제26회 한국정보처리학회 추계학술발표대회 논문집 제13권 제2호 (2006. 11)

GUI 편집기 재목적을 통한 GUI 개발환경 구현 방법

장한일*, 우균*, 김원영**, 최완**
*부산대학교 컴퓨터공학과
**한국전자통신연구원 디지털홈연구단
e-mail:daystar@pusan.ac.kr

Implementation Methods of GUI Development Environments Using GUI Editor Retargeting

Hanil Jang*, Gyun Woo*, Won-Young Kim**, Wan Choi**
*Dept. of Computer Engineering, Pusan National University
**Digital Home Research Division, Electronics and Telecommunication Research Institute

요 약

응용프로그램의 쉽고 빠른 개발을 위한 개발도구의 중요성이 대두되고 있다. 이 중 GUI 개발환경은 개발자의 편의성과 신속한 개발을 위해서 필수적이나 다양한 언어와 플랫폼을 위한 GUI 개발환경을 개발하는데 어려움을 겪고 있는 실정이다. 그리하여 본 논문에서는 여러 언어와 플랫폼을 위한 GUI 개발환경을 신속히 얻기 위해 기존의 GUI 편집기를 재목적하여 GUI 개발환경을 구현하는 방법을 제안한다. 기존에 개발된 FarPy GUI 편집기를 전단부로 하고 GUI를 구성하는 GroovyMarkup 코드를 생성하는 방법을 사용하여 Groovy를 위한 GUI 개발환경을 쉽고 빠르게 얻을 수 있었다.

1. 서론

90년대 이후 소프트웨어의 주 사용자층이 전문가 집단에서 일반 사용자로 옮겨가고 있다. 컴퓨터에 대한 전문 지식이 없는 일반인이 소프트웨어의 주 소비자로 변하면서 사용 편의를 위한 GUI(Graphical User Interface) 개념이 창시되었으며, 현재 대부분의 응용 프로그램이 GUI 기반으로 작성되고 있다. 한편 소프트웨어의 주 사용자 층 뿐만 아니라 소프트웨어의 개발 측면에서도 일반인의 참여가 커지고 있다. 이러한 추세에서 기존의 대형 소프트웨어 제공자들은 일반 사용자를 위한 플랫폼과 환경을 제공하여 참여를 유도하는 방법으로 자사 플랫폼의 보급율과 영역을 넓히는 전략을 추진하고 있다.

이러한 상황에서 특정 언어와 플랫폼의 성공적인 대중화를 위해서는 개발자의 편의를 제공하여 사용자의 참여를 유도해야 한다. 이를 위해서는 일반 응용 프로그램의 쉬운 개발을 돕는 개발환경이 반드시 필요하여 현재 각 플랫폼을 위한 쉬운 개발환경에 대한 연구·개발이 활발히 진행되고 있다.

하지만 현재 많은 프로그래밍 언어와 플랫폼이 개발되고 있어 이를 위한 개발환경을 일일이 개발하는데 많은 어려움을 겪고 있다. 특히, GUI를 구성하는 작업은 GUI 라이브러리의 습득 비용과 GUI 배치를 위한 단순 코드로 이루어지기 때문에 개발도구의 지원이 절실한 부분이지만 각 플랫폼마다 새로운 GUI 개발환경을 개발하는데 어려움이 있다.

본 논문에서는 기존에 개발된 WYSIWYG 형태의 GUI 편집기를 이용하여 GUI 개발환경을 개발하는 방법을 제안한다. 그리고 이를 이용하여 Groovy를 위한 GUI 개발환경의 프로토타입을 구현한다. 본 논문에서 제안하는 방법은 새로운 플랫폼이나 언어를 위한 GUI 프로그램 개발환경의 개발에 도움이 될 것이다.

2. 연구 배경

2.1 GUI 개발환경 현황

현재 대형 소프트웨어 제공자들이 제공하는 Microsoft의 Visual Studio 통합개발환경, Borland의 Delphi 통합개발환경, Sun의 NetBeans와 같이 통합

개발환경에 포함되는 GUI 개발도구를 GUI 개발환경으로 많이 사용하고 있다. 그 외 Java를 이용한 개발에서 많이 사용되는 Eclipse GUI 편집기 플러그인이나 다른 기타 플랫폼이나 언어를 위한 여러 GUI 편집기들이 오픈소스로 개발되고 있다.

