

# 프로그램 변환을 통한 J2ME 프로그램의 동적 이벤트 분석\*

최윤정<sup>o</sup> 창병모  
숙명여자대학교 컴퓨터학과  
{swany<sup>o</sup>, chang}@sookmyung.ac.kr

## Dynamic Event Analysis of J2ME Programs by Program Transformation

Yoon-jeong Choi<sup>o</sup> Byeong-Mo Chang  
Dept. of Computer Science, Sookmyung Women's Univ.

### 요 약

J2ME 모바일 프로그램은 모바일 기기가 보편화되면서 그 사용이 증가하고 있다. J2ME 프로그램은 자원이 제한된 모바일 환경에서 실행되므로 자원의 효율적인 사용이 매우 중요하다. J2ME 프로그램은 대부분 이벤트-구동 프로그램이며 효과적인 이벤트 관리가 자원 최적화에 많은 영향을 미치며 이벤트 관련 디버깅은 매우 어려운 부분이다. 본 연구에서는 실행 중에 실시간으로 이벤트 발생 및 처리 정보를 보여줄 수 있는 시스템을 설계 개발하였다. 이 시스템은 사용자 옵션에 따라 사용자가 관심 있는 이벤트만을 실행 중에 추적하거나 실행 후에 이벤트 관련 요약 프로파일 정보를 제공한다.

### 1. 서 론

최근 들어 핸드폰과 같은 모바일 기기 사용이 보편화 되면서 이를 위한 Java 언어 J2ME(Java 2 Micro Edition)가 널리 사용되고 있다[3]. J2ME 프로그램은 모바일 기기의 특성상 제한된 자원을 사용할 수밖에 없으며 이러한 제한된 자원을 효율적으로 사용하는 것이 매우 중요하다. J2ME 프로그램에서 다루는 자원은 메모리, CPU 등의 통산적인 자원뿐만 아니라 입력 키 등에서 발생하는 이벤트 등을 들 수 있다[4].

실제로 대부분의 J2ME 프로그램은 이벤트-구동 프로그램으로 이러한 이벤트의 효과적인 처리가 전체 프로그램의 안전성과 신뢰성뿐만 아니라 효율성에 큰 영향을 미칠 수 있다. 또한 이러한 이벤트-구동 프로그램은 보통 디버깅이 매우 어려운 단점을 가지고 있다. 현재 많이 사용되고 있는 Wireless Toolkit은 이벤트 처리에 대한 정보를 전혀 제공하지 않고 있다[5]. 따라서 프로그램 개발자는 이벤트 관련해서 효과적인 디버깅이 어려우며 이는 프로그램 신뢰성에 저해 요인이 될 수 있다.

본 연구에서는 이러한 문제점을 해결하기 위해 실행 중에 실시간으로 이벤트 발생 및 처리 정보를 보여줄 수 있는 시스템을 설계 개발한다. 이 시스템은 사용자 옵션에 따라 사용자가 관심 있는 이벤트만을 실행 중에 추적해 볼 수 있으며 또한 실행 후에 이벤트 관련 요약 프로파일 정보를 제공한다. 이를 이용하여 사용자는 이벤트 처리 과정을 관심 있는 부분을 중심으로 살펴볼 수 있으며 이러한 기능은 J2ME 같은 모바일 프로그램의 신뢰성

및 효율성 향상에 기여할 수 있을 것이다.

본 연구에서는 동적 분석에 의한 실행 시간 부담을 줄이기 위해 프로그램 변환을 기반으로 설계하였다. 사용자 옵션에 따라 관심 있는 정보만을 출력하도록 코드를 삽입하여 입력 프로그램을 변환한다. 이 변환된 프로그램은 원래 프로그램과 동일하게 동작하면서 실행 중에 트레이스 정보를 출력하고 실행 직후에는 이벤트 프로파일을 출력한다.

본 논문의 구성은 다음과 같다. 2장에서는 먼저 시스템의 설계에 대해 설명한다. 3장에서는 구현 및 실행 결과를 제시하고 4장에서는 결론 및 향후 연구 과제에 대한 소개를 한다.

