

ADL 모델 관리를 위한 XML기반 ADL모델 변환기 설계 및 구현

강미나^o 양현미 승현우 전태웅*
 서울여자대학교 컴퓨터학과, 고려대학교 전산학과*
 {gmmn^o, merong, hwseung}@swu.ac.kr
 jeon@selab.korea.ac.kr*

The Design and Implementation XML Based ADL Model Converter for the Management of ADL Model

Mina Kang^o Hyunmi Yang Hyonwoo Seung Taewoong Jeon*
 Dept. of Computer Science, Seoul Women's University
 Dept. Computer Science, Korea University*

요 약

CBD(Component-Based Development)에 있어 컴포넌트 시스템은 잘 정의된 아키텍처를 기반으로 개발되어야 한다. 컴포넌트 시스템의 아키텍처를 정확하고 엄밀하게 설계, 분석하기 위해서는 아키텍처 기술 언어(ADL)의 사용이 필요하다. 아키텍처 모델링에 ADL을 효과적으로 사용하기 위해서는 ADL로 기술된 아키텍처 모델을 효율적으로 분석, 처리, 관리할 수 있는 지원 환경이 필수적이다. 본 논문에서는 ADL 모델 지원 환경 시스템의 구조를 기술하고 본 연구팀에서 개발 중인 ADL 모델 관리기의 구조를 기술한다. ADL 모델 관리기 중 ADL 모델 정보들을 XML 형태로 저장, 관리하기 위한 XML 기반 ADL 모델 변환기의 구조를 기술하며 ADL-XML 변환 방법을 설명하고 그 변환 결과를 보여준다.

1. 서 론

소프트웨어의 재사용을 위한 방법 중에 최근 빠르게 확산되고 있는 방식으로 컴포넌트 기반의 소프트웨어 개발(CBD : Component-Based Development)방식이 있다[1].

독립된 단위로 기능을 하는 소프트웨어 조각(Component)을 조합하여 소프트웨어를 개발하는 방식인 CBD에 있어 컴포넌트 시스템은 잘 정의된 아키텍처를 기반으로 개발되어야 하며, 아키텍처를 정확하고 엄밀하게 설계, 분석하기 위해서는 아키텍처 기술 언어(ADL : Architecture Description Language)의 사용이 필요하다. 따라서 아키텍처 모델링에 ADL을 효과적으로 사용하기 위해서는 ADL로 기술된 아키텍처 모델을 효율적으로 분석, 처리, 관리할 수 있는 지원 환경이 필수적이다.

본 연구는 ADL 모델 지원 환경 시스템 개발을 목표로 주관연구와 공동 연구로 나누어 수행된다. 주관연구에서는 ADL을 정의하고 아키텍처 모델의 구조적 행위적 성질들을 분석하여 정의된 스타일 규칙들의 위반 여부를 검사하는 방법 및 이를 지원하는 ADL 모델 분석기를 개발한다. 공동연구에서는 아키텍처 모델링 및 모델 분석 과정 중에 생성된 모델 정보들을 서로 다른 ADL 지원 도구들이 공유 할 수 있도록 XML 형태로 변환, 저장, 관리하는 방법 및 이를 지원하는 ADL 모델 관리기를 개발한다.

본 논문에서는 ADL 모델 지원 환경 시스템의 구조를 기술하고 공동연구 기관인 본 연구팀에서 개발 중인 ADL 모델 관리기의 구조를 기술한다. 또한 ADL 모델 관리기 중 ADL 모델 정보들을 XML 형태로 저장, 관리하기 위한 XML 기반 ADL 모델 변환기의 구조를 기술한다. XML 기반 ADL 모델 변환기는 내부적으로 ADL-XML 변환과 XML-ADL 변환을 수행하며 본 논문에서는 ADL-XML 변환 방법을 설명하고 그 변환 결과를 보여준다.

2. ADL 모델 지원 환경 시스템

아키텍처 모델링에 ADL을 효과적으로 사용하기 위해서는 ADL로 기술된 아키텍처 모델을 효율적으로 분석, 처리, 관리할 수 있는 지원 환경이 필수적이다. ADL 모델 지원 환경 시스템 구조는 (그림 1)과 같이 크게 ADL 모델 분석과 ADL 모델 관리의 두 부분으로 구성된다.

