

# XML 기반 컴포넌트 명세서 모델링

강선희\*, 이상돈\*, 최한석\*, 박서영\*\*

\*목포대학교 정보공학부 / \*\*한국전자통신연구원 컴퓨터소프트웨어 연구소

## XML-based Component Specification Modeling

Sunny Kang\*, SangDon Lee\*, HanSuk Choi\*, Seoyoung Park\*\*

\*Division of Information Engineering, Mokpo National University /

\*\*Computer Software Technology Lab., Electronics and Telecommunication Research Institute

### 요 약

컴포넌트는 소프트웨어 재사용 기술의 핵심 기술로서, 소프트웨어 개발 시간을 단축시키고 생산성을 향상시킨다. 인터넷의 급속한 보급과 컴포넌트 시장의 확산으로 컴포넌트 사용자들이 인터넷 상에서 검색, 저장, 유통하기 위하여 컴포넌트 명세서가 정의되어야 한다. 컴포넌트 명세는 컴포넌트 특성을 명확히 이해하기 위하여 구문적(syntactic), 의미(semantic), 그들 사이의 관계, 사용 문맥, 품질 속성들을 포함한 정보가 기술되어야 한다. 본 논문에서는 컴포넌트를 명세서기 위한 명세 요소들을 살펴 보며, 컴포넌트를 표현하기 위한 기술 방법들을 정의한다. 또한, 컴포넌트 메타데이터 및 리소스 정보, 외부 인터페이스들의 논리적 구조 및 시맨틱 표현기법을 연구하고 컴포넌트 명세서 표현을 위한 메타데이터 모델을 근거로 컴포넌트 명세서를 효과적으로 생성할 수 있는 XML 기반 컴포넌트 명세서 확장 모델을 개발하고자 한다.

### 1. 서론

소프트웨어 규모는 커지고 있으나, 소프트웨어 개발 기술의 발전 속도는 하드웨어 발전 속도에 비해 매우 낙후된 실정이다. 소프트웨어 개발의 시간 지연, 예산 초과, 유지보수 비용 등의 증가로 인해 소프트웨어 위기 문제가 발생하게 되었다. 이에 대한 대안으로 소프트웨어 재사용 기술이 소개되면서 소프트웨어 공장(Software Factory) 개념이 도입되었다. 소프트웨어 컴포넌트들이 새로운 소프트웨어를 개발할 때 빌딩 블록으로 재사용될 수 있어 컴포넌트를 이용한 제품의 효율적인 개발이 보편화되고 있다. 따라서 소프트웨어를 처음부터 전부 개발함으로써 생기는 비용이나 위험 요소들을 감소시킬 수 있게 된다.[1]

컴포넌트는 구조와 행위를 캡슐화하여 사용자가 내부 동작을 몰라도 인터페이스만 사용하여 원하는 서비스를 제공받는 장점을 가지고 있고, 이러한 컴포넌트를 기반으로 한 개발은 상호작용하는 컴포넌트들을 조립하는 과정으로 볼 수 있다. COTS(Commercial Off The Shelf)와 같은 소프트웨어 컴포넌트를 시스템으로 통합하는 경우가 많다. 이러한 통합에서 가장 큰 문제점은 조립된 시스템과 컴포넌트가 문맥적으로 일치하지 않는다는 것이고 문맥적 불일치가 쉽게 식별되지 않은 이유는 컴포넌트 명세서가 명확히 기술되지 못하여 개발자가 완벽히 이해하지 못하기 때문이다. 또한 컴포넌트 명세는 개발과 관련한 세부적인 내용보다는 오히려 인터페이스 설명을 나타

내고 있다. 그러나 이러한 인터페이스 설명은 자연어로 쓰여졌기 때문에 컴포넌트를 이해하기 위한 정확한 정보를 제공하지 못하므로 시스템과 문맥 불일치를 야기한다.[2]

현재 대부분의 컴포넌트 명세는 CORBA IDL(Interface Definition Language)과 같이 컴포넌트의 구문적(syntactic)인 측면만을 고려한다. 그러나 컴포넌트의 특성을 명확히 이해하기 위하여 컴포넌트의 의미(semantic), 그들 사이의 관계, 사용 문맥, 품질 속성들을 포함한 정보가 기술되어야 한다. 그러나 대부분의 기업들은 컴포넌트 명세서기법에 대한 표준이 없고, 명세서에 대한 데이터 모델링 방법 및 표현언어에 대한 기준이 없이 컴포넌트 개발을 하고 있는 실정이다. 그리고 응용 컴포넌트 부족 및 컴포넌트 공유체제의 미흡으로 인해 개발된 컴포넌트를 쉽게 찾아 활용할 수 있는 국가적인 공유체제가 구축되어있지 않으므로 불필요한 중복 개발이 발생하며, 정형화된 컴포넌트 개발 절차 및 기법, 컴포넌트 기반의 소프트웨어 개발 기술이 취약하며, 통합개발도구도 부족한 실정이다.