최근 소프트웨어의 요구사항이 다양해지고 빠르게 변해감에 따라 소프트웨어 개발 생산성 향상을 위해 소프트웨어의 빠른 개발이 요구되고 있다. 또한 Web 2.0의 트렌드에 맞추어 기존 서비스의 소비자였던 일반 사용자들이 서비스의 생산자로 참여함에 따라 해당 플랫폼을 위한 소프트웨어의 쉬운 개발이 요구되고 있다. 이러한 경향에 맞추어 서비스를 제공하던 대형 소프트웨어 공급자들이 자사가 제공하는 플랫폼의 사용자를 모으기 위해 개발의 편의를 제공하고 쉬운 개발을 지원하는 개발도구를 제공하고 있으며 쉽고 빠른 소프트웨어 개발을 위한 개발환경에 관한 연구가 진행되고 있다.

2.2 GUI 개발환경의 필요성

새로운 플랫폼이나 언어의 성공을 위해서는 소프트웨어의 쉽고 빠른 개발을 지원하는 개발환경을 제공해야 한다. 특히 GUI 구성은 새로운 GUI 개발을 위한 라이브러리 습득의 비용과 GUI 구성과 배치를 위한 단순 작업의 비용이 큰 부분이다. 현재 대부분의 응용 프로그램이 GUI 형태로 작성되고 있음을 감안할 때 GUI 구성 부분을 쉽고 빠르게 개발할 수 있도록 지원하는 것을 개발환경의 필수적인 요소로 볼 수 있다.

3. GUI 편집기 재목적을 통한 GUI 개발환경 설계

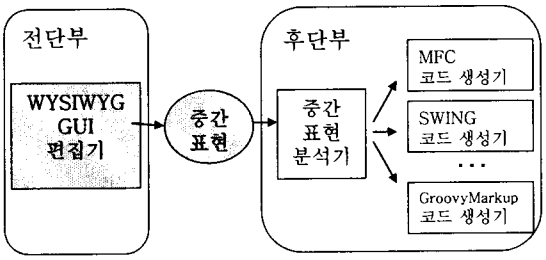
3.1 GUI 개발환경 구현 방법

GUI 개발도구는 GUI 컴포넌트의 배치와 구성을 도와주는 GUI 편집기와 실행 가능한 GUI 프로그램 코드를 생성하는 코드 생성기로 크게 나눌 수 있다. 여기에서 GUI 편집기는 GUI 컴포넌트의 배치 즉, 레이아웃을 결정하는 요소이기 때문에 WYSIWYG 형태의 GUI 편집을 지원해야 한다. 일반적으로 GUI 라이브러리에서 제공하는 GUI 컴포넌트를 WYSIWYG 형태로 편집하는 편집기를 일일이 구현하는 것은 개발에 부담이 된다.

컴파일러와 마찬가지로 GUI 개발도구도 전단부와 후단부로 나눌 수 있다. GUI 편집기는 전단부에 해당하며 GUI 코드 생성기는 후단부에 해당한다. GUI를 구성하는 버튼, 대화창 등의 컴포넌트는 여러 GUI 플랫폼마다 유사한 모습을 보인다. 실제로 많은 사용자층이 있는 GUI 개발 도구에서 제공하는 GUI 컴포넌트에는 공통적인 것이 많다. 따라서 GUI 편집기에서 작성한 GUI 구성 내용을 중간표현으로 하여 GUI 코드 생성기의 입력으로 사용한다면, GUI 플랫폼에 크게 종속적이지 않은 GUI 편집기를 재사용할 수 있으므로 여러 플랫폼을 위한 GUI 개발환경을 신속히 개발할 수 있다.

3.2 GUI 개발환경 상위 설계

기존의 WYSIWYG GUI 편집기를 재사용하여 GUI 개발환경을 구현하기 위해서는 그림 1과 같은 구조로 GUI 개발환경이 구성되어야 한다. 전체 GUI 개발환경을 GUI 컴포넌트의 구성과 레이아웃을 담당하는 전단부와 GUI 구성의 기술인 중간표현, 각 GUI 플랫폼에서 동작하는 코드를 생성하는 후단부로 나눌 수 있다.



(그림 1) GUI 생성기 구조

여기서 중간표현으로 UIML[2]과 같은 표준화된 표현을 사용한다면 여러 GUI 편집기나 GUI 플랫폼으로 이식할 수 있는 장점이 있다. 하지만 이러한 표준 형태는 일반화에 의해 기술이 복잡해 질 수 있으며 각 GUI 편집기의 출력을 표준 형태로 변환해야 하는 단점이 있다.

