

XML 기반 인터페이스 명세 중심의 컴포넌트 저장소 모델

김경민, 김태웅, 김정현, 김태공, 최항목
인제대학교 전산학과

e-mail:(kmkim, twkim, jhkim, ktg, hmchoi)@cs.inje.ac.kr

An XML-Based Component Repository Model Focused on Interface Specification

Kyung-Min Kim, Tae-Woong Kim, Jeung-Hyun Kim,
Tae-Gong Kim, Hang-Mook Choi
Dept of Computer Science, Inje University

요 약

시스템이 복잡해지고 대형화됨에 따라 소프트웨어 생산성, 품질, 효율성을 높이기 위해서 부품화와 재사용의 특성을 지닌 컴포넌트가 여러 응용프로그램에서 개발되고 있다. 따라서 컴포넌트 기반 소프트웨어공학에서는 컴포넌트를 분류, 검증, 검색하기 위한 컴포넌트 명세화 방안과 컴포넌트 기반의 개발을 위한 개발공정에 대해 많은 연구가 요구되고 있다. 이에 본 논문에서는 컴포넌트의 실제 재사용을 위해 컴포넌트의 조립 시점 정보를 나타내는 인터페이스 명세 중심의 컴포넌트 저장소 모델을 제안한다. 본 논문에서 제안하고 있는 컴포넌트 명세에는 인터페이스 정보 뿐 아니라 상속 등과 같은 재사용에 대한 정보와 컴포넌트들간의 결합관계를 나타내는 상호작용에 관한 정보를 포함하며, 이 명세는 이식성 및 구조적 표현이 뛰어나고 다양한 형태의 문서로 확장 가능한 XML을 사용하여 기술한다.

1. 서론

컴포넌트를 사용한 소프트웨어 시스템 개발 기술은 과거 구조적 방법이나 객체 지향 방법이 해결하지 못하였던 소프트웨어 재사용과 생산성, 시스템 품질 관리 등에 대한 새로운 해결책으로 제시되고 있다[1]. 이처럼 소프트웨어 시스템을 하나의 컴포넌트 조합으로 설계하고 구현하는 것은 이제 더 이상 새로운 개념이 아니며 규모가 큰 소프트웨어 개발은 컴포넌트의 적절한 검색, 검증, 그리고 통합 과정을 통해 이를 수 있게 되었다.

이러한 소프트웨어 컴포넌트를 재사용하기 위해서는 재사용 될 수 있는 컴포넌트들을 분류하여 컴포넌트 저장소에 저장하고, 개발자의 요구와 일치하는 컴포넌트들을 컴포넌트 저장소로부터 검색해야 한다. 그러나 현재 연구, 개발되고 있는 컴포넌트 저장소는, 재사용을 위해 다른 소프트웨어나 컴포넌트와

연결할 수 있는 방법을 정의하고 있는 인터페이스 [2]에 대한 명세를 간과한 채 기능중심의 컴포넌트 저장소가 대부분[3,4]이며 인터페이스에 대한 명세를 포함하더라도 컴포넌트들 간의 상호작용에 관한 조건은 잘 표현하지 못하고 있다[5].

이에 본 논문에서는 컴포넌트 재사용의 활성화를 위해 컴포넌트의 특성 및 용도를 나타내는 일반적인 정보뿐만 아니라 조립시점의 정보를 나타내는 인터페이스 정보, 상속등과 같은 재사용에 대한 정보, 컴포넌트간 결합관계를 나타내는 상호작용에 관한 정보를 포함하는 명세 기반의 컴포넌트 저장소 모델을 제안한다. 또한 이 명세를 XML로 기술함으로써, 구조적이며 이식성이 뛰어나고, 변경과 확장성이 용이하며 웹 환경에서 적합한 컴포넌트를 검색할 수 있도록 한다. 그리고 사용된 태그들과 데이터 타입을 DTD로 정의하여 XML 문서의 유효성을 유지한다.

2장에서는 본 논문에서 제안하는 저장소 모델의 기반이 되는 컴포넌트 명세의 상세 정보를 표현하는 XML DTD를 제안하고 3장에서 이 명세의 적용 사례를 들며 4장에서 결론을 맺는다.

2. 컴포넌트 저장소

컴포넌트의 개발은 세계 곳곳에서 이루어지고 있으며 이들 컴포넌트의 숫자가 기하 급수적으로 증가하고 있는 추세이다. 때문에 막대한 숫자의 컴포넌트들 사이에서 사용자가 자신의 요구사항을 만족하는 가장 적합한 컴포넌트를 선택하고 사용하기 위해서는, 컴포넌트에 대한 특성을 상세히 기술하는 컴포넌트 명세서를 사용해야 한다.

