

분산처리 진단/교정 시스템을 위한 자바 디버깅 정보 추출에 관한 연구

류 동 항*, 정 민 수**, 윤 기 송***

(주) 에듀뱅크*

경남대학교 컴퓨터공학과**

한국전자통신연구원 컴퓨터&소프트웨어기술연구소 인터넷서비스연구부***

A Study of Java Debugging Information Extract for Distributed Diagnosis and Correction System

Donghang Ryu*, Minsoo Jung**, Kisong Yoon***

*Eduback Corp.

**Dept. of Computer Engineering, Kyungnam University

***Internet Service Dep., Computer & software Technology Lab., ETRI

요 약

RPC 에 기반한 분산처리 진단/교정 시스템에서는 C/C++ 언어를 위한 디버깅을 지원한다. 본 논문에서는 이 시스템에 자바 언어에 대한 디버깅 지원을 위한 디버깅 정보 추출부를 설계 하였으며 각각의 정보 추출 모듈을 작성하였다. 자바 언어에서는 C/C++과 달리 디버깅 정보가 런 타임시에 결정되는 경우가 많기 때문에 이에 대한 적절한 정보 제공이 요구된다. 본 논문에서 제공하는 자바 디버깅 정보를 이용하여 분산환경의 네트워크 상에서 사용자에게 다양한 자바 디버깅 정보를 실시간으로 제공할 수 있다.

1. 서 론

분산처리 진단/교정 시스템은 이기종 분산환경에서 작동하는 어플리케이션을 진단/교정하는 도구이다. 이 시스템은 선 워크스테이션(SunOS 4.1.x, Solaris 2.x) 및 HP, LINUX 그리고 마이크로소프트사의 운영체제(NT 4.0, NT 3.5, 윈도우즈 95/98)를 사용하는 시스템을 지원한다.

네트워크 상의 이벤트 추적 및 통신 지원 기능은 RPC(Remote Procedure Call)을 이용하며, 지원언어는 C/C++이다. 본 논문에서는 이 시스템에 자바 언어에 대한 디버깅을 지원하기 위한 정보 추출부를 설명한다.

자바 언어는 다른 언어와는 달리 클래스 파일이라는 특수한 구조로 되어 있으며, 이 파일 내에 필요한 모든 디버깅 정보가 저장되어 있다.

본 논문에서는 이 클래스 파일을 분석하여 스테이트먼트 정보나 지역 변수의 이름과 같은 정적 디버깅 정보 및 배열의 크기와 같은 가변적인 정보의 디스플레이를 위한 동적 디버깅 정보를 추출한다. 또한 자바 언어로 작성된 객체 지향 프로그램의 분석 및 이해를 돕기 위해 클래스 브라우저를 제공한다.

본 논문의 구성은 다음과 같다. 2 장에서는 자바 디버깅 정보 추출부를 기술하고, 3 장에서는 클래스 브라우저에 대해 소개하며, 4 장에서 구현 결과를 5 장에서 결론을 제시한다.

2. 자바 디버깅 정보 추출부

자바 디버깅 정보 추출부는 스테이트먼트 및 지역 변

수등과 같은 정적 디버깅 정보와 가변적인 배열의 크기 정보와 같은 동적 디버깅 정보 추출부로 구성된다. 각각의 정보는 서버 및 클라이언트로 전송되어 디버깅을 위한 기초자료가 된다.

2.1 정적 디버깅 정보 추출부의 개요

자바 정적 디버깅 정보 추출부는 *Main* 이 포함된 주 클래스로부터 시작하여 프로그램 수행 시에 사용하게 될 모든 클래스 정보를 찾고 각각의 클래스에 해당되는 모든 메소드 및 필드 정보를 얻는다. 그리고 각각의 클래스에 해당하는 모든 메소드 정보도 매달게 된다.

나아가 메소드의 소스 코드 및 브레이크 포인트와 같은 내부 정보도 포함한다.

2.2 정적 디버깅 정보 추출부의 설계 및 기능

● *Module Info*

SVOID	sourceFileOID_;
RWWSTRING	language_;
SVOID	superClassOID_
SVOID	linkUnitOID_;
SVScopeList	scopeList_;
SVSrcFileList	srcFileList;
SVMODULESTATE	moduleState_

<그림 2-1> *Module Info* 부의 필드 구성

Module Info 부는 클래스의 정보를 저장하기 위한 곳으로서 현 클래스 및 클래스의 소스 파일 그리고 클래스 내에 포함된 메소드 정보를 포함하고 있다.

Module Info 부의 각 필드 정보를 얻기 위해 *GetModule()* 함수를 이용한다. 이 함수에서는 *Main* 이 포함된 클래스 파일의 코드부를 검색하여 이 클래스와 관련된 모든 클래스와 메소드의 리스트를 구한다.