본 연구의 목적은 여러 GUI 편집기를 사용할 수 있는 GUI 개발환경의 프레임워크를 제안하는 것이 아니라 다양한 플랫폼을 위한 GUI 개발환경을 신속하게 개발할 수 있게 하는 것이다. 이러한 목적에서 볼 때, 표준화된 표현을 사용하는 것보다는 사용 가능한 GUI 편집기에서 사용하는 GUI 구성의 저장 형태를 중간표현으로 사용하는 것이 더욱 적합하다.

이러한 개발을 하기 위해서는 다음의 요소를 제공하는 GUI 편집기를 재사용하는 GUI 편집기로 선정해야 한다.

- GUI 구성에 대한 충분한 표현력

기존의 GUI 편집기를 이용하여 다른 플랫폼을 위한 GUI 컴포넌트를 구성해야 하므로 원하는 플랫폼에서 제공하는 기본 컴포넌트들을 표현할 수 있어야 하며 지원하는 컴포넌트의 형태의 차이가 크지 않아야 한다.

- 플랫폼에 종속적이지 않은 GUI 구성

GUI 구성에 특정 GUI 플랫폼에 종속적이지 않아야 한다. 예를 들어 GUI 컴포넌트 배치에 Java의 GridLayout 속성을 이용한다면 Java의 GridLayout 형태의 레이아웃 요소를 지원하지 않는 플랫폼에서는 사용할 수 없다.

· 분석하기 용이한 저장 형태

이 시스템의 최종 목적은 원하는 플랫폼을 위한 코드를 생성하는 것이다. 그러므로 코드 생성을 하기 위한 GUI 속성을 분석하기 용이한 형태로 저장해야 한다. 본 논문에서는 GUI 편집기의 저장형태를 중간표현으로 사용하므로 중간표현 분석이 용이한 형태로 GUI 구성을 저장해야 한다.

앞서 기술한 조건을 만족하는 GUI 편집기를 선정하여 전단부를 얻었다면 선정된 중간표현을 분석하여 GUI 컴포넌트의 구성과 속성 정보를 추출하는 중간표현 분석기를 구현한다. 그리고 중간표현 분석기에서 얻은 GUI 구성 정보를 바탕으로 원하는 플랫폼을 위한 GUI 코드를 생성하는 코드 생성기를 구현하여 후단부를 구성하면 원하는 플랫폼을 위한 GUI 개발환경을 얻을 수 있다.

4. Groovy를 위한 GUI 개발환경 프로토타입 구현

본 논문에서 Groovy[3]를 위한 GUI 개발환경 프로토타입을 구현하였다. Groovy는 Java 기반 스크립트 언어로서 많은 주목을 받고 있지만 현재 Groovy를 위한 GUI 개발환경은 개발되지 않은 상태이다.

4.1 Groovy GUI 개발환경

Groovy는 James Strachan이 2003년에 발표한 언어로 Java 가상기계에서 동작하며 Java와 유사한 구문구조를 가지는 Java 기반의 스크립트 언어이다. 클로저(closure), 마크업(markup)등의 고급 개념을 지원하며 동적 타입 시스템을 가지는 특징이 있다. 이 중 마크업 기능은 HTML 코드를 작성하듯이 GUI를 구성하는 코드를 작성할 수 있게 해주는 특징이 있다.

Groovy를 위한 GUI 개발환경은 WYSIWYG 형태로 GroovySWT의 컴포넌트를 구성하고 GUI를 구성하는 GroovyMarkup 형태의 코드를 생성해야 한다. 본 논문의 프로토타입에서 구현할 요소는 GroovySWT의 다이얼로그를 지원하는 frame 컴포넌트와 텍스트 입력과 출력을 지원하는 text 컴포넌트, 버튼 입력을 제공하는 button 컴포넌트 등이다. 이 외의 컴포넌트들은 프로토타입에서 구현할 컴포넌트 및 속성 등의 사용법이 비슷하므로 같은 방법으로 간단한 작업을 통해 GroovySWT의 전체 요소를 지원하는 개발환경을 얻을 수 있으리라 생각한다.

4.2 재목적할 GUI 편집기 선정

Groovy GUI 개발환경의 전단부를 구성하는 GUI 편집기를 선정하였다. 사용할 GUI 편집기의 대상으로는 SourceForge[4]에서 오픈 소스 형태로 개발되는 것 중 완성도와 사용자 수가 많은 순으로 5개의 GUI 편집기를 후보로 선정하여 표 1의 기준에 가장 적합한 GUI 편집기를 Groovy GUI 개발환경의 전단부로 선정하였다.

