XML기반 ADL 모델 관리기 설계 및 구현

조용한° 강미나 승현우, 전태웅*
서울여자대학교 컴퓨터학과, 고려대학교 전산학과*
{quiethan°, gmnmn, hwseung}@swu.ac.kr
jeon@selab.korea.ac.kr*

The Design and Implementation of XML Based ADL Model Management

Yonghan Cho°, Mina Kang, Hyonwoo Seung, Taewoong Jeon*
Dept. of Computer Science, Seoul Women's University
Dept. of Computer Science, Korea University*

요 약

최근 소프트웨어 재사용과 소프트웨어의 응용범위가 넓어짐에 따라 잘 정의된 아키텍처를 기반으로 개발된 컴포넌트 소프트웨어 개발(CBD : Component-Based Development)방식의 채택이 빠르게 확산되고 있다. 이러한 컴포넌트 소프트웨어 개발 방식이 빠르게 확산되면서 아키텍처를 정확하게 기술, 분석, 정제할 수 있는 능력 또한 점차 중요시 되고 있다. 아키텍처를 정확하게 기술, 분석 정제하기 위해서는 아키텍처 기술 언어(ADL)의 사용이 필요하다. 또한 ADL로 기술된 모델 정보들을 서로 다른 ADL지원도구들이 공유할 수 있도록 ADL모델 관리기의 개발이 필요하다. 본 논문에서는 ADL 모델 관리기의 구성요소를 기술하며, ADL 모델 정보를 XML형태로 변환하는 변환기의 변환 과정을 설명하고, XML 형태로 변환된 ADL 모델 정보들을 저장하는 저장소 구조를 각각 기술하고, 검색기의 검색 과정 및 검색 결과를 보여준다.

1. 서 론

최근 소프트웨어 재사용과 소프트웨어의 응용범위가 넓어짐에 따라 그 응용영역의 복잡성과 방대함이 커지고 있다. 따라서 잘 정의된 아키텍처를 기반으로 개발된 컴포넌트 소프트웨어 개발(CBD : Component-Based Development)방식의 채택이 빠르게 확산되고 있다. 이러한 컴포넌트 방식의 채택이 빠르게 확산 되면서 아키텍처를 정확하게 기술, 분석, 정제할 수 있는 능력이 점차 중요시 되고 있다. 아키텍처를 정확하게 기술, 분석 정제하기 위해서는 아키텍처 기술 언어(ADL)의 사용이 필요하며, ADL로 기술된 아키텍처 모델을 효과적으로 분석, 처리, 관리할 수 있는 지원 환경이 필수적이다.

본 연구는 ADL 모델 지원 환경 시스템 개발을 목표로 주관연구와 공동 연구로 나누어 수행되었다. 주관연구에서는 ADL로 기술된 소프트웨어 아키텍처 모델의 구조적, 행위적, 성질들을 정증적으로 분석, 검사하는 방법 및 이를 지원하는 아키텍처 모델 분석기를 개발하였다. 공동연구에서는 아키텍처 모델 분석기를 통해 생성된 아키텍처 정보들이 XML 형태로 변환, 저장, 검색될 수 있도록 ADL 모델 관리기를 개발하였다.

본 논문에서는 ADL 모델 정보들을 서로 다른 ADL 지원 도구들이 공유할 수 있도록 XML 형태로 변환하는 변환기, XML 형태로 변환된 ADL 모델 정보들을 저장하기 위한 저장소, 저장소로부터 USER의 query에 맞게 다양한 정보를 검색하는 검색과정 및 그 결과를 보여준다.

2. ADL 모델 관리기

ADL 모델 관리기는 아키텍처 모델링 분석 작업 과정 중에 생성된 아키텍처 모델링 정보들을 서로 다른 ADL 지원도구들이 원활하게 공유할 수 있도록 ADL 모델들을 XML 형태로 변환, 저장, 관리하기 위한 도구이다.[1]

ADL 모델 관리기는 크게 XML기반 ADL 모델 변환기, 저장소, 검색기로 구성되어 있다. ADL 모델 관리기 내에서 ADL파서에 의해 파싱된 ADL모델 정보들은 변환기를 통해 ADL->XML 변환 과정을 통해 XML형태의 모델로 변환되며 변환된 XML형태의 모델 정보들은 저장소에 저장된다. 또한 저장된 XML형태의 모델 정보들은 USER의 query에 의해 검색기를 통해 검색된다.