이에 본 연구에서는 사용자 요구사항에 맞는 가장 적합한 컴포넌트 식별을 위해 명세 기반의 컴포넌트 저장소 모델을 제안하고 있다. 본 모델에서 제안하고 있는 컴포넌트 명세 기술(description) 항목은 컴포넌트 일반 정보, 제공되는 인터페이스들의 서비스 정보, 컴포넌트들간의 상호작용을 나타내는 결합관계 정보로 이루어져 있으며, 이 정보들은 의미적 표현이 가능한 XML의 요소(element)로 기술한다.

2.1 XML 기반의 컴포넌트 명세

XML은 사용하는 시스템에 상관없이 일관성을 유지하는 데이터 표현 형식으로 의미정보를 기술할 수 있다. 그리고 가볍고 빠르며 계층적 구조와 사용자 정의 태그를 지원함으로써 확장성과 이식성이 뛰어나다.

이에 본 연구에서는 컴포넌트 명세의 세부 정보를 의미적 표현이 가능한 XML의 요소로서 정의하고, 여기에 사용한 태그들은 요소 사이의 관계를 지정할 수 있는 DTD로 정의[6]함으로써 XML 문서의 유효성을 유지하고 있다.

이처럼 컴포넌트 명세를 구조적 형태를 갖는 XML로 정형화하여 사상(mapping) 시킴으로서 컴포넌트 명세 언어에 대한 특정 지식 없이도 원하는 컴포넌트에 대한 요구 사항 정보를 기술할 수 있게 된다. 또한 명세의 문서 구조가 논리적으로 명확하여 명세 파악의 신뢰도와 속도 향상을 가져올 수 있으며 새로운 정보의 추가 및 기존정보의 수정도 용이하다[7]. 기술된 명세는 XML 문서로 저장되므로 효율적으로 관리할 수 있고, 웹 상에서 원하는 정보를 효율적으로 검색할 수도 있게 된다.

2.2 컴포넌트의 일반적 정보

컴포넌트의 명세를 기술하는 항목 중 <일반적 정보>에는 컴포넌트의 일반적이고 대략적인 정보를

나타내기 위해 컴포넌트의 분류 정보, 이름, 기능 설명, 문서 정보, 버전, 간략한 인터페이스 정보를 포함한다. 여기에서 인터페이스 정보는 해당 컴포넌트가 제공하는 서비스의 기능 설명만을 기술한 것으로 사용자가 상세한 조립 정보를 접하기에 앞서, 컴포넌트가 제공하는 서비스의 대략적인 기능 설명만으로 사용자에게 적합한 컴포넌트를 식별할 수 있도록 단계적 효율성을 준다.

그리고 분류는 다양한 컴포넌트 분류법[8]을 지원하며, 하나의 컴포넌트 명세는 다양한 분류법을 동시에 가질 수 있다. 다음은 동시 분류법을 지원하기 위한 XML 데이터베이스 스키마를 나타낸다.

```
<?xml version="1.0" encoding="EUC-KR" ?>
<assortment>
  <classification id="cl001">
    <name>NAICS</name>
    <content>산업분류시스템</content>
    <tableName>T_NAICS</tableName>
  </classification>
  <classification id="cl002">
  </classification>
  <category name="cl001">
    <id name="ca001">
      <name>전자계산기</name>
      <parent>0</parent>
      <degree>0</degree>
      <difference>1</difference>
      <content>전자계산기에 관련된....</content>
    </category>
    <category name="cl002">
    </category>
  </assortment>
```

(그림1) 동시 분류법을 지원하는 XML DB 스키마

```
<!ELEMENT General_Info (Classification, Name,
  Content, Document?, Version?, Interface+)>
<!ELEMENT Classification (Class+)>
<!ELEMENT Class (#PCDATA)>
<!ATTLIST Class id CDATA #REQUIRED>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Content (#PCDATA)>
<!ELEMENT Document (#PCDATA)>
<!ELEMENT Version (#PCDATA)>
<!ELEMENT Interface (Name, Content, Method+)>
<!ATTLIST name CDATA #REQUIRED
  ref CDATA #REQUIRED>
<!ELEMENT Content (#PCDATA)>
<!ELEMENT Method (Name, Content)>
<!ATTLIST name CDATA #REQUIRED
  ref CDATA #REQUIRED>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Content (#PCDATA)>
```

(그림2) 명세의 일반적 정보를 기술하는 XML DTD

2.3 컴포넌트의 서비스 정보

컴포넌트의 <서비스 정보>에는 컴포넌트가 제공하는(provide) 서비스에 대한 실제 조립 정보를 나타내는 인터페이스들의 상세 정보를 기술한다. 여기에는 메소드의 리턴타입, 인자 값 정보, 속성의 데이터 타입, 이름, 초기 값, 기능 설명 등 세부적인 정보들이 포함된다.