<표 1> GUI 편집기 비교 기준

	기준 내용
가	충분한 GUI 컴포넌트를 제공하는가?
나	플랫폼에 독립적인 GUI 구성이 가능한가?
다	저장형태가 분석하기에 용이한가?
라	GUI 컴포넌트의 배치와 속성 편집이 용이한가?
마	쉬운 인터페이스를 제공하는가?

대표적인 오픈 소스 개발 사이트인 SourceForge에서 완성도와 사용자 수가 많은 순으로 SpecTix[5], wxGlade[6], RADi[7], jvider[8], FarPy[9]를 후보로 선정하였다. 이들 GUI 편집기에 대하여 표 1의 기준데 따라 비교하여 상·중·하로 분류하였다. 비교한 결과는 표 2와 같다.

<표 2> GUI 편집기 비교 결과

비교 기준	가	나	다	라	마
SpecTix	상	하	하	하	중
wxGlade	상	상	하	하	중
RADi	상	중	하	하	하
jvider	상	중	상	상	중
FarPy	상	상	상	상	상

표 1의 비교기준으로 GUI 편집기를 비교한 결과 FarPy를 Groovy GUI 개발환경을 위한 GUI 편집기로 선정하였다. FarPy는 Microsoft의 Visual Basic의 GUI 편집기와 유사한 인터페이스와 기능을 제공하여 사용자 편의성 면에서 가장 우수하였다. 또 XML 형태로 GUI의 구성을 기술하여 중간표현으로 사용하기에 용이한 장점이 있었다. 또한 비교 대상 중 가장 특정 플랫폼에 종속적인 요소가 적은 것으로 나타났다.

4.3 Groovy GUI 개발환경의 중간표현

Groovy GUI 개발환경의 전단부로 선정한 FarPy는 GUI 구성 형태와 속성을 XML 형태로 저장한다. 그림 2는 FarPy에서 GUI 컴포넌트의 구성과 속성을 저장하는 예이다.

그림 2의 예와 같이, GUI를 구성하는 컴포넌트를 control 태그를 사용하여 표현한다. 위치와 크기는 Location 태그와 Width, Height 태그로 표현한다. Name 태그로 각 컴포넌트의 식별자를 표현하므로 각 컴포넌트로 접근하기 위한 코드를 생성하기에 용이하다.

4.4 XML2GroovyMarkup 프로토타입

Groovy GUI 개발환경의 전단부를 결정했다면 전

```
<control type="textbox">
  <Text type="System.String">텍스트상자</Text>
  <BackColor type="System.Drawing.Color">Color [Window]</BackColor>
  <Font type="System.Drawing.Font">[Font: Name=Microsoft Sans Serif,
          Size=8.25, Units=3, GdiCharSet=1, GdiVerticalFont=False]</Font>
  <Multiline type="System.Boolean">False</Multiline>
  <Location type="System.Drawing.Point">{X=31,Y=15}</Location>
  <Grouped type="System.Boolean">False</Grouped>
  <GroupName type="System.String" />
  <Width type="System.Int32">309</Width>
  <Height type="System.Int32">20</Height>
  <Cursor type="System.Windows.Forms.Cursor">[Cursor: Default]</Cursor>
  <Name type="System.String">textbox1</Name>
</control>
```

(그림 2) GUI 중간표현의 예

단부인 FarPy가 생성하는 중간표현을 분석하여 Groovy의 GUI를 구성하는 GroovyMarkup 코드를 생성하는 후단부를 구성해야 한다. 각 다이얼로그를 이루는 form 태그 이하의 control 태그의 type 속성으로 GUI 컴포넌트를 결정하고 이하 서브 태그의 속성으로 GUI 컴포넌트의 속성을 결정하면 된다.

본 프로토타입을 Groovy 언어를 이용하여 구현하였다. XML2GroovyMarkup은 크게 XML을 파싱하여 속성과 내용을 가져오는 부분과 얻어온 컴포넌트의 정보를 바탕으로 GroovyMarkup 코드를 생성하는 부분으로 나눌 수 있다.