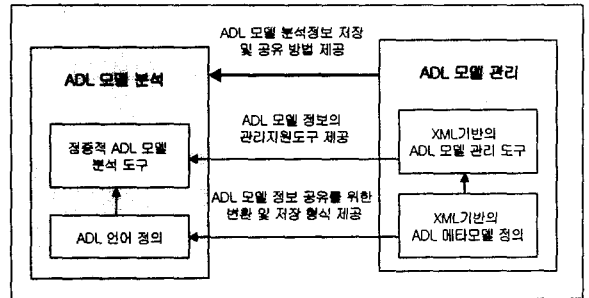


그림 1. ADL 모델 지원 환경 시스템 구조

ADL 모델 분석에서는 ADL 언어를 정의하고 ADL 모델 분석 도구를 개발한다. ADL 분석의 범위는 ADL로 기술된 아키텍처 모델에 대한 구문 분석, 의미 분석, 일관성 분석으로 구문 오류(syntax error)를 검사하고 파스 트리를 생성하는 ADL 파서 개발을 포함한다.

ADL 모델 관리에서는 ADL 모델을 XML 형태로 변환, 저장, 관리하는 것을 목적으로 XML 기반 ADL 메타모델을 정의하고, ADL 모델 관리 도구를 개발한다. ADL 메타 모델은 XML Schema로 정의하며 이는 XML 형태로 변환된 ADL 모델의 유효성 체크에 사용된다. 또한 ADL 관리 도구는 XML기반 ADL 모델 변환기, 변환

본 연구는 한국과학재단 목적기초연구(R01-2002-000-00044-0) 지원으로 수행되었음

모델 분석 정보 저장을 위한 저장소, 저장된 분석 정보 검색을 위한 검색기 개발을 포함한다.

2.1 ADL 모델 관리기

ADL 모델 관리기는 아키텍처 모델 분석 과정 중에 생성된 아키텍처 모델 정보들을 서로 다른 ADL 지원 도구들이 공유할 수 있도록 XML 형태로 변환, 저장, 관리하기 위한 도구이다[2]. ADL 모델 관리기는 (그림 2)와 같이 XML기반 ADL모델 변환기, 저장소, 모델 검색기로 구성되어 있다.

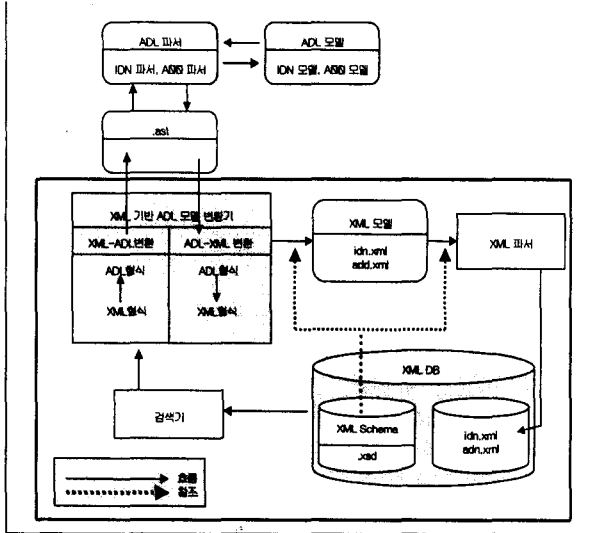


그림 2. ADL 모델 관리기 구조

XML기반 ADL모델 변환기는 ADL-XML변환, XML-ADL변환을 수행한다. ADL파서에 의해 파싱된 ADL모델 정보들은 ADL-XML변환을 통해 XML형태의 모델로 변환되어 변환된 XML 형태의 모델 정보들은 DB에 저장된다. 저장된 모델 정보들은 특정 요구에 의해 검색기를 통해 검색되어 XML-ADL변환을 통해 다시 ADL 모델 정보들로 복원된다.

3. XML 기반 ADL 모델 변환기

ADL 모델 관리의 아키텍처 모델링 및 모델 분석 과정 중에 생성된 모델 정보들을 서로 다른 ADL 지원 도구들이 공유할 수 있도록 XML 형태로 변환, 저장, 관리하는데 목적이 있다. 따라서 ADL 모델 정보들을 XML 형태로 변환하기 위한 변환기의 개발은 필수적이다. XML 기반 ADL모델 변환기의 구조는 (그림 3)과 같다.

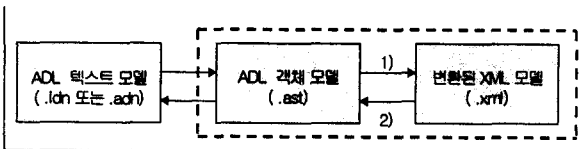


그림 3. XML 기반 ADL 모델 변환기 구조

XML 기반 ADL모델 변환기에서는 텍스트 형태의 ADL 모델이 파서를 통해 파싱된 형태의 ADL 객체 모델을 XML 형태로 변환하는 1) ADL-XML 변환과 변환된 XML 모델을 ADL 객체 모델로 역 변환하는 2) XML-ADL 변환을 수행한다.