본 논문에서는 컴포넌트 메타데이터 및 리소스 정보, 외부 인터페이스들의 논리적 구조 및 시맨틱 표현기법을 연구하고 컴포넌트 명세서 표현을 위한 메타데이터 모델을 근거로 컴포넌트 명세서를 효과적으로 생성할 수 있는 XML 기반 컴포넌트 명세서 확장 모델을 개발하고자 한다.

### 2. 관련 연구

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 정의하고 있다[4]. Szyperski는 "컴포넌트는 계약 상으로 명시된 인터페이스들과 명확한 문맥 의존성을 가진 컴포지션의 단위로 독립적으로 보급될 수 있다"라고 정의하고 있다[4]. Kozaczynski는 자발적인(Autonomous) 비즈니스 객체 또는 비즈니스 로직을 소프트웨어로 구현한 것을 컴포넌트로 정의하고 있다[3,4]. Desmond Francis Souza는 컴포넌트는 소프트웨어 구현을 갖는 패키지이다라고 정의한다.

2.2 컴포넌트 명세서 구성요소 분석

컴포넌트 명세서에는 컴포넌트를 인터넷상에 개방하면서 그 컴포넌트에 대한 검색을 하거나 컴포넌트를 이해하고자 하는 사용자들에게 도움이 되는 정보들이 상세하게 기술되어야 한다.

2.2.1 검색 기반의 명세서 구성 요소

인터넷상이나 상용으로 컴포넌트를 사용할 경우에는 컴포넌트의 기능(Functionality), 운영체제, 언어, 시스템과 같은 환경(Environment), 매소드(Method) 및 입력/출력 매개변수(Input/Output Parameter), 성능(Performance), 제약성(Limitations), 데이터베이스 사용 등과 같은 서비스 수준 등이 포함되어야 하고 컴포넌트가 가져야하는 정보들로서 컴포넌트 공급회사 정보, 컴포넌트에 대한 일반적인 정보와 기술적 정보, 시스템 요구사항, 라이선스 정보, 가격 정보, 컴포넌트의 버전이나 구성정보 등을 포함해야 한다.[5]

2.2.2 공용 컴포넌트의 명세서 구성 요소

본 논문에서는 한국전자통신연구원에서 개발한 공용 컴포넌트로서 대부분의 영역에 공통적으로 사용되는 컴포넌트를 가지고 구성 요소를 살펴보고자 한다. 1999년에 개발한 Banking 컴포넌트를 기준으로 분석한 결과 아래와 같은 구성요소로 이루어져 있음을 알 수 있다.

- 가) 개요
- 나) 용어 정리
- 다) 공용 컴포넌트 계층 구성도
- 라) 영역 분류도
- 마) 컴포넌트 분류표
- 바) 컴포넌트 상호 관계도
- 사) 컴포넌트 명세
  - 아) Component Context Diagram
  - 자) Component Diagram
  - 차) Component Interaction Diagram
  - 카) Component Interface 명세

2.3 XML 요소 기술 분석

XML(Extensible Markup Language)은 HTML과 SGML과 함께 일반 마크업 언어의 일종으로 HTML에서의 간결한 데이터 표현 양식과 웹 브라우저에서의 뛰어난 표현 능력뿐만 아니라 SGML에서의 뛰어난 데이터 표현 능력을 동시에 갖도록 설계되었다. XML 문서는 태그(tag), 요소(element), 속성(attribute), 처리명령, 주석문으로 구성된다. 태그는 문맥을 나타내는 것

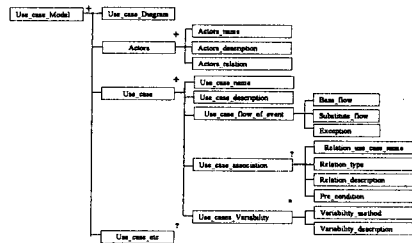
으로 문서의 의미를 표현할 수 있게 해준다. 요소는 태그와 태그 사이의 텍스트를 일컫는다. 태그는 속성을 가질 수 있으며 처리 명령은 파싱을 위하여 파서에 전달되는 정보이며 주석문은 문서에 대한 설명을 나타낸다. XML 문서의 형태는 요소(element)로 구성되어 있으며, 요소는 <tag> text </tag> 형태를 갖는다. 여기서 text는 요소의 형태를 반복적으로 취할 수 있기 때문에 XML 문서는 트리 형태의 구조적 표현이 가능하다.

