능하다.

Eclipse 플러그인은 프래그먼트, 플러그인, 휘쳐의 지원을 통해 모듈성을 향상시키고 큰 기능성을 제공할 수 있다. 프래그먼트는 불완전한 플러그인으로 다른 플러그인과의 조립을 통해 이용될 수 있고, 플러그인은 독립된 기능성을 가지는 배포단위이고 휘쳐는 플러그인의 조립으로써 어플리케이션이 될 수 있다. 이렇듯 다양한 크기의 배포단위를 가짐으로써 내부의 워크플로우를 숨기고 큰 기능성을 가지고 배포될 수 있다.

#### 5. Eclipse 플랫폼을 이용한 CBD 장점

앞 절에서는 Eclipse 플랫폼에서의 CBD 기법에 대해서 알아보았다. 본 절에서는 Eclipse 플랫폼에서의 CBD를 통해 얻을 수 있는 장점에 대해서 알아본다.

Eclipse 플랫폼은 열린 가변성에 대한 단일화된 처리기법을 제공하는 장점을 가진다. 기존의 CBD 플랫폼은 가변성 처리 기법이 확립되지 않았기 때문에 컴포넌트 개발자에 따라 가변성 처리기법이 다양한 문제점을 가졌다. Eclipse 플랫폼을 이용할 경우 확장점을 정의하여 어플리케이션에 특화된 정보를 플러그인으로 입력 받아 열린 가변성을 처리한다. 또, Eclipse 플랫폼을 이용한 컴포넌트 개발 시 기존의 자바 라이브러리를 사용할 수 있는 장점이 있다. 이미 널리 산업계에서 사용되고 있는 자바의 다양한 상용, 공개용 라이브러리를 획득하여 개발 비용의 절감과 개발 시간의 단축을 이룰 수 있다. 또한, 개발에 필요한 상용 컴포넌트의 획득을 위한 시장이 활발하게 형성되어 있는 장점이 있다. Eclipse는 Common Public License (CPL) [7]를 따르므로 Eclipse 플랫폼은 개방형 소프트웨어이지만 개발된 플러그인은 공개되지 않고 상업적인 용도로써 사용될 수 있다. 이러한 환경을 기반으로 다수의 Eclipse 플러그인의 판매가 활발하게 이루어지고 있다 [8].

#### 6. 결론 및 향후 연구과제

지금까지 CBD 플랫폼에서의 요구사항과 컴포넌트의 요구사항에 대해 알아보았고 이러한 요구사항이 Eclipse 플랫폼과 플러그인에서 어떻게 표현할 수 있는지와 Eclipse 플랫폼과 플러그인에서의 지원성에 대해서 알아보았다. 끝으로 Eclipse 플랫폼을 이용한 CBD를 통해 얻을 수 있는 장점에 대해서 알아보았다.

Eclipse 플랫폼에서의 CBD 기법 연구를 통해 부족한 특정 플랫폼에 관한 연구를 할 수 있고 기존의 CBD 연구에서 제시된 이론을 적용할 수 있는 장점이 있다.

앞으로 본 논문에서 제시한 Eclipse 플랫폼을 통한 CBD 기법을 적용 시키기 위한 프로세스와 지침이 정의되어야 하고 이를 검증하는 사례연구가 추가 되어야 할 것이다. 또한, 부족한 닫힌 가변성에 대한 컴포넌트 커스터마이제이션 기법에 관한 연구가 필요하다.

#### 7. 참고문헌

- [1] Kim, s., "Lessons learned from a nationwide CBD promotion project", *Communications of the ACM*, Vol. 45, No. 10, pp 83-87, 2002.
- [2] Gamma, E. and Beck, K., *Contributing to eclipse*, Addison-Wesley, 2003.
- [3] Heineman and Council, *Component-Based Software Engineering*, Addison Wesley, 2001.
- [4] Eclipse Web Site, <http://www.eclipse.org/>
- [5] Kim, s., Min, h. and Rhew, s., "Variability Design and Customization Mechanisms for COTS Components," *LNCS 3480*, pp.57-66, 2005.
- [6] Apache Ant Web Site, <http://ant.apache.org/>
- [7] Common Public License (CPL), <http://www.opensource.org/licenses/cpl1.0.php>
- [8] Commercial Projects and Plug-ins Based on Eclipse, <http://www.eclipse.org/community/commercialplugins.html>