3. XML기반 ADL 모델 변환기

XML기반 ADL 모델 변환기의 구조는 다음(그림 1)과 같다. 변환기는 파서를 통해 파싱된 형태의 ADL 모델 정보를 XML 형태로 변환하는 ADL->XML변환과 변환된 XML 모델을 ADL 로 변환하는 XML->ADL변환을 수행한다.

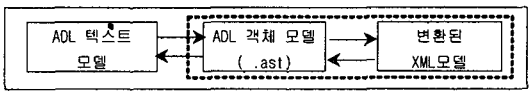

그림 1. XML 기반 ADL 모델 변환기 구조

3.1 ADL->XML 변환 과정

ADL 모델정보들을 XML 형태로 변환하기 위한 ADL->XML 변환은, 먼저 IDN 모델, ADN 모델에서 모델 정보를 얻는다. IDN모델은 하나의 컴포넌트에 대한 모델이고, ADN모델은 컴포넌트 다수가 모여 만들어진 아키텍처에 대한 모델이다.[2] 여기서 추출된 각각의 모델 정보를 DOM파서를 이용하여 DOM트리에 삽입하고 XML 모델로 생성한 뒤 XML_Schema로 정의된 ADL 메타모델을 기준으로 유효성을 검사하게 된다. ADL->XML로 변환하는 작업에서 사용되는 클래스로는 XMLGUIManager, MakeADNXMLModel, MakeIDNXMLModel, ValidChecker이며, 이 중 XMLGUIManager는 변환 작업과정에서 사용하는 클래스로 ADL모델을 XML형태로 변환하는 전체적인 과정에 관여한다. XMLGUIManager 클래스는 XML변환 작업의 실행 시 ADN모델일 경우 MakeADNXMLModel을 호출하여 XML모델로 변환함과 동시에 import된 IDN모델의 변환 작업을 수행한다. 또한 입력 받은 값이 IDN모델일 경우 MakeIDNXMLModel을 호출하여 XML모델로 변환한다. MakeADNXMLModel과 MakeIDNXMLModel은 ADN 모델과 IDN 모델을 XML모델로 변환하는 클래스 이다. ValidChecker는 XML

모델의 유효성 검사를 위한 클래스이다. 다음(그림 2)은 ADN모델이 XML형태로 변환된 결과를 보여준다.

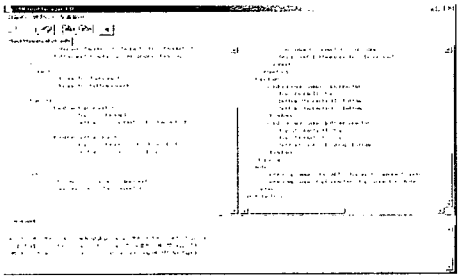

그림 2. Convert to XML 실행화면

USER의 요청에 의해 ADN모델을 변환할 경우 "Convert to XML" 메뉴를 통해 변환이 이루어지며, 변환된 XML모델 내용이 보여지고, XML파일은 유효성 검사 결과도 동시에 실행된다.

3.2 XML->ADL 변환 과정

다음(그림 3)은 XML모델을 ADN모델로 변환 과정이다. XML모델은 "Convert to ADL" 메뉴를 통해 변환이 이루어지며, 이때 변환 된 ADN 모델은 저장소에 저장되어 있는 XML모델 이름과 비교하여 XML모델에 대응되는 ADN모델을 찾아 보여주게 된다. StackVisualization이란 이름을 가진 XML형태의 모델이 ADN모델로 변환 된 결과를 보여준다.