XML 형태의 중간표현을 분석하는 부분은 Groovy에서 XPath[10]와 같은 기능을 하는 GPath 라이브러리를 이용해서 쉽게 해결할 수 있었다. Groovy에서는 GPath를 통해 XML 노드의 속성과 하위 노드에 대한 쉬운 접근을 문법적인 차원에서 지원하기 때문에 XML 형태의 중간표현에서 각 GUI 컴포넌트에 대한 정보를 쉽게 얻을 수 있다.

컴포넌트의 정보를 바탕으로 GroovyMarkup 코드를 생성하는 부분은 Groovy에서 클로저 형태로 제공하는 라이브러리를 통해 간단히 구현할 수 있었다. 특히 Groovy에서는 스트링 처리를 위한 커리드 클로저(curried closure)를 지원하기 때문에 GUI를 구성하는 전체 코드의 틀에서 속성과 내용이 바뀌는 부분만 원하는 내용을 치환할 수 있다. 이를 이용하면 특정한 틀을 가지는 GUI 구성 코드를 간단히 생성할 수 있다.

그림 3은 그림 2의 중간표현으로 생성되는 GroovyMarkup 코드의 예이다. 그림 3의 예와 같이 간단히 코드를 생성할 수 있다.

```
def  textbox1 = text(text:"텍스트상자",
                     location:[31, 15], size:[309, 20], SWT.LEFT) {
     selectionListener( widgetSelected:{ event -> /* To do : ... */  } )
}
```

(그림 3) GroovyMarkup 코드의 예

5. 결론

본 논문에서는 기존의 GUI 편집기를 재목적하여 GUI 개발환경을 얻는 방법에 대해 기술하였으며 Groovy GUI 개발환경의 개발에 적용하였다. FarPy GUI 편집기를 전단부로 하여 GUI를 구성하는 GroovyMarkup 코드를 생성하는 GUI 생성기를 구현하였다. 기존 GUI 편집기를 재사용하기 때문에 WYSIWYG 방식의 GUI 개발환경의 프로토타입을 단시간에 개발할 수 있었다.

실제 WYSIWYG 방식의 GUI 개발환경은 응용프로그램 개발자의 편의와 빠르고 쉬운 GUI 구성을 위해서는 필수적이지만, 각 플랫폼이나 사용 언어마다 WYSIWYG 방식의 GUI 편집기를 구현하기가 까다로워 대형 소프트웨어 공급자 외에는 GUI 개발환경을 제공하기 힘들었다. 이러한 점에서 본 논문에서 제안한 방법은 기존에 개발된 GUI 개발환경에서 사용하는 GUI 편집기를 재사용하여 다른 언어나 플랫폼을 위한 GUI 개발환경을 비교적 빠르고 쉽게 얻을 수 있다는 점에서 바람직하다고 할 수 있다.

현재 이 방법을 적용하여 개발된 Groovy GUI 개발환경은 GUI 편집기의 출력이 일방적으로 XML2GroovyMarkup의 입력으로 전달되는 구조로 되어있다. XML2GroovyMarkup에서 생성하는 GUI 코드에 사용자가 수동으로 GUI를 구성하기 위한 코드를 삽입할 수 있기 때문에 사용자가 GUI 코드를 수정하는 경우, GUI 편집기에서 보는 GUI 구성 내용과 실제 구현되는 GUI가 다를 수 있다. 현재 많은 GUI 개발환경에서 이러한 문제를 해결하고 있지 않지만, 역으로 사용자의 GUI 프로그램 코드를 읽어서 XML 형태의 중간표현을 생성한다면 앞서 언급한 문제를 해결할 수 있을 것으로 예상한다.

이후 여러 GUI 플랫폼에도 적합한 GUI 중간표현이나 GUI 개발환경의 프레임워크가 연구된다면 사용자 편의를 위한 개발환경에 큰 도움이 될 것이다.

참고문헌

[1] Wikipedia, GUI : http://en.wikipedia.org/wiki/Graphical_user_interface

[2] Marc Abrams, James Helms, "User Interface Markup Language Specification" Working Draft 3.1, 2004

[3] Groovy Homepage, http://groovy.codehaus.org

[4] SourceForge Homepage, http://www.sourceforge.net

[5] SpecTix Homepage, http://www.python.net/crew/mike/src/Spectix/Spectix.html

[6] wxGlade Homepage, http://wxglade.sourceforge.net/

[7] RADi Homepage, http://www.muntjak.com

[8] jvider Homepage, http://www.jvider.com

[9] FarPy Homepage, http://farpy.holev.com

[10] XPath Homepage, http://www.w3.org/TR/xpath