이렇게 인터페이스에 대한 세부 정보를 나타냄으로서 실제 재사용을 위해 다른 컴포넌트와 연결할 수 있는 방법을 제공한다. 이것은 사용자 요구사항에 적합한 컴포넌트의 식별 시점에서, 구현 시 필요한 정보들을 제공함으로써 컴포넌트 기반 개발 방법론에서의 비용 절감과 활용의 효율성을 높일 수 있다.

```
<!ELEMENT Service_Info (Interface+)>
<!ELEMENT interface (Method+,Property*)>
<!ATTLIST Interface id CDATA #REQUIRED
                type CDATA #FIXED "provide">
<!ELEMENT Method (ReturnType, Parameter*)>
<!ATTLIST Method id CDATA #REQUIRED>
<!ELEMENT ReturnType (#PCDATA)>
<!ELEMENT Parameter (Name, Type)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Property (Type, Name, Initial, Content)>
<!ATTLIST Property id CDATA #REQUIRED>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Initial (#PCDATA)>
<!ELEMENT Content (#PCDATA)>
```

(그림3) 명세의 서비스 정보를 기술하는 XML DTD

2.4 컴포넌트간 결합관계 정보

컴포넌트는 다른 컴포넌트들과 결합되어 사용되어지는 경우가 대부분이며, 이런 컴포넌트들 간의 상호작용에 관해 기술한 항목이 <결합관계 정보> 이다. 여기에는 인터페이스 명세의 재사용을 지원하기 위해 상속과 같은 정보를 포함하고 컴포넌트 명세 시점에서의 결합관계를 나타내는 상호작용에 대한 정보를 포함한다.

이는 명세 관점에서 컴포넌트의 상호작용을 쉽게 파악할 수 있다는 장점을 가지며 여기서 생성되어진 XML 문서는 확장된 다양한 문서로 변환을 가능하게 한다.

```
<!ELEMENT Relation_Info (Inheritance*, Relation*)>
<!ELEMENT Inheritance (Component_Spec+)>
<!ELEMENT Component_Spec (#PCDATA)>
<!ELEMENT Relation (Component_Spec+, Interface+)>
<!ATTLIST Component_Spec id CDATA #REQUIRED
                type CDATA #FIXED "required">
<!ELEMENT Interface (#PCDATA)>
```

(그림4) 명세의 결합관계 정보를 기술하는 XML DTD

3. 컴포넌트 명세 적용 사례

본 연구에서 제안하는 컴포넌트 명세를 현재 구축 중인 컴포넌트 저장소에서 XML과 데이터베이스를 연결하는 컴포넌트의 한 부분을 예로 하여 적용해 본다. 이것은 앞서 살펴본 컴포넌트 명세의 항목별 XML DTD를 기초로 하여 작성된다. 이러한 문서는 다시 3개의 항목별로 분리하여 저장이 가능하며 각각은 독립적인 하나의 정보 단위를 이룰 수 있다.

```
<?xml version="1.0" encoding="EUC-KR" ?>
<Component_Spec id="cs0001" ?>
  <General_Info>
  </General_Info>
  <Classification>
    <class id="cf0001">cf0001</class>
    <class id="cf0002">cf0001</class>
  </Classification>
  <Names_XML_Data_Manage<Name>
    <Content>XML 기반으로 데이터베이스와의 자료처리를 수행한다</Content>
  </Content />
  <Version>beta1.2</Version>
  <Interface names="I_ProcessXML" ref="i01">
    <Content>데이터베이스에 접근하고 결과를 XML 문서로 만든다</Content>
    <Method names="DBConnect" ref="m01">
      <Content>데이터베이스에 연결된다</Content>
    </Method>
    + <Method names="makeXMLDoc" ref="m02">
    </Method>
  </Interface>
  <General_Info>
  </General_Info>
  <Service_Info>
    <Interface id="i01" type="provide">
      <Method id="m01">
        <Return_Type />
        <Parameter>
          <Name>procName</Name>
          <Type>String</Type>
        </Parameter>
      </Method>
      + <Method id="m02">
      + <Property id="p01">
      </Property>
    </Interface>
  </Service_Info>
  <Relation_Info>
  </Relation_Info>
  <Inheritance>
    <Component_Spec>cs0017</Component_Spec>
  </Inheritance>
  <Relation>
    <Component_Spec id="cs0009" type="required">
      <Interface>i03</Interface>
    </Component_Spec>
  </Relation>
  </Relation_Info>
</Component_Spec>
```