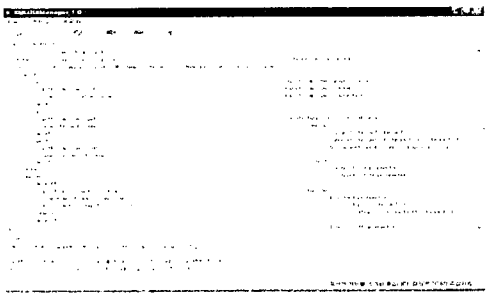

그림 3. Convert to ADL 실행화면

4. XML기반 ADL 모델 저장소

ADL 모델 저장소는 변환 과정 중에 생성된 모델 정보들을 저장하여 후에 검색이나 관리를 용이하게 하는 데에 목적이 있다. 본 연구팀은 컴포넌트 명세 언어인 idn과 아키텍처 명세 언어인 adn의 구문형식을 효율적으로 저장하고 관리하기 위해서 adn-저장소와 idn-저장소로 각각 생성하였다. 또한 XML기반 ADL 모델을 저장하기 위하여 본 연구팀은 저장소를 관계형 DBMS인 MYSQL과 Java를 이용하여 구현하였다. MYSQL에 저장된 adn-저장소와 idn-저장소의 구조는 다음(그림 4)과 같다. 먼저 adn-저장소구조는 각 모델별 데이터베이스는 한개 이상의 테이블을 가진다. 각각의 테이블을 선택 할 경우 오른쪽 창에 그 테이블의 속성을 칼럼 형태로 보여준다. 저장되어 있는 테이블 중 component라는 특정 테이블을 선택했을 경우의 그 테이블에 해당하는 칼럼과 레코드가 나타난다. idn-저장소 구조는 저장되어 있는 테이블의 이름과 크기, 저장된 레코드의 수들을 보여준다. 기본적으로 데이터베이스와 연동되는 응용프로그램은 하나의 데이터베이스를 생성할 때 하나의 Connector를 갖지만 본 연구팀은 adn-저장소와 idn-저장소를 모두 충족시키기 위하여 Connector 또한 두 가지로 구현하였다. Connector는 adn-저장소를 연결하는 DBConnector와 idn-저장소를 연결하는 DBIDNConnector가 존재한다. 다음은 두 개의 Connector DBConnector의 내용이다. 먼저(그림 5)는 DBConnector 클래스

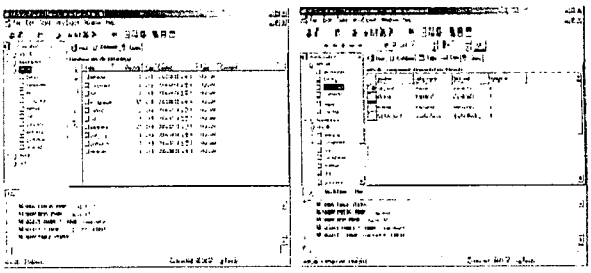

그림 4. adn-저장소 & idn-저장소 구조

의 일부분이다.

```
public class DBConnector{
   public static Connection getConn() {
                .
      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/adn_db","root","");
                return conn;
                }
}
```

그림 5. DBConnector 클래스

DBConnector 클래스는 adn-저장소와 응용프로그램을 연결해주는connector클래스로 riverManager.getConnection ("jdbc:DBMS:// localhost:3306/DB이름","사용자ID","비밀번호"); 형식으로 사용되며 DriverManager .getConnection함수는 connection객체를 리턴 한다. 다음 (그림 6)은 DBIDNConnector 클래스의 일부분이다. DBIDNConnector 클래스는 idn-저장소와 프로그램을 연결해주는 또 다른 Connector클래스이다. 다른 부분은 DBConnector와 같지만 데이터베이스 이름을 명시하는 부분만 idn_저장소로 바뀐 것을

```
public class DBIDNConnector
{
  public static Connection getConn(){
                .
     conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/idn_db","root","");
                .
                return conn;
                }
}
```

그림 6. DBIDNConnector 클래스

```
public void makeBody() throws java.io.FileNotFoundException, java.io.IOException,
ClassNotFoundException
{
                .
                .
                for(int i=0;i<impList.size();i++){
                .
                query = "insert into import values ("+""+imp.getPackPath()+""+","+""+imp.getN
ame()+""+","+primary_a+")";
                stmt = conne.createStatement();
                stmt.executeUpdate(query);
                stmt=null;
                }
```

그림 7. MakeADNXMLModel 클래스의 makeBody()함수

알 수 있다. 다음(그림 7)은 makeBody()함수의 일부분을 보여준다. query를 살펴보면 imp객체에서 각각의 정보를 추출해 오는 것을 알 수 있다. 정보가 저장되는 시점은 ADL의 모델 값을 추출하는 단계에서 이루어진다. 즉 "convert to XML" 명령이 실행되는 순간 데이터베이스에 insert되는 것이다. 예를 들어 adn파일의 하위 속성인 import는 MakeADNXMLModel 클래스의 makeBody()함수 호출 시 데이터베이스에 저장된다.

