XML 기반 컴포넌트 명세서 생성 지원 도구 설계

박준범° 최한석 오수열
목포대학교 정보공학부
ospark01@hanmail.net, chs@kware21.com, syoh@chungkye.mokpo.ac.kr

A study on the XML-based software Components Specification Method and Supporting Design

Joon-Bum Park° Soo-Lyul Oh Han-Suk Choi
Division of Information Computer Engineering, Mokpo National University

요 약

본 논문은 컴포넌트 S/W의 저장, 관리, 유통 활성화를 위하여 컴포넌트 명세서를 규격화하고 규격화된 컴포넌트 명세서를 표준 메타언어인 XML 기반으로 모델링하는 것이다. 클래스 라이브러리의 한계점들을 극복하고 소프트웨어 재사용성을 향상시키기 위한 단위로 만들어진 컴포넌트 단위의 재사용 기법의 XML 기반의 컴포넌트 명세서로 인해 더욱 활성화 될 수 있을 것이다.

또한, 컴포넌트 메타데이터 및 리소스 정보, 외부 인터페이스들의 논리적 구조 및 시맨틱 표현 기법을 연구하고 컴포넌트 명세서 표현을 위한 메타데이타 모델을 근거로 컴포넌트 명세서를 효과적으로 생성할 수 있는 GUI 기반 명세서 기반 도구를 설계한다.

1. 서론

소프트웨어 규모는 커지고 있으나, 소프트웨어 개발 기술의 발전 속도는 하드웨어 발전 속도에 비해 낙후된 실정이다. 대부분의 소프트웨어 프로젝트들은 개발 시간이 지연되거나 예산을 초과하거나, 고객이 원하는 고품질의 소프트웨어를 생산하지 못함으로 인해 실패하는 경우가 많다. 이처럼 소프트웨어의 개발과 유지보수 비용의 증가로 인해 소프트웨어 위기 문제가 발생하게 되었다.[1][2][3] 이에 대한 대안으로 소프트웨어 재사용 기술이 소개되면서 소프트웨어 고장 개념이 도입되었다. 다른 공학 분야에서는 컴포넌트를 이용한 제품의 효율적인 개발이 보편화되어 있다. 이처럼 소프트웨어 컴포넌트들도 새로운 소프트웨어를 개발할 때 빌딩 블록으로 사용될 수 있다. 따라서 소프트웨어를 처음부터 전부 개발함으로써 생기는 비용이나 위험 요소들을 감소시킬 수 있게 된다.[4][5][6][7]

2. 컴포넌트 명세서 구조

2.1 컴포넌트 정의 및 구성요소

컴포넌트에 대한 정의는 다양하다. Rational 사의 Philippe Krutchen은 "컴포넌트는 잘 정의된 아키텍처 상에서 어떠한 기능을 수행하는 시스템 독립적이면서 대치 가능한 부분으로서 인터페이스들의 집합에 대한 물리적인 구현을 제공한다"라고 정의한다.[4] Gartner 그룹은 "컴포넌트는 동적으로 바인드 할 수 있는 하나 이상의 프로그램들을 하나의 단위로 관리하는 패키지로서, 실행시간에 인터페이스를 통해 접근이 가능하다"라고 정의했으며,[5] Kozaczynski는 자발적인 비즈니스 객체 또는 비즈니스 로직을 소프트웨어로 구현한 것을 컴포넌트로 정의하고 있다.[6]

컴포넌트 구성요소로는 첫째, 컴포넌트는 모듈화되어 있는 단위라는 것이다.

둘째, 컴포넌트는 인터페이스를 가져야한다는 것이다.

셋째, 공통점으로 볼 수 있는 것은 컴포넌트는 구현되어 있는 단위라는 것이다.

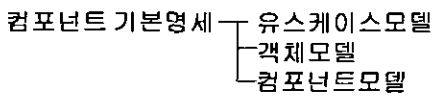
넷째, 컴포넌트는 아키텍처를 기반으로 한다는 것이다.

다섯째, 컴포넌트는 커스터마이즈가 가능해야 한다는 것이다.

2.2 컴포넌트 명세서 구조

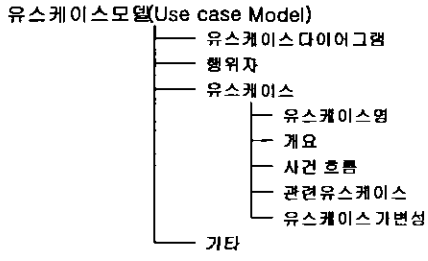
(1) 컴포넌트 명세서 구조 분석

컴포넌트 명세서는 <그림1>과 같이 3가지 영역으로 구분된다.



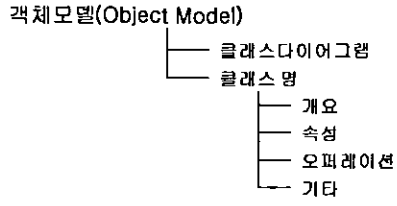
<그림 1> 컴포넌트 기본 명세

유스케이스 모델은 <그림 2>의 트리 구조로 표현된다.



