

빈 인트로스펙션 메커니즘을 이용한 자바 소스 코드 생성기 구현

손승우*, 김창갑, 김홍남

한국전자통신연구원 컴퓨터·소프트웨어기술연구소

An Implementation of Java Source Code Generator using Bean Introspection Mechanism

Seungwoo Son, Changkap Kim, Heungnam Kim
ETRI-Computer & Software Technology Lab.

요약

자바빈 컴포넌트를 이용한 RAD 개발 도구들은 일반적으로 자바빈 컴포넌트를 시각적으로 배치하고, 속성 및 이벤트를 편집할 수 있는 개발 환경을 제공한다. RAD 개발 도구들은 배치된 컴포넌트에 대한 속성과 이벤트를 편집할 수 있는데 특히 이벤트는 개발자의 응용 논리가 포함된 부분으로 완전한 시각화를 통한 구현이 어렵다. 본 논문에서는 자바빈 컴포넌트 기반 개발 환경에서 자바 언어의 인트로스펙션(Introspection) 메커니즘을 이용한 자바 소스 코드 생성 기법을 제시한다. 자바 소스 코드의 생성은 태그로 표시된 템플릿 코드의 생성과 자바의 위임 이벤트 모델을 이용한 이벤트 처리 코드를 생성하여 템플릿 코드에 합성하는 방법을 따른다. 본 논문에서 제시하는 기법은 인트로스펙션에 의해 추출된 정보를 개발자에게 시각적으로 보여주며 개발자의 시각적 조작에 따라 이벤트에 대한 소스 코드를 자동으로 생성해준다.

1. 서론

자바빈과 같은 컴포넌트의 개념은 IDE나 RAD 도구의 효율성과 사용성에 전적으로 의존한다. IDE는 텍스트 기반의 프로그래밍에서 시각적인 컴포넌트 조작 쪽으로 빠르게 변하고 있다 [1][2][3]. 비주얼 베이직, 파워빌드, 비주얼 카페와 같은 상용의 IDE들은 보통 다음의 기능 요소들을 가지며, 응용 논리를 추가할 수 있게 VBScript나 PowerScript와 같은 스크립트 언어를 가지고 있다 [4][5][6]

- 사용 가능한 컴포넌트를 보여주는 하나 또 그 이상의 팔레트(아이콘으로 표현됨)
- 컴포넌트가 위치하고 드래그앤드롭이나 팝업 메뉴 등을 통하여 상호 연결될 캔버스 컨테이너(폼 디자인너로 표현됨)
- 개발자가 컨테이너 내의 컴포넌트를 사용자 정의할 수 있게 하는 속성 및 스크립트 편집기

일반적으로 RAD 계열의 자바 응용 개발 도구는 폼 디자인너에 의해 생성된 폼에 대한 소스 코드를 기본적으로 생성한다. 이렇게 템플릿으로 생성된 코드는 사용자가 추가로 폼상의 컴포넌트의 속성을 변경하거나, 이벤트를 사용자가 정의할 수 있어야 하며, 이에 대한 변경도 소스 코드에 반영하여야 한다. RAD 도구들의 필수 기능 요소 중 응용의 생산성을 극대화하려면 컴포넌트간의 이벤트 처리를 시각화하고 이에 대한 소스 코드를 자동 생성하여 응용 논리의 코딩을 도와주는 기술

이 필요하다. 자바빈을 이용한 응용은 속성 및 이벤트를 자바의 인트로스펙션 메커니즘을 이용하면 쉽게 그 정보를 추출할 수 있다 [1][2]. 추출된 정보를 어떻게 시각적으로 개발자에게 보이고 개발자의 입력을 받아서 이를 소스 코드에 반영하는 것은 소프트웨어의 생산성 향상에 많은 도움을 준다.