### 2. 설 계

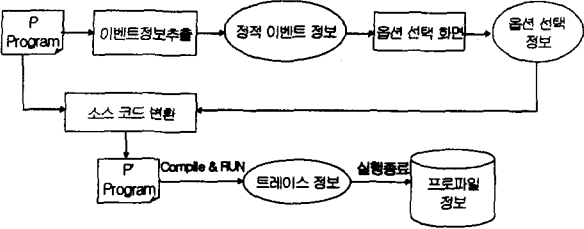
본 연구에서 개발할 시스템은 사용자가 동적 이벤트 분석을 위한 옵션을 선택할 수 있으며 이를 바탕으로 사용자는 관심 있는 이벤트만의 발생 및 처리 과정을 실행 중 실시간으로 추적할 수 있도록 설계하였다. 또한 실행 후에 실행 중 발생 혹은 처리된 이벤트들에 대한 요약 정보인 이벤트 프로파일을 제공하여 사용자가 이벤트 처리 과정을 요약해서 살펴볼 수 있도록 하였다.

본 연구에서는 동적 분석에 의한 실행 시간 부담을 줄이기 위해 프로그램 변환 방법을 기반으로 설계하였다. 사용자 옵션에 따라 해당 트레이스 정보만을 출력하도록 입력 프로그램 P에 코드를 삽입하여 변환된 새로운 프로그램 P'를 생성한다. 이 변환된 프로그램은 원래 프로그램과 동일하게 동작하면서 실행 중에 트레이스 정보를 출력하고 실행 후에 이벤트 요약 프로파일을 출력한다.

본 시스템은 크게 세 부분으로 구성되어 있다. 첫 번째 부분은 입력 프로그램을 간단하게 정적 분석하여 이벤트 관련 프로그램 구조를 보여주며 이 정보를 이용하여 사용자는 관심 있는 부분 혹은 정보만을 선택할 수

\* 본 연구는 한국과학재단 특장기초과제 “정적 분석을 이용한 모바일 자바 프로그램의 효율적인 자원 사용을 위한 환경 연구”(R01-2002-000-003630-0)의 지원에 의해 수행되었음.

있다. 두 번째 부분은 프로그램 실행 중의 이벤트 발생과 처리에 대한 트race 정보로 이 정보는 옵션 선택에 따라 변환된 프로그램을 실행하면 얻을 수 있는 정보이다. 세 번째는 프로그램 실행 후에 이벤트 실행에 대한 요약 정보인 이벤트 프로파일로 이 역시 변환된 프로그램을 실행하면 실행 직후에 제공되도록 설계하였다. 시스템의 전체적인 구조는 [그림 1]과 같다.



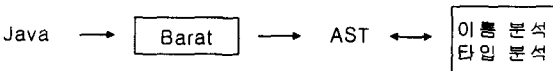
[그림 1] 시스템 구조

첫 단계인 디스플레이는 일종의 시작화면으로 대상 프로그램을 열고 옵션을 선택할 수 있다. 옵션 선택을 위해 입력 프로그램 P를 간단하게 정적 분석하여 프로그램 내의 발생 가능한 이벤트 종류와 관련 컴포넌트 등의 정적 이벤트 정보를 추출한다. 사용자는 이 정보를 이용하여 관심 있는 이벤트나 컴포넌트를 선택할 수 있다. 두 번째 단계에서는 사용자가 선택한 옵션에 따라 기존의 프로그램과 똑같이 동작하면서 동적 이벤트 정보를 출력할 수 있도록 코드를 삽입하여 기존의 프로그램 P를 새로운 프로그램 P'로 변환한다. 이 부분은 [그림 1]의 소스 코드 변환에서 이루어진다. 마지막으로 이렇게 변환된 프로그램 P'를 실행시키면 삽입된 코드에 의해 실행 중에 트race 정보를 출력하고 실행 후에 이벤트 요약 프로파일 정보가 출력된다.