● *Scope Info*

RWCString	baseAddr_;
RWCString	endAddr_;
RWBoolean	isLeaf_
SVScopeStatus	scopeStatus_;
SVMODULE	*module_;
SVSrcFile	*srcFile_;
SVSrcRange	*srcRange_;
SVStmTable	*stmtTable_;

<그림 2-2> *Scope Info* 부의 필드 구성

Scope Info 부는 실제 디버깅을 위한 메소드의 *Start* 어드레스, *End* 어드레스등과 소스 파일, 그리고 메소드 바이트코드부의 PC 및 브레이크 포인트 정보로 구성된다.

각 필드의 정보를 얻기 위해 *getScope()* 함수의 호출을 통하여 메소드의 코드 어트리뷰트 및 라인 넘버 테이블을 검색하여 프로그램 카운터 및 라인 넘버 정보를 구한다.

파라미터 및 지역 변수의 경우에는 필요할 때 마다 *getParamLocals()*를 호출하여 각 메소드에 해당하는 파라미터 및 지역 변수의 정보를 추출한다.

자바 언어에서는 지역 변수 뿐만 아니라 필드를 제공하는데 필드는 자료형으로 기본형 및 클래스를 가질 수 있다. 따라서 런 타임시에 필드 객체의 형 정보를 추출 할 수 있어야 한다. 이를 위해 필드의 형 정보를 이용하여 배열 일 경우에는 배열의 크기를 구하게 되고, 클래스 일 경우에는 역시 객체의 필드의 추출한 후 트리 형태로서 객체를 표현할 수 있게 하였다.

2.3 동적 디버깅 정보 추출부의 개요

동적 디버깅 정보는 런 타임시에 지속적으로 변화되는 변수등과 같이 하나의 명령 단위를 수행할 때 마다 정보의 갱신이 요구되는 부분이다. 대표적인 동적 디버깅 정보로는 로컬 변수의 값을 변경하는 것과 배열이 크기가 변경되는 것을 검사하는 것이다.

C/C++과는 달리 자바 언어는 프로그래머가 배열의 크기를 자유자재로 변경할 수 있기 때문에 하나의 명령을 실행하고 난 후에는 반드시 변수의 크기를 검사해야만 한다. 물론 디스플레이되는 지역 변수의 값도 계속적으로 갱신해야 함은 당연하다.

2.4 동적 디버깅 정보 추출부의 설계 및 기능

● *Array Info*

배열의 크기를 구하기 위해서는 클래스의 이름, 현재의 PC, 배열의 차원, 배열의 *Slot* 넘버가 필요하다. 여기서 배열의 차원은 처음에 수행시에 결정되는 사항이고, 배열의 차원에 따른 원소의 개수가 변화가 되는 부분이다. 그리고 *Slot* 넘버는 배열의 이름을 구분 짓는 요소이다.

이러한 배열의 크기 변경을 찾아내기 위해서는 자바 바이트 코드부를 검색해야만 한다.

● *Parameter Info*

메소드의 파라미터 정보는 기본형은 물론 객체도 가능하다. 따라서 기본형이 아닌 객체의 경우에는 런 타임시에 그 정보가 결정 되기 때문에 실행 시에 그 정보를 유지해야만 한다.

3. 클래스 정보 추출부

이 장에서는 직접적인 디버깅 정보와는 관계가 없지만 프로그래머의 개발 편의 및 효율을 높이기 위해 클래스 파일 브라우저를 제공한다. 본 클래스 파일 브라우저는 자바 스펙에 규정된 클래스 파일 형태로 정보를 디스플레이한다.

3.1 클래스 파일 브라우저의 개요

클래스 파일은 자바 가상머신이 실행하는 이진 파일이다. 따라서 실제로 클래스파일이 어떻게 구성되고, 클래스내에 있는 각각의 변수와 메소드들의 실제 코드가 어떻게 유기적으로 구성되는지 알게 함으로써 자바 언어를 더욱 깊게 이해할 수 있게 한다.

클래스 파일 브라우저는 매직넘버로부터 어트리뷰트 카운트까지의 일반 클래스 정보와 인터페이스, 필드, 메소드에 대한 상세한 정보를 제공한다.

3.2 클래스 파일 브라우저의 설계 및 기능

클래스 파일 브라우저는 사용자가 편리하게 클래스의 다양한 정보를 사용할 수 있게 하기 위하여 자바 스펙에 규정된 클래스의 자료구조 형식대로 화면에 출력해 준다. 출력되는 주요한 정보로는 *Constant pool* 과 자신의 슈퍼 클래스와 사용된 메소드 등이 있다.

자바 클래스 파일을 순차적으로 이진파일로 읽어들이 스펙에 규정된 형태로 만들게 되며, 이 객체를 적절히 분석하여 사용자가 읽을 수 있도록 변환한다.