자바로 구현된 소스 코드는 순수 객체지향 코드로서 징형화된 구조를 이용하고 자바의 인트로스펙션을 사용해 추출한 정보를 시각적으로 속성/이벤트 편집기를 써서 쉽게 응용 논리를 자동 생성하는 소스 코드 생성기를 제안한다. 제안된 생성기를 자바를 기반으로 구현된 RAD계열 자바 응용 개발 도구인 ShakeHands [1]에 연동하여 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 자바빈 컴포넌트에서 속성 및 이벤트를 추출하는 방법을 살펴보고 3장에서는 자바의 인트로스펙션 메커니즘을 이용한 개발 도구의 기본 구조를 제시하고 4장에서 결론과 향후 연구 방향을 기술한다.

2. 관련 연구

본 절에서는 IDE에서 컴포넌트의 사용자 정의 방법과 이것을 자바빈에서는 어떻게 처리할 수 있는지 살펴본다. 또한 이벤트 처리에 대한 소스 코드 생성을 위한 자바의 위임(delegate) 이벤트 모델을 살펴본다.

2.1 통합 개발 환경과 속성/이벤트의 사용자 정의

개발자는 그림 1에 나타난 바와 같이 폼 디자이너와 속성 및 이벤트 편집기를 이용하여 속성 값을 설정하고 빈의 이벤트에 대한 새로운 핸들러를 제공함으로써 빈 컴포넌트를 사용자 정의할 수 있다. 빈은 IDE 메뉴나 편집기와 같이 명시적으로 수정되거나 폼 상에 빈을 배치하는 것과 같은 시각적 조작의 효과로써 수정된다. 두 방법 모두 IDE는 빈의 메타데이터를 접근해서 메타데이터를 어떻게 보여줄 것인지 해당 속성이나 트래젝트 이벤트 처리할 편집기를 활성화시키는지 알고 있어야 한다 [5].

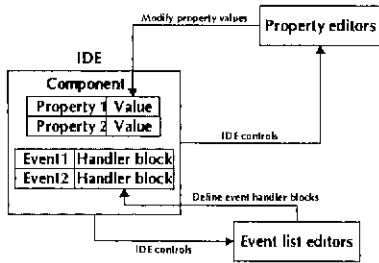


그림 1 통합개발환경에서 컴포넌트의 사용자 정의

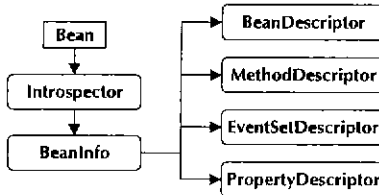


그림 2 'Bean' 클래스의 정보 추출 과정

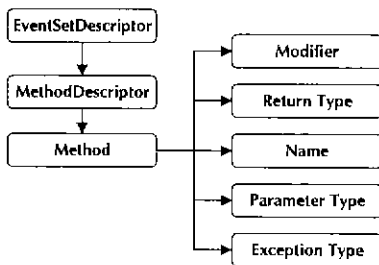


그림 3 'EventListener' 메소드의 정보 추출 과정

2.2 자바빈즈 [2]

자바를 이용한 프로그램을 재사용 가능하게 한 컴포넌트 소프트웨어로서 비주얼 개발 도구를 이용해 컴포넌트 소프트웨어가 제공하는 속성이나 이벤트 정보를 추출할 수 있고, 속성을 설정해줄 수 있는 설계 시점에서 프로그래밍이 가능한 컴포넌트이다. java.beans 패키지 형태로 제공되는 API는 이러한 IDE 환경에서 필요한 빈의 정보(속성/이벤트) 추출과 속성/이벤트 편집기

에 관한 API를 제공하여 빈을 기반한 응용의 개발을 쉽게 할 수 있다 그림 2는 인트로스펙터를 통한 빈 정보 추출과정을 나타낸 것이다. 그림 3은 추출된 정보 중 이벤트에 대한 처리 구조이다. 추출된 이벤트 정보를 이용하여 소스 코드에 반영하려면 자바의 위임 이벤트 모델의 이해가 필요하다.