(그림5) XML 기반 컴포넌트 명세 예

'cs0001'이라는 아이디를 갖는 컴포넌트 명세는 크게 <General_Info>, <Service_Info>, <Relation_Info> 3개의 항목으로 이루어져 있다. <General_Info>에서

는 아이디 'cl001'와 'cl002'를 갖는 두 개의 분류법에 속하는 'XML_Data_Manager'라는 이름의 컴포넌트 명세임을 나타내며 컴포넌트의 기능설명, 버전 정보와 함께 'DBConnect', 'makeXMLDoc' 메소드를 가지는 'I_ProcessXML'라는 이름의 인터페이스에 대한 간략한 정보를 명시하고 있다. <Service_Info>에서는 이 컴포넌트가 제공하는 인터페이스 세부정보로 <General_Info>에서의 'I_ProcessXML' 인터페이스와 그 메소드들에, 인터페이스 아이디 'i01'와 메소드 아이디 'm01', 'm02'로 각각 사상되며 여기에 리턴타입과 인자 값 등 좀더 세부적인 정보들이 포함된다. <Relation_Info>에서는 아이디 'cs0017'인 컴포넌트 명세로부터 상속을 받으며, 아이디 'cs0009' 컴포넌트 명세에서 아이디 'i03'의 인터페이스와 결합관계가 있음을 명시하고 있다.

이러한 XML 기반의 컴포넌트 명세는 그 문서가 논리적으로 명확하여 명세의 신뢰도와 가독성을 향상시킨다.

4. 결론 및 향후연구

본 연구에서 제안한 컴포넌트 명세는 컴포넌트의 일반적이고 대략적인 정보를 나타내는 <일반적 정보>, 컴포넌트가 제공하는 서비스의 인터페이스 상세 정보를 기술한 <서비스 정보>, 컴포넌트들 간의 상호작용에 관한 <결합관계 정보>로 구성되어 있다. 이러한 3단계에 기초한 각각의 문서는 서로 독립적임과 동시에 상호 의존적이며 각각의 부분에 적합한 기술을 충분히 기술하고 있다. 예로 <일반적 정보>만으로는 컴포넌트의 서비스에 대한 정보를 습득할 수 있고, <서비스 정보>를 이용하여 사용자는 어플리케이션 및 컴포넌트 연결에 대한 정보를 습득할 수 있다. 또한 컴포넌트들간의 상호작용 정보는 <결합관계 정보>를 통해 습득할 수 있다.

이러한 명세는 XML 기반으로 기술되므로 컴포넌트 명세의 문서 구조가 논리적으로 명확하여 명세 파악의 신뢰도와 속도 향상을 가져올 수 있으며 새로운 정보의 추가 및 기본 정보의 수정도 용이하다는 장점을 가진다.

이 같은 XML 기반의 컴포넌트 명세 정보를 저장하는 본 연구의 컴포넌트 저장소는 현재 제안된 모델로의 설계와 부분적인 구현이 진행중이며, 앞으로 이 명세 정보에 기반한 검색의 효율을 극대화할 수 있는 활용 방안이 요구된다.

그리고 더 나아가 XML 구조는 다양한 문서로의

확장과 변환이 가능하다 장점을 가진다. 이에 본 논문에서 제안한 XML 구조의 문서를 UML과 같은 비교적 가독성이 높고 표준화되어 있는 소프트웨어 모델링 언어로 자동 변환하는 연구가 진행되어야 하겠다. 이는 컴포넌트 상호작용에 대한 정보를 상세히 기술할 목적으로 기존의 UML을 확장하여 표현하고자 한다.

참고문헌

- [1] Paul C. Clements, "Form Subroutines to Subsystems: Component-Based Software Development", American Programmer, 1995
- [2] Dedmond F.D'Souza, Alan C. Wills, "Object, Component and Frameworks With UML", ADDISON-WESLEY, 1998
- [3] J. Han, "An Approach to Software Component Specification", Proceedings of 1999 International Workshop on CBSE, Los Angeles, 1999
- [4] Sherif Yacoub Ammar, "A Model for Classifying Component Interface", Proceedings of 1999 International Workshop on CBSE, Los Angeles, 1999
- [5] 백경원, "컴포넌트 결합 명세서에 기반한 컴포넌트 결합 모델", 정보처리학회논문지 제8-D권 제6호, 2001.12
- [6] 정지훈, "웹 서비스", 한빛미디어, 2002
- [7] 이윤수, "재사용을 위한 XML 기반 소프트웨어 아키텍처 명세 언어", 한국정보처리학회 논문지 제7권 제3호, 2000.3
- [8] Componentsource, "http://www.componentsource.com/BuyComponents/ProductCatalog/default.asp"