3. 구현 및 실험 결과

3.1 구현 환경

본 연구에서는 이벤트 정보 추출과 소스 코드 변환을 위해 자바 컴파일러 전단부인 Barat을 이용하였다. Barat은 Java 프로그램의 정적 분석기 구현을 지원하는 컴파일러 전단부로 자바 소스 파일이나 클래스 파일을 입력 받아 이름과 타입 정보를 포함하는 AST (Abstract Syntax Tree)를 구성한다[1].



[그림 2] Barat의 구조

이렇게 구성된 AST의 각 노드를 비지터(Visitor)라는 디자인 패턴을 이용한 트리 횡단 루틴을 이용하여 방문하면서 필요한 연산을 수행할 수 있다. Barat이 기본적으로 제공하는 비지터에는 AST의 모든 노드들을 깊이

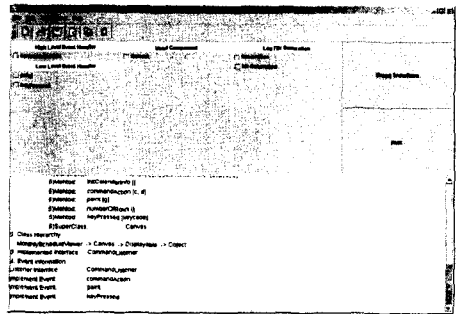
우선탐색을 해주는 DescendingVisitor, 모든 노드를 검색한 후 AST정보를 기반으로 소스 코드를 다시 생성해주는 OutputVisitor 등이 있다. 따라서 이러한 비지터를 확장하여 AST노드 방문 시 원하는 연산을 수행하는 비지터를 구현할 수 있다[2].

본 논문에서는 Barat에서 제공하는 DescendingVisitor와 OutputVisitor의 확장하여 이벤트 정보 추출과 소스 코드 변환을 구현하였다.

3.2 구현

[그림 1]의 이벤트 정보 추출은 DescendingVisitor를 확장하여 구현하였으며 이 분석기는 사용자의 옵션 선택을 위해 입력 프로그램을 간단하게 정적 분석하여 프로그램 내의 발생 가능한 이벤트 종류와 생성된 컴포넌트 등의 정적 이벤트 정보를 추출한다. 특히 구현된 이벤트 리스너와 이벤트 처리 메소드들에 대한 정보를 제공한다.

이렇게 정적 분석된 이벤트 정보를 가지고 만들어진 옵션 정보를 이용하여 [그림 3]와 같은 화면을 구성한다. 사용자는 관심 있는 이벤트나 컴포넌트만을 선택하여 실행 시 원하는 정보를 제공 받을 수 있다.



[그림 3] 옵션 선택 화면

[그림 1]의 소스 코드 변환은 OutputVisitor를 확장하여 구현하였다. 이 부분은 입력 프로그램 P를 옵션 선택에 따라 해당 이벤트들의 처리 과정을 추적할 수 있도록 코드를 추가하여 프로그램 P'를 생성한다.

예를 들어 간단한 J2ME 프로그램인 HelloWorld.java 프로그램의 변환 과정을 살펴보자. 이 프로그램은 [그림 4]와 같이 이벤트 핸들러 commandAction이 구현되어 있다. 소스 프로그램을 변환할 때 옵션 정보에 따라서 관심 있는 정보만을 출력하도록 변환한다. 옵션으로 commandAction 핸들러와 TextBox 컴포넌트를 선택했을 경우의 변환된 프로그램 [그림 5]를 살펴보자.

```
public void commandAction(Command c, Displayable s) {
    if (c == exit) {
        destroyApp(false);
        notifyDestroyed();
    }
}
```