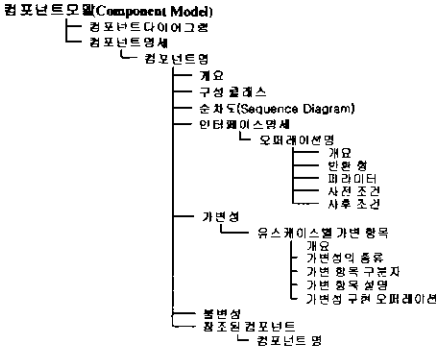
<그림 2> 유스케이스 모델

객체 모델은 <그림 3>의 트리 구조로 표현된다.



<그림 3> 객체 모델

컴포넌트 모델은 <그림 4>의 트리 구조로 표현된다.



<그림 4>컴포넌트 모델

(2) 컴포넌트 명세서 다이어그램 구조분석

각 컴포넌트의 다이어그램은 유스케이스 모델에서 유스케이스 다이어그램, 객체 모델에서 클래스 다이어그램, 컴포넌트 모델에서 컴포넌트 다이어그램과 순차도가 나온다.

유스케이스 다이어그램은 컴퓨터 시스템과 사용자가 상호작용하는 유스케이스를 그려놓은 다이어그램이다.

클래스 다이어그램은 시스템 내부에 존재하는 클래스들을 선별하여 나타내고 각 클래스들의 속성과 행위를 기입한다.

컴포넌트 다이어그램은 소프트웨어의 물리적 단위의 구성과 연결상태를 나타내게 된다.

시퀀스 다이어그램은 횡축을 시간축으로 하여 시간의 흐름을 나타내어 메시지의 순서에 역점을 두고 표현하다.

3. XML 기반 컴포넌트 명세서 모델링

3.1 컴포넌트 명세서 문서형 정의(DTD)

앞의2.2의 컴포넌트 명세서 구조의 <그림1>, <그림2>, <그림3>, <그림4>와 같이 컴포넌트 명세서 문서형 정의(DTD)를 정의 하였다.

각 모델에 대한 DTD는 다음과 같다.

(1) 유스케이스모델

```
<!ELEMENT UD_Usecase_Diagram (Usecase_Module,
Usecase_Actor+, Usecase_System*, Usecase_Usecase+,
Usecase_Generalization*, Usecase_Extends*,
Usecase_Entity_Position+, Usecase_Relation_Position+)>
    <!ELEMENT Usecase_Usecase_Module
            (Usecase_Module_Name,
            Usecase_Module_Document)>
        <!ELEMENT Usecase_Module_Name
            (#PCDATA)>
        <!ELEMENT Usecase_Module_Document
            (#PCDATA)>
                <중략>
```

(2) 객체모델

```
<!ELEMENT Class_Diagram (CD_Base,CD_Module,CD_Class+,
        CD_Operation*, CD_Attribute*,
        CD_Generalization*, CD_Dependency*,
        CD_Type*, CD_Association*, CD_Aggregation*,
        CD_Composition*, CD_Role*,
        CD_Entity_Position+, CD_Relation_Position+)>
    <!ELEMENT CD_Base EMPTY>
    <!ATTLIST CD_Base
        CD_Base_Items_Count CDATA #IMPLIED
        CD_Base_Version CDATA #IMPLIED>
                <중략>
```

(3) 컴포넌트모델

```
<!ELEMENT Component_Diagram (CPD_Base,
CPD_Module, CPD_Component+, CPD_Interface+,
CPD_Node*, CPD_Dependency+, CPD_Entity_Position+,
CPD_Relation_Position+)>
    <!ELEMENT CPD_Base EMPTY>
    <!ATTLIST CPD_Base
        CPD_Base_Items_Count CDATA #IMPLIED
        CPD_Base_Version CDATA #IMPLIED>
```

3.2 컴포넌트 명세서 XML 모델링

위의 DTD를 이용하여 XML 문서를 작성하였다.

```
<!-- 유스케이스 모델 표현 부분 -->
 <UD_Module_Parent_Package>2</UD_Module_Parent_Package>
            <UD_Module_Name>Usecase Diagram그림
</UD_Module_Name>
            <UD_Module_Document>유스케이스 다이어그램
설명부분입니다.
            </UD_Module_Document>
```

```
        </UD_Module>
        <UD_Actor UD_Actor_Identify="1">
            <UD_Actor_Name>Teller</UD_Actor_Name>
            <UD_Actor_Document>고객의 돈을 입출금 관리하는 은행 직원
            </UD_Actor_Document>
        </UD_Actor>
        <UD_System UD_System_Identify="2">
            <UD_System_Name></UD_System_Name>

<UD_System_Usecase_Identify></UD_System_Usecase_Identify>
        </UD_System>
        <UD_Usecase UD_Usecase_Identify="3">
            <UD_Usecase_Name>CM1:Register Customer</UD_Usecase_Name>
            <UD_Usecase_Document>개인이나 기업이 은행거래를 개시하고자 하는 경우 실명을 확인한 후 해당 고객에 정보를 등록하는 업무이다. 고객등록 업무는 고객이 은행에서 하고자 하는 거래를 위해 가장 먼저 해야하는 일로써 모든 은행 업무에서 이를 이용하게 된다.
            </UD_Usecase_Document>
        </UD_Usecase>
                    <중략>
```