3.1 ADL-XML 변환

ADL 모델 정보들을 XML 형태로 변환하기 위한 ADL-XML 변환 과정은 다음과 같다. 먼저 IDN 모델, ADN 모델에서 모델 정보를 얻는다. 여기서 IDN(Interface Definition Notation)모델은 하나의 컴포넌트에 대한 모델이고 ADN(Architecture Definition Notation) 모델은 컴포넌트 여러 개가 모여 만들어진 아키텍처에 대한 모델이다. 다음으로 추출된 각각의 모델 정보를 DOM파서(아파치 그룹에서 만든 Xerces파서)를 이용하여 DOM트리에 삽입하고 XML 모델로 생성 한 뒤 XML Schema로 정의된 ADL 메타모델을 기준으로 유효성을 검사한다[3].

ADL을 XML로 변환하는 작업에서 사용되는 클래스는 XMLGUIManager, MakeADNXMLModel, MakeIDNXMLModel, ValidChecker이다. 이 중 XMLGUIManager는 USER가 변환작업으로 사용하는 클래스 이고 MakeADNXMLModel과 MakeIDNXMLModel은 ADN 모델과 IDN모델 각각을 XML 모델로 변환하는 클래스 이며 ValidChecker는 XML모델의 유효성 검사를 위한 클래스이다. ADL모델을 XML형태로 변환하는 전체적인 과정은 XMLGUIManager 클래스를 통해 이뤄진다. 이 클래스는 사용자에게 쉬운 ADL-XML 변환 작업을 제공하기 위해 작업환경을 GUI로 만든 것으로 (그림 4)는 XMLGUIManager 클래스의 일부이다.

```
public class XMLGUIManager extends JFrame implements
ActionListener{
    public void actionPerformed(ActionEvent e){
        Object obj = e.getSource();
        :
        elseif(obj == toXML || obj == tooltoXML)
        { //XML로 변환

            convertToXML(adName,xmlName);
        //XML로 변환
        }
        void convertToXML(String adName, String xmlName){
            // ADL모델을 XML 모델로 변환
        try{
            // 모델 정보 추출
            String workPath,packPath,name;
            workPath="C:\\workspace\\WWW\\Home\\WWW\\Arch";
            int comma=adName.lastIndexOf(".");
            name = adName.substring(0,comma);
            if((adName.substring(comma+1)).equals("adn")){
                //ADN 모델일 경우
                packPath = "";
                MakeADNXMLModel driver=new Make
                ADNXMLModel(workPath,packPath, name);
                driver.makeBody();
                bottomText.append(driver.resultMsg);
            }
            else{ //IDN 모델일 경우
                packPath = "comp_spec";
                MakeIDNXMLModel driver=new Make
                IDNXMLModel(workPath, packPath, name);
                driver.makeBody();
                bottomText.append(driver.resultMsg);
            }
        }
    }
}
```

그림 4. XMLGUIManager 클래스

USER에 의해 XML변환작업 요청이 들어오면 XMLGUIManager에서는 먼저 ADL모델을 XML모델로 변환하기 위해 변환 될 모델로부터 workPath, packPath, name 등의 정보를 입력받는다. 다음으로 입력받은 모델이 IDN 모델일 경우 MakeIDNXMLModel을 호출하여 XML 모델로 변환하며, ADN 모델일 경우 MakeADNXMLModel을 호출하여 XML 모델로 변환함과 동시에 import된 IDN 모델의 변환 작업도 수행한다. 이때 import된 IDN 모델 변환은 내부적으로 MakeIDNXMLModel을 호출하여 수행한다. 마지막으로 변환 과정 중 내부적으로 ValidChecker 클래스를 호출하여 변환된 XML모델의 유효성을 검사한다.

3.2 XML기반 ADL모델 변환기 GUI 환경

ADL-XML 변환 작업에 대한 상황, 결과 메시지 등은 (그림 5)와 같이 GUI환경을 통해 USER에게 전달된다.

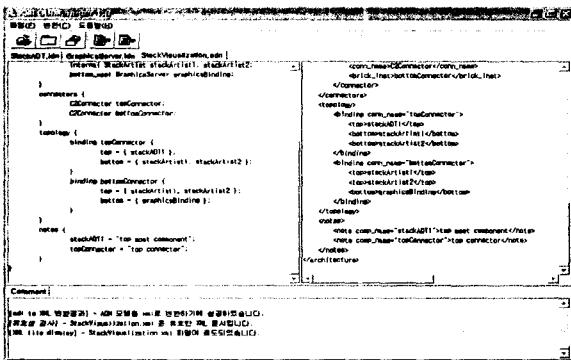


그림 5. XML기반 ADL모델 변환기 GUI 환경

GUI환경은 Swing의 JTabbedPane 클래스를 사용하여 여러 개의 변환 작업을 같은 작업환경에서 동시에 할 수 있도록 하였으며, 사용자에게 작업상황 및 결과 메시지를 전달하기 위해 Comment 영역을 마련하였다.