2.3 자바의 이벤트 모델

자바빈즈의 이벤트 통지 매커니즘은 3개의 자바 클래스 인터페이스인 이벤트(Event), 이벤트소스(EventSource), 이벤트리스너(EventListener)로 되어있다. 이벤트소스는 등록된 모든 이벤트리스너에게 통보되며, 관심사에 있는 이벤트가 발생하면 각 이벤트 객체를 전달한다. 이 3개의 클래스들은 그림 4와 같은 위임 모델로 표현되어진다 [2][5]. 이벤트 모델은 사용의 용이성을 향상시키는 두 개의 또 다른 기능을 지원한다. 이벤트소스는 멀티캐스트(multicast)가 기본이나 유니캐스트(unicast)로도 설정할 수 있다. 즉, 최대 하나의 이벤트리스너와 이벤트어댑터(EventAdapters)가 더해져서 개발자가 하나의 리스너 인터페이스에 정의된 모든 이벤트들에 대한 핸들러를 코딩해야 하는 부담을 덜어준다.



그림 4 자바의 위임 이벤트 모델

3. 자바 소스 코드 생성기의 구조

3.1 전체 구조

소스 코드 생성기의 전체 구조는 크게 템플릿 소스 코드 생성기인 Source와 전체 폼의 자료 구조를 나타내는 JCode, JCode의 변경에 따른 소스 코드를 만들어주는 SourceMaker와 컴포넌트의 이벤트를 연결해주는 InteractionDialog로 구성된다. 이외에 생성된 소스 코드를 보여주는 EditorPanel이 있다 폼 데이터인 JCode는 자바의 감시자 패턴을 이용하여 설계 및 구현하였고 InteractionDialog는 스윙(Swing)을 이용하여 구현하였다. 이들은 모두 JDK1.2.2를 이용하여 윈도우 NT4.0 상에서 구현하였다. 그림 5는 이러한 구성 요소들의 클래스 다이어그램을 나타낸 것이다.

3.2 템플릿 코드 생성

자바의 AWT나 스윙 컴포넌트를 이용한 응용은 크게 웹브라우저에서 실행할 수 있는 애플릿과 단독으로 실행될 수 있는 애플리케이션으로 구별된다. 이러한 자바 응용은 특정한 몇 개의 영역으로 구분되어 소스 코드가 작성된다. 컨트롤 초기화 부분, 컨트롤 선언부(자바 애플리케이션의 경우 매뉴도 포함), 매뉴 초기화 부분(자바 애플리케이션의 경우), 리스너 등록 부분, 이벤트 처리를 위한 코드(내부 클래스로 표현됨)로 구분되

는데 이들을 각각 특정 주석(//{{ 영역이름})을 경계로 하여 템플릿 코드를 생성한다. 이 주석을 태그로 하여 생성된 코드는 개발자가 속성/이벤트 편집기를 통하여 그 값을 변경하면 템플릿 코드 내에 그 위치를 찾아 소스 코드를 변경한다.

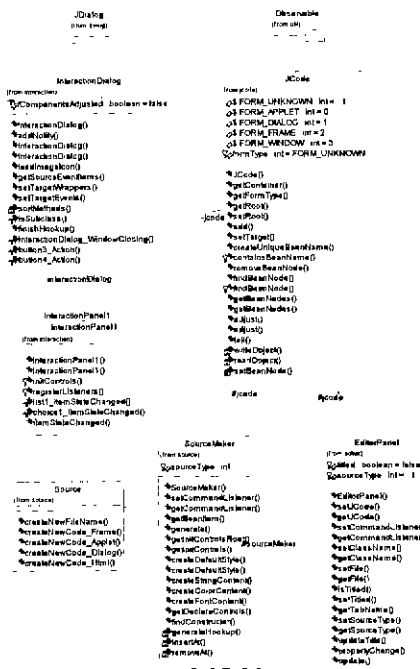


그림 5 소스 코드 생성기의 클래스 다이어그램

3.5 이벤트 설정 및 이벤트 처리 소스 생성

그림 6은 컴포넌트간의 이벤트를 설정하기 위한 예로 2개의 버튼간의 이벤트를 설정하기 위한 화면이다. 시각적으로는 크게 3부분으로 나뉘어지는데 먼저 개발자는 이벤트 근원 컴포넌트(button1)의 시작 이벤트를 선택한다. 그 다음 중간에 있는 이벤트 대상 컴포넌트(button2)의 리스너에 해당하는 메소드 리스트가 나타난다. 사용자는 원하는 메소드를 선택하여 두 컴포넌트간의 이벤트를 설정해준다.