5. XML기반 ADL 모델 검색기

ADL모델 검색기는 USER가 저장소에 저장된 모델 정보를 쉽게 검색하도록 하는 것을 목적으로 한다. 이를 위해 검색기는 GUI 형태의 검색환경을 지원하며, combo box를 이용하여 ADN모델과 IDN모델을 구별하여 검색하도록 하였다. 따라서 사용자는 ADN

모델과 IDN모델 의 기본 형태를 미리 숙지하고 있어야 한다.

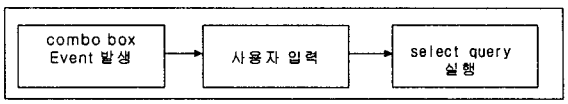

그림 8. XML 기반 ADL 모델 검색기 구조

XML 기반 ADL 모델 검색기 구조는 다음(그림 8)과 같다. 우선 검색할 모델이 무엇인지를 선택하기 위한 combo box event가 발생하고 그 후 사용자가 검색할 정보를 입력하면 입력한 정보에 맞는 모델을 찾아준다.

5.1 XML기반 ADL 모델 검색 과정

XML기반 ADL 모델 검색기의 검색과정에 사용되는 클래스는 XMLGUIManager와 DBSearch로 나뉜다. XMLGUIManager 클래스는 검색메뉴를 선택하기 위하여 사용되고, Search 클래스는 실제 검색을 수행하는 용도로 사용되었다. 먼저 (그림 9)는

```
public class XMLGUIManager extends JFrame implements ActionListener{
        public void actionPerformed(ActionEvent e){
                .
                .
                else if(obj == toolsearch) { // DB data search
                        DBSearch sc = new DBSearch(this);
                .
                .
                }
        }
}
```

그림 9. XMLGUIManager 클래스

XMLGUIManager 클래스의 이벤트 처리 함수의 일부분을 보여준다. 사용자가 검색 메뉴를 클릭할 경우 발생하는 이벤트에 대한 부분으로 검색 메뉴 선택 시 DBSearch가 실행되고, toolsearch라는 메뉴가 선택되면 현재 window의 핸들을 DBSearch로 넘겨주게 되며, 모든 검색에 관련 된 것은 DBSearch에서 이루어지게 된다. 다음(그림 10)은 DBSearch 클

```
public class DBSearch extends JDialog implements ItemListener, ActionListener{
        private ADNSearch as= new ADNSearch();
        private IDNSearch is = new IDNSearch();
        public DBSearch(XMLGUIManager cmain){
                }
                ..
        public void start(){
                jcbtable.addItemListener(this);
                showtable.addActionListener(this);
                this.setDefaultCloseOperation(WindowConstants.HIDE_ON_CLOSE);
        }
        public void itemStateChanged(ItemEvent ie) {
                if(ie.getSource()==jcbtable) {
                        table = ((String)ie.getItem()).trim();
                        if(table.equals("ADN")) { //adn파일을 선택한 경우
                                con.add("Center",as.createSearch());
                                con.validate();
                                this.setSize(350,400);
                        }
                        else if(table.equals("IDN")){ //idn파일을 선택한 경우
                                con.add("Center",is.createSearch());
                                con.validate();
                                this.setSize(350,450);
                        }
                        jcbtable.setEnabled(false);
                }
```

그림 10. DBSearch 클래스

래스의 일부분으로서, 검색 메뉴가 선택되면 DBSearch 클래스에서 검색 작업이 시작되는데, 먼저 itemStateChanged함수를 통해 combo box event가 발생하게 되며 선택한 모델에 따라 다른 Search 폼 형태를 보여준다. USER가 폼 형태에 따라 검색할 모델 정보를 입력하되 여러 개의 조건 중 하나의 값만 입력해도 검색이 가능하도록 구현하였다. 또한 검색할 정보를 입력한 후 확인버튼을 클릭하면 저장소에 저장되어 있는 데이터와 사용자가 찾고자 하는 데이터를 비교하여 일치하는 파일 찾게 된다.