4. XML 기반 컴포넌트 명세서 생성 지원도구 설계

4.1 지원도구 요구사항 분석

컴포넌트명세서 Editor는 컴포넌트 개발자들의 UML기술을 이용하여 개발과 동시에 컴포넌트명세서를 작성할 수 있도록 개발한다. 컴포넌트명세서의 구조화된 DTD를 이용하여 컴포넌트명세서의 일률성과 정확성을 기하고 수정, 삭제, 저장할 수 있어야 한다.

컴포넌트명세서 Editor 개발을 위한 요구사항은 다음과 같다.

첫째, UML을 이용한 컴포넌트명세를 가능하도록 환경을 제공한다.

둘째, 컴포넌트명세서 Editor 개발을 위한 구현 환경을 철저히 분석하여 윈도우 기반의 편리한 사용자 인터페이스를 제공한다.

4.2 지원 도구 설계

컴포넌트명세서의 특징은 아래와 같다

- 다이어그램의 작성 및 편집
- 다이어그램별 부가기능
- Drag & Drop 편집기능
- 미리 보기 화면이나 웹 브라우저에서 보기기능을 별도로 지원
- 다이어그램과 트리 구조의 작업이 일치
- 기능 및 화면설계

컴포넌트명세서 Editor에서 제공되는 주요 기능은 다음과 같다.

(1) 파일 메뉴

새파일 : 새로운 컴포넌트 명세서를 작성

열기 : XML문서를 불러와 유효성검사와 UML로 변환

저장 : 컴포넌트 명세서를 XML기반으로 작성

다른이름으로 저장 : 활성화된 창의 UML 다이어그램을 컴포넌트명세서로 저장

인쇄 : 활성화된 창의 컴포넌트명세서를 인쇄

닫기 : 활성화된 창을 닫는다.

모두닫기 : 열린 모든 창을 닫는다.

끝내기 : 모든 창을 닫고 저장 여부를 물은 후 프로그램을 종료한다.

(2) 편집메뉴

실행취소 : 실행한 명령을 취소한다.

잘라내기 : 선택된 부분을 지우고 기억시킨다.

복사하기 : 선택된 부분을 기억시킨다.

붙이기 : 잘라내기나 복사하기 한 내용을 삽입한다.

지우기 : 선택한 그림은 지운다.

(3) 보기 메뉴

기본도구상자 : 기본 도구모음 Toolbar를 보여줄 것을 선택한다.

컴포넌트명세서 보기 : 컴포넌트명세서를 보여준다.

DTD내용보기 : 컴포넌트명세서의 DTD구조를 보여준다.

(4) 창 메뉴

창정렬 : 바둑판모양, Cascade등 기본 창 정렬을 지원한다.

모두최소화 : 열려있는 창을 모두 최소화한다.

(5) 도움말

색인 : 도움말을 찾는다.

도움말 : 도움말의 내용을 보여준다.

5. 결론

현재 XML기반 컴포넌트 명세기법으로 XML문서만 생성하였으나 컴포넌트 명세를 위한 Diagram 모델링을 구현함으로써 규격화된 컴포넌트 명세서를 작성할 수 있게 되었다. 또한 UML 기반 Diagram의 세부사항을 입력하게되면 XML 컴포넌트 명세서가 생성될 수 있도록 모델링하였다.

또한 컴포넌트 모델링에 따른 컴포넌트 명세서를 자동으로 생성 할 수 있는 컴포넌트 명세 지원도구를 설계하였다. 향후에는 컴포넌트 명세서를 자동생성할 수 있는 컴포넌트 명세 지원도구를 개발하는 것이다.

6. 참고문헌

[1] James Rumbaugh, Michael Blaha, William Precerlani, Frederick Eddy, William Lorenzen, "Object-Oriented Modeling and Design", Prentice Hall, 1991.

[2] Ivar Jacobson, "Object-Oriented Softsware Engineering : A Use Case Driven Approach", Addison-Wesley,1994.

[3] Wojtek Kazaczynski and G. Booch, "Component-Based Software Engineering",IEEE Software, pp.34-36, Sept./Oct. 1998.

[4] A. W. Brown and K. C.Wallnau, "The Current State of CBSE", IEEE Software, pp.37-46, Sept./Oct. 1998.

[5] Desmon F. D'souza and A. C. Wills, Objects, Components, and Components with UML,Addison-Wesley,1998.

[6] Object Management Group, CORBA Component Model RFP, December 1998.

[7] Sun Microsystems, Enterprise JavaBens Specification 1.1 at URL:http://www.javasoft.com