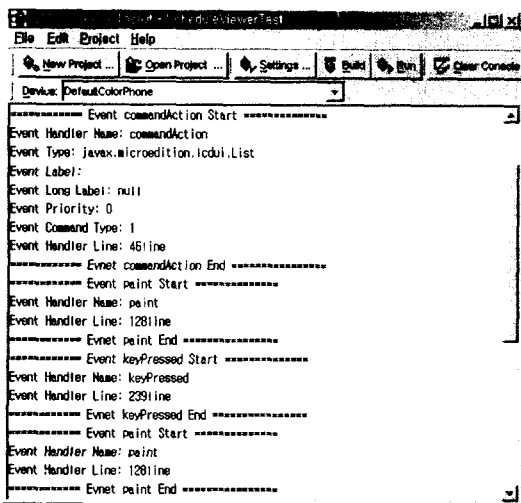
[그림 4] 변환 전 소스프로그램

```

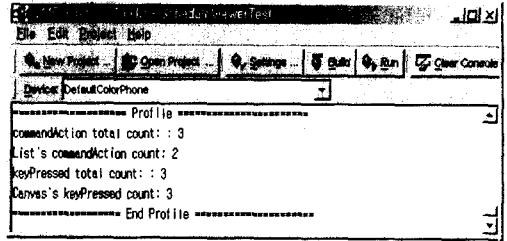
public void commandAction
    (Command c, Displayable s){if(c==exit) {
String command2 ="javax.microedition.lcdui.TextBox";
if(command2.equals(s.getClass().getName())) {
System.out.println
(*===== Event commandAction Start =====*);
System.out.println("Event Handler Name: commandAction");
System.out.println("Event Type: " + s.getClass().getName());
System.out.println("Event Label: " + c.getLabel());
System.out.println("Event Long Label: " + c.getLongLabel());
System.out.println("Event Priority: " + c.getPriority());
System.out.println("Command type:" + c.getCommandType());
System.out.println("Event Handler Line: 41line");
System.out.println
(*===== Evnet commandAction End =====*);
}
if (c == this.exit) {
this.destroyApp(false);
this.notifyDestroyed();
}
}
}
    
```

[그림 5] 변환 후 소스 프로그램

변환된 프로그램 P'은 컴파일 후에 실행되어 사용자에게 이벤트 처리 과정을 추적할 수 있는 정보를 제공한다. 변환된 프로그램을 실행하면 실행 중에 실시간으로 발생된 이벤트 정보와 처리과정 등의 트레이스 정보들 [그림 6]과 같이 제공한다. 또한 프로그램이 종료되었을 때 [그림 7]과 같이 실행 중에 발생된 이벤트들의 요약 프로파일 정보를 제공한다.



[그림 6] 이벤트 트레이스 정보



[그림 7] 프로파일 정보

#### 4. 결론 및 향후 과제

J2ME 프로그램의 이벤트 처리에 대한 분석은 자원의 효과적인 사용과 더불어 이벤트 디버깅에 있어서 꼭 필요한 부분이다. 본 연구에서는 이를 위하여 프로그램 변환을 통하여 실행 시간에 이벤트 발생 및 처리과정을 추적할 수 있는 시스템을 설계 개발하였다. 특히 사용자 옵션에 따라 관심 있는 이벤트를 선택적으로 추적할 수 있도록 하였다. 이를 이용하여 사용자는 이벤트 처리 과정 중 관심 있는 부분을 중심으로 추적할 수 있으며 이러한 기능은 모바일 프로그램의 신뢰성 및 효율성 향상에 기여할 수 있을 것으로 기대된다.

현재까지는 프로그램의 변환 관련 부분만을 구현하였으며 변환된 프로그램을 수동으로 실행해야 한다. 앞으로 변환된 프로그램을 사용자에게 보이지 않으면서 자동으로 실행해야 하고 실행 결과인 트레이스 정보나 프로파일 정보를 별도로 시각화해서 보여주는 등의 연구를 계속 수행할 계획이다.

#### 5. 참고 문헌

[1] Boris Bokowski, André Spiegel. Barat - A Front-End for Java. Technical Report B-98-09 December 1998

[2] <http://www.sharemation.com/bokowski/barat/index.html>

[3] James White. An Introduction to Java 2 Micro Edition (J2ME); Java in Small Things. 23rd International Conference on Software Engineering (ICSE'01) May 12-19, 2001, Toronto, Canada

[4] <http://java.sun.com/j2me/>

[5] <http://java.sun.com/products/j2mewtoolkit/>

[6] Bruno Dufour, Karel Driesen, Laurie Hendren and Clark Verbrugge. Dynamic Metrics for Java. ACM OOPSLA '03, October, 2003, Anaheim, CA.