3. 컴포넌트 명세서 확장 모델

컴포넌트 명세서는 한국전자통신연구원에서 개발하는 공용 컴포넌트를 명세하는 것을 기본으로 컴포넌트 명세 지침에 따라 명세서의 모델을 구성하고자 한다. 이번 장에서는 컴포넌트 명세서를 유스케이스 모델(use case model), 객체 모델(object model), 컴포넌트 모델(component model)로 구성하는 컴포넌트 명세서 확장 모델을 정의하고자 한다.

3.1 유스케이스 모델(Use case Model)

유스케이스 모델은 특정 도메인 내에서 컴포넌트를 개발하는 대상 응용 시스템들에 대한 기능들을 정의한 것으로 한다. 유스케이스 모델은 UML 표기법에 따른 유스케이스 다이어그램(use case diagram)과 행위자(actor), 유스케이스(use case) 등의 구성 요소들로 이루어진다. 유스케이스 모델을 분석한 결과 <그림 1>와 같은 트리 구조로 표현하는 것을 정의한다.



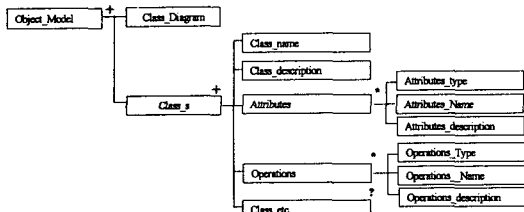
<그림 1> 유스케이스 모델 트리 구조

3.2 객체 모델 (Object Model)

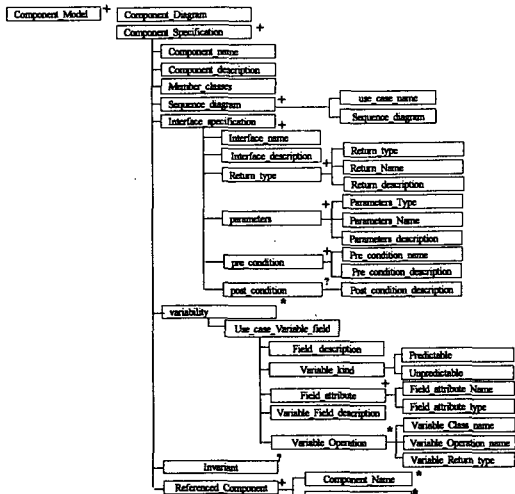
컴포넌트를 식별 추출하려는 대상 도메인의 응용 시스템들의 부분 집합에 대해 시스템의 객체(클래스)를 모형화 한다. 여기에서 식별된 클래스들을 기반으로 컴포넌트를 추출한다. 객체 모델은 식별된 클래스들의 관계, 기수성(multiplicity), 속성(attributes), 오퍼레이션(operation)을 기술한 class diagram, 클래스도에 있는 각 클래스에 대해 개별적으로 기술하는 클래스들로 구성되어 있으며, <그림 2>와 같은 구조로 표현하는 것을 정의한다.

3.3 컴포넌트 모델 (Component Model)

객체 모델을 바탕으로 컴포넌트를 식별하고 컴포넌트간의 의존성(dependency)을 보여주는 컴포넌트 다이어그램(component diagram)과 식별된 컴포넌트 단위로 기술하는 컴포넌트 명세 부분으로 컴포넌트 모델을 구성한다. 컴포넌트 명세 부분에서는 컴포넌트 명, 컴포넌트에 대한 개요, 구성 클래스(member classes), 순차도(sequence diagram), 인터페이스 명세(interface specification), 가변성(variability), 불변성(invariant), 참조된(referenced) 컴포넌트 등을 명세하는 구성으로 <그림 3>과 같은 구조로 표현하는 것을 정의한다.



<그림 2> 객체 모델 트리 구조



<그림 3> 컴포넌트 모델 트리 구조

#### 4. XML 기반 컴포넌트 명세서 모델링

컴포넌트 사용자가 컴포넌트를 검색하고 컴포넌트를 이해하는데 필요한 정보를 컴포넌트 제공자들은 기술할 수 있도록 하여야 한다. 컴포넌트 명세서를 작성하기 위하여 컴포넌트 명세서 확장 모델에 따라 XML DTD(Document Type Definition)을 정의한다.