5.2 XML기반 ADL 모델 검색 결과

다음(그림 11) 검색화면 Search-ⓐ는 Component에서 Filename이 StackArtist인 idn파일을 검색한 것으로서, idn 모델 검색을 위한 Dialog Box가 나타나게 되며, 사용자는 Dialog Box에 있는 여러 조건 중에 하나의 조건에 해당하는 값만을 입력하면 된다. 저장소에 저장되어 있는 데이터를 검색하여 일치하는 파일을 찾게 되면, (그림 11)의 Search_ⓑ와 같이 저장된 위치를 알려주는 검색 결과 창과 함께 (그림 11)의 Search_ⓒ와 같이 검색된 파일을 보여준다.

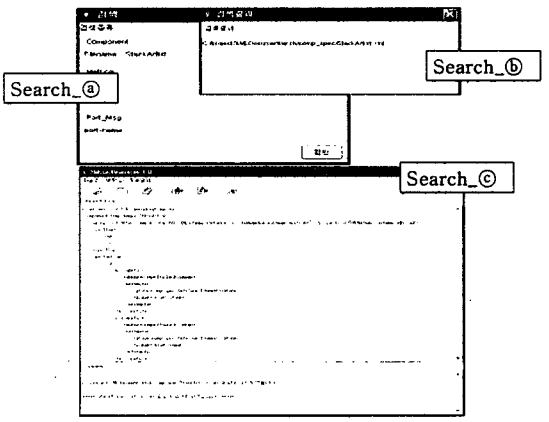

그림 11. 검색 결과 화면

6. 결론 및 향후 연구

본 논문에서는 ADL 모델관리기 내의 변환기의 변환 과정을 기술하였고, 또한 컴포넌트 명세 언어인 idn과 아키텍처 명세 언어인 adn의 구문형식을 효율적으로 저장하고 관리하기 위해서 adn-저장소와 idn-저장소로 각각 구현하였다. ADL 모델 검색기는 USER가 저장소에 저장된 모델 정보를 쉽게 검색하도록 GUI 형태의 검색환경을 기반으로 구현하였으며, 특히 본 논문에서는 XML기반 ADL 모델 검색기의 구현에 XMLGUIManager와 DBSearch 클래스가 사용되었다.

본 논문은 ADL 모델 관리를 사용자가 용이하게 하기 위해 관계형 데이터베이스로 구현하였다. 향후에는 XML과 호환이 용이한 객체지향형 데이터베이스로의 구현을 통해 소프트웨어의 재이용성과 ADL 모델 관리방법을 더욱 높일 수 있도록 연구할 계획이다.

7. 참고문헌

[1] 강미나, 승현우, "ADL 모델 관리기 설계를 위한 XML기반 ADL 메타 모델 정의" 정보과학회 추계 학술발표논문집 제 30권 2호, 2003.10

[2] 강미나, 양현미, 승현우, 전태웅, "ADL 모델 관리를 위한 XML 기반 ADL 모델 변환기" 정보과학회 춘계 학술발표논문집 제 31권 1호, 2004.4

[3] 이경하, 정명희, 홍의석 역, 이누이 미노루, 다니카 사토시, 다나카 유키히로 저, "실전 XML 데이터베이스 구축" 성안당,2002

[4] 천주석 역,Jon Duckett 외 8인 공저. "PROFESSIONAL XML Schemas", 정보문화사, 2002

[5] 손광수, "데이터베이스관리와 실습" 한빛미디어,2003

[6] Kal Ahmed 외, 김선태 옮김, "Professional Java XML", 정보문화사,2002

[7] Patrick Cauldwel외, 안성욱 지음. 황세진, 서민구, 이종훈, 김진중 옮김,"Professional XML Web Services",정보문화사,2002