변환 작업 시 우선 변환하고자 하는 ADL모델을 불러온 후, 도구모음이나 메뉴를 통해 "Convert to XML"을 수행하면 ADL모델의 XML 모델로의 변환이 이루어짐과 동시에 변환된 XML 파일의 내용이 ADL 텍스트 파일 옆에 나란히 출력됨으로써 변환된 모델과 원본 사이의 비교가 용이하다. 또한 USER는 변환시 발생하는 상황이나 결과 메시지 등은 Comment창을 통해 볼 수 있다.

3.3 ADL-XML 변환 결과

앞 절에서 기술한 XML기반 ADL모델 변환기 GUI 환경에서 StackADT라는 컴포넌트를 명세해 놓은 StackADT.idn 파일(그림 6)의 StackADT.xml 파일(그림 7)로의 변환결과는 다음과 같다.

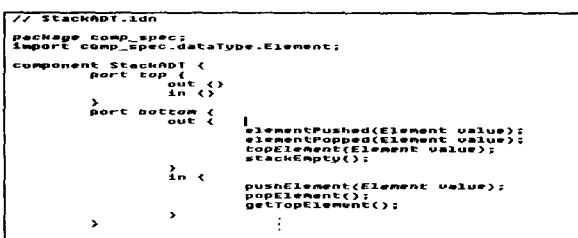


그림 6. StackADT.idn

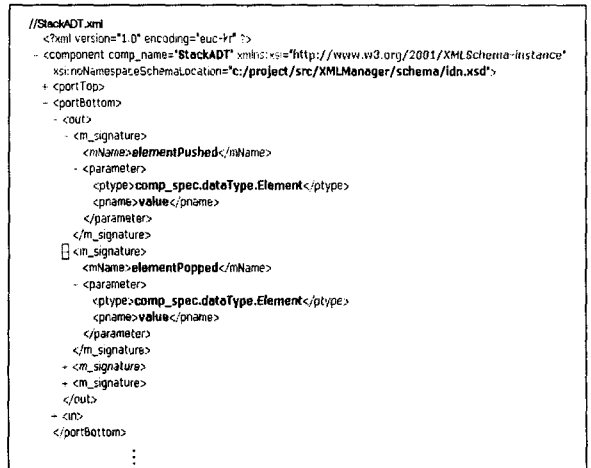


그림 7. StackADT.xml

4. 결론 및 향후 연구과제

본 논문에서는 아키텍처 기반의 소프트웨어 개발(CBD)에 효과적으로 사용될 수 있는 ADL 모델 지원 환경 시스템을 제안하였으며 현재 개발 중인 ADL 모델 지원 환경 시스템 내의 ADL 모델 관리의 구조를 기술하였다. 또한 ADL 모델 관리기 중 ADL 모델 정보들을 XML 형태로 저장, 관리하기 위한 XML 기반 ADL 모델 변환기의 구조를 기술하였다. 변환기는 아키텍처 모델링 및 모델 분석 과정 중에 생성된 모델 정보들을 서로 다른 ADL 지원 도구들이 공유 할 수 있도록 XML 형태로 변환하는 역할을 하므로 XML 기반 ADL 모델 관리에 있어 필수적이다.

향후에는 본 논문에서 XML 형태로 변환된 ADL 모델 정보들을 저장하기 위한 저장소와 저장소로부터 USER의 query에 따른 모델 정보를 검색할 수 있는 검색기를 개발할 계획이다. 또한 검색된 XML 형태의 모델 정보들을 ADL 모델 정보로 역변환 하는 XML-ADL 변환도 구현할 계획이다.

5. 참고문헌

- [1] 노성환, 신동익, 최재각, 전태웅, "C2스타일의 아키텍처 기술을 지원하는 ADL정의" 정보과학회 추계 학술발표논문집 제28권 2호, 2001.10
- [2] 강미나, 승현우, "ADL 모델 관리기 설계를 위한 XML 기반 ADL 메타 모델 정의" 정보과학회 추계 학술발표논문집 제 30권 2호, 2003.10
- [3] <http://xml.apache.org/xerces2-j/index.html>
- [4] <http://java.sun.com/j2se/1.4.2/docs/api/>
- [5] 최범균, 김인희, 김봉수 역, 브릿 맥라프린 저, "자바와 XML", 한빛미디어, 2002
- [6] 김명주 역, Hiroshi Maruyama Kent Tamura Naohiko Uramoto 저, "XML and Java Developing Web Applications". 2000
- [7] 신민철, 이규미, 채규태 저, "기초에서 실무까지 XML", 프리렉, 2003