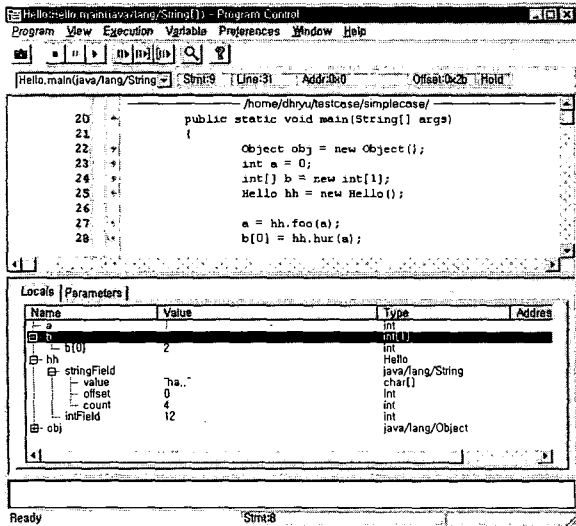
4. 구현 결과

4.1 구현 환경

- 서버 : Sun Solaris 2.6, GCC
- 클라이언트 : Visual C++ 6.0

4.2 자바 디버깅 추출부의 구현 결과

자바 디버깅 추출부의 구현 결과는 그림 4-1에 나타났다. 구현 결과는 클라이언트측의 사용자 인터페이스이며 본 논문에서 작성한 코드는 서버측에서 실행되어 RPC를 이용하여 클라이언트측으로 정보를 전송하는 구조를 가진다.



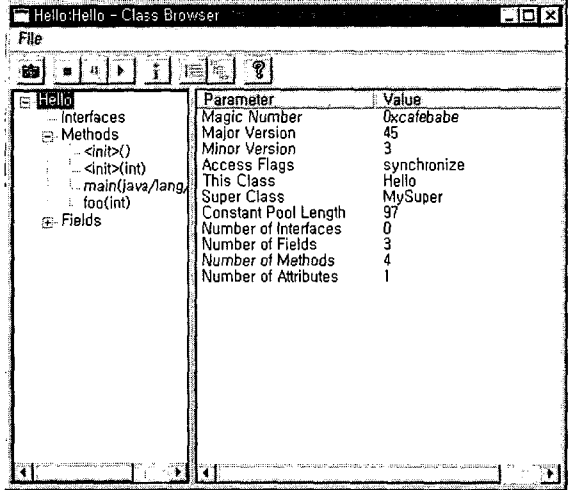
<그림 4-1> 자바 디버깅 정보 추출부의 구현 결과

4.3 클래스 브라우저의 구현 결과

클래스 브라우저는 사용자가 클래스를 보다 쉽게 이해하고 브라우징 할 수 있도록 해주는 유틸리티이다.

본 논문에서 구현한 클래스 브라우저는 그림 4-2에 나타내었다.

자바 스펙에 규정된 형식대로 사용자가 쉽게 정보를 이용할 수 있다.



<그림 4-2> 클래스 브라우저의 구현 결과

5. 결론

본 논문에서 구현한 자바 디버깅 정보 추출부와 클래스 브라우저는 C/C++만 지원되는 기존의 분산처리 진단/교정 시스템에 자바 지원기능을 추가하였다. 따라서 본 진단/교정 시스템은 분산된 환경에서 다양한 언어에 대한 디버깅이 가능하였다.

향후 과제로는 분산처리 진단/교정 시스템의 처리 속도의 개선 및 시스템 안정화가 요구된다.

참고 문헌

- [1] C. Laffra, "Advanced Java", Prentice Hall
- [2] E. Anuff, "Java Sourcebook", Wiley
- [3] G. Vanderburg, "Maximim Java 1.1", Sams
- [4] J.A.Pew, "Instant Java 2/E", Prentice Hall
- [5] J.C. Cherniavsky, P.B. Henderson, and J. Keohane, "On the equivalence of URE flow graphs and reducible flow graphs," Proc. 1976 Conference on Information Sciences and Systems, Johns Hopkins Univ., 423-429, 1976.
- [6] J. Gosling,, "Java Language Specification", Addison-Wesley
- [7] J.R. Jackson, "Java by Example 2/E", Prentice Hall
- [8] T. Lindholm and F. Yellin, "The Java Virtual Machine Specification" ADDISON-WESLEY 1997.
- [9] 조영욱 외 4인, "분산처리 디버거 유니뷰 서버의 구조", 정보과학회 97년 추계학술발표 논문집, 제 24 권 2호(I), pp.327-330, 이화여자대학교, 1997.
- [10] 성명제 외 4인, "분산 디버거 유니뷰 시스템의 개발", 정보과학회 98년 논문지, 8월호 1998.