이렇게 설정된 이벤트는 자바빈의 이벤트 리스트에 등록이 되므로 이들을 이용하여 이벤트어댑터 클래스를 작성하여 소스 코드 내에 삽입한다.

```
hook0 = new MyActionAdaptors();
button1.addActionListener(hook0);
```

위에서 MyActionAdaptors 클래스는 자바의 내부 클래스(Inner Class)로 소스 코드를 생성하여 삽입하고 개발자가 설정한 이벤트에 대한 구현 코드를 추가한다.

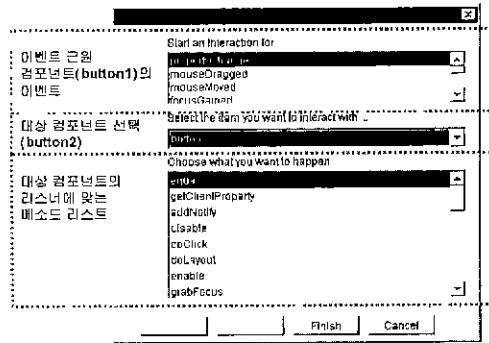


그림 6 컴포넌트간의 이벤트 설정 화면

4. 결론

본 논문에서는 자바의 인트로스펙션 메커니즘을 이용하여 RAD 도구에서 생성된 폼 화면의 소스를 자동 생성하고 속성 및 이벤트 편집기의 변화에 따른 소스 코드의 변경을 할 수 있는 자바 소스 코드 생성기를 제안하였다. 자바 애플릿 및 애플리케이션은 일반적으로 컨트롤의 초기화 부분, 컨트롤의 선언부, 리스너의 등록 부분 및 이벤트어댑터에 대한 코드 부분으로 구분이 되므로 먼저 폼 화면에 대한 템플릿을 생성하고 속성 및 이벤트 편집기의 설정에 따라 해당 소스 코드를 생성하여 템플릿 코드의 해당 부분에 삽입한다. 본 논문에서 제안한 소스 코드 생성기를 이용하면 자바빈 컴포넌트를 이용한 RAD 개발도구에서 속성 및 이벤트 편집기를 통한 자동 소스 코드 생성이 가능하므로 프로그램의 생산성을 높일 수 있다. 향후 TWA(Two-way Access) 프로그래밍이 되려면 버튼의 사이즈, 색깔과 같은 속성을 사용자가 소스 코드에 직접 수정했을 때 폼 상에 반영되어야 한다.

5. 참고 문헌

- [1] 김대현, 유철중, 장옥배, 김창갑, 이상덕, "Java의 객체 및 컴포넌트 정보 추출을 위한 Object Inspector," *정보과학회 가을 학술발표논문집*, 제 24권 제 2호, pp. 293-296, 1997.
- [2] Software Engineering Institute. *Software Technology Reference Guide*, CMU/SEI-97-HB-001, Jan 1997.
- [3] L. Vanhelsuwe, *Inside Secrets JavaBeans*. SYBEX, 1998.
- [4] John C. Zubeck, "Implementing Reuse with RAD Tools' Native Objects," *IEEE Computer*. Vol. 30, No. 10, pp. 60-65, Oct 1997.
- [5] David Krieger, Richard M. Aider, "The Emergence of Distributed Component Platforms," *IEEE Computer*, Vol 31, No. 3, pp. 43-53, Mar 1998.
- [6] D. S Linthicum, *David Linthicum's Guide to Client/Server and Intranet Development*, John Wiley & Sons, Inc., 1997.