##### 4.1 컴포넌트 명세서 확장 모델의 DTD 생성

```

<!ELEMENT Component_Description_Document
  (Use_case_Model+, Object_Model+, Component_Model+)
<!ELEMENT Use_case_Model
  (U_Use_case_Diagram, U_Actors+, U_Use_case+, U_Use_case_etc?+)
<!ELEMENT U_Use_case_Diagram (#PCDATA)
<!ELEMENT U_Actors
  (U_Actors_name, U_Actors_description, U_Actors_relation)
<!ELEMENT U_Use_case (U_Use_case_name, U_Use_case_description, U_Use_case
  flow_of_event, U_Use_case_association?, U_Use_case_Variability?)
<!ELEMENT U_Use_case_etc (#PCDATA)
<!ELEMENT Object_Model (O_Class_Diagram, O_Class+)
<!ELEMENT O_Class_Diagram (#PCDATA)
<!ELEMENT O_Class (O_Class_name, O_Class_description, O_Attributes*,
  O_Operations?, O_Class_etc?)
<!ELEMENT Component_Model (C_Component_Diagram,
  C_Component_Specification+)
<!ELEMENT C_Component_Diagram (#PCDATA)
<!ELEMENT C_Component_Specification (C_Component_name, C_Component_desc
  ription, C_Member_classes, C_Sequence_diagram+, C_Interface_specification+,
  C_variability, C_Invariant?, C_Refernced_Component+)
<!ELEMENT C_Component_name (#PCDATA)
<!ELEMENT C_Invariant (#PCDATA)
<!ELEMENT C_Refernced_Component(Component_Name*, Component_Parameters*)
  
```

#### 4.2 컴포넌트 명세서 XML 표현

```

<?xml version="1.0" encoding="EUC-KR" ?>
<!DOCTYPE Component_Description_Document (View Source for full
doctype...)
<Component_Description_Document>
<Use_case_Model>
<U_Use_case_Diagram>그림부분입니다</U_Use_case_Diagram>
<U_Actors>
<U_Actors_name>장우현</U_Actors_name>
</U_Actors>
<U_Use_case>
<U_Use_case_name>CM1:Register Customer(고객등록)</U_Use_case_name>
<U_Use_case_description>개인이나 기업이 은행 거래를 개시하고자 하
는 경우 실명을 확인한 후 해당 고객에 대한 정보를 등록하는 업무이다. 고객등록
업무는 고객이 은행에서 하고자 하는 거래를 위해 가장 먼저 해야하는 일로
써 모든 은행 업무에서 이를 이용하게 된다.</U_Use_case_description>
</U_Use_case>
<U_Use_case_etc>기타(선택사항)</U_Use_case_etc>
</Use_case_Model>
<Object_Model>
<O_Class_Diagram>그림부분입니다</O_Class_Diagram>
<O_Class>
<O_Class_name>해당 클래스 명</O_Class_name>
<O_Class_description>개인고객과 기업고객에 의해 상속을 받은 상위 클래스
이며, 한 고객은 여러 개의 계좌를 가지고 있을 수 있다. 또한 계좌를 신장하
기 위해서는 고객의 정보가 등록되어야 한다.</O_Class_description>
<O_Attributes>
<Attributes_type>string</Attributes_type>
<Attributes_type_name>customerID</Attributes_type_name>
</O_Attributes>
<O_Class_etc>기타</O_Class_etc>
</O_Class>
</Object_Model>
<Component_Model>
<C_Component_Diagram>그림부분입니다</C_Component_Diagram>
<C_Component_Specification>
<C_Component_name>해당 컴포넌트 명</C_Component_name>
</C_Component_Specification>
</Component_Model>
</Component_Description_Document>
  
```

#### 5. 결론 및 향후 연구과제

본 논문에서는 컴포넌트 명세서를 위한 확장 모델을 정의하여 기술하였다. 결론을 요약하면 다음과 같다.

첫째, 컴포넌트의 정의와 구성요소들을 파악하고, 컴포넌트 명세서에 필요한 구성요소들을 분석하였다. 둘째, 컴포넌트 명세서를 구성하는 정보의 모델링 측면에서 컴포넌트의 요소, 세부 속성들을 도출 하였으며 XML 기반 명세서를 연구하였다. 셋째, 컴포넌트의 규격화를 위해서 XML 기반의 DTD를 개발하였고, 적용에 필요한 요소와 애트리뷰트를 추출하였다. 마지막으로 DTD를 적용한 XML 기반 컴포넌트 명세서를 개발하였다.

현재 XML 기반 컴포넌트 명세서기법으로 XML 문서는 개발되었으며 앞으로 컴포넌트 명세서를 위한 입력도구를 개발할 계획이다. 그리고 XML 문서에서 다이어그램에 대해서 분석하고 표현하지 못하였으나 앞으로 분석 후 JAVA(Swing)으로 다이어그램을 모델링할 계획이다. 다이어그램이 모델링되면 UML기반의 입력도구를 개발할 예정이다.

#### 참고 문헌

- [1] 김수동, 컴포넌트 정의 및 관련 기술동향, 소프트웨어공학회지 12권 3호(1999.9)
- [2] 박찬진, 이명정, 우치수, 컴포넌트 명세서 기법, 소프트웨어공학회지 12권 3호(1999.9)
- [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] N. H. Lassing, D. B. B. Rijsenbrij, and J. C. van Vliet, "A View on Components," Proc. of 9th Int. Workshop on Database and Expert Sys. Applications, pp.768-777, 1998