한국공작기계학회 2004 추계학술대회 논문집 pp.78-85

XML과 온톨로지를 이용한 공학 설계 데이터의 상호운용성 증진에 관한 연구

정태형*, 박승현+

Improving the interoperability of mechanical design data using XML and ontology

Tae Hyong Chong*, Seung Hyun Park+

Abstract

As the complexity of engineering design environment has increased, it is difficult to exchange design data among design support systems. The purpose of this paper is to develop the XML-based Generalized Mechanical Data Exchange Formats (GMDEF) independent of specific mechanical element and improve the interoperability of them using ontology, in order to integrate diverse design data and facilitate communication between design support systems. GMDEF consists of PartDoc and AssemblyDoc. PartDoc represents the information of a single part. AssemblyDoc represents the relation of parts constituting an assembly. GMDEF is validated by GMDEF Schema. GMDEF Schema consists of separated XML Schemas and has flexible architecture to facilitate extension. We apply ontology to GMDEF Schema to share and reuse vocabularies of specific mechanical elements.

Key Words : Design Methodology(설계 방법론), XML(eXtensible Markup Language), RDF(Resource Description Framework,) Ontology

1. 서 론

일반적으로 공학 설계 환경에는 다양한 설계 데이터와 설계 시스템들이 혼재되어 있다. 이는 기계 설계 활동의 복잡성에 기인한 것으로서, 이러한 복잡성을 낮추고 설계 활동의 생산성을 높이기 위해 시스템간의 연동 및 통합에 관한 다양한 연구들이 시도되고 있다. 최근 들어 IT 기술의 발전과 함께 관련 연구들이 보다 활발하게 이루어지고 있고 그 중에서도 특히 인터넷 기반 협력 설계에 관한 연구와 PDM(Product Data Management) 및 CAD를 기반으로 하는 설계 정보의 처리에 관한 연구들이 활발하다. 이러한 분야의 연구에서 중요한 이슈 중 하나는 설계 시스템의 연동 및 통합이며 다양한 연구들이 시도되었음에도 불구하고 현재의 설계 시스템의 통합은 많은 문제점을 가지

* 정태형, 한양대학교 기계공학과 (thchong@hanyang.ac.kr)
주소: 139-791 서울시 성동구 행당동 17 한양대학교 기계공학과
+ 한양대 대학원 기계설계학과

고 있다. 첫째로, PDM 시스템을 중심으로 하는 현재의 설계 시스템 통합 환경은 CAD 기반의 시스템간의 연동만을 지원하고 있고 특화된 기계요소 설계 시스템과는 연동하기 힘들다는 것이다. 일반적으로 PDM 및 CAD 시스템은 2D 도면이나 3D의 솔리드 모델같은 CAD 데이터를 기반으로 하는 CAE 시스템 사이에서만 연동이 지원된다. 따라서 PDM 및 CAD를 기반으로 하는 환경에서는 대부분의 해석이 도면이나 솔리드 모델과 같은 CAD 데이터를 이용하여 이루어질 수밖에 없고, 사용할 수 있는 해석 기법이 FEM과 같은 매우 일반적인 기법에 한정된다.

둘째로, 설계 시스템간의 표준적인 연동방식의 부재이다. 상이한 시스템 간에 데이터를 주고받기 위해서는 표준적인 데이터 형식이 필요한데, 이를 위해 STEP(STandard for the Exchange of Product model data)[1]과 같이 도면이나 솔리드 모델에 관한 표준 데이터 형식들이 오래전부터 개발되어왔다. 그러나 이러한 표준 데이터 형식들은 대부분 CAD와 관련된 데이터로 한정되어 있어 이들을 활용할 수 있는 분야는 상당히 제한적이다. 예를 들어 수학적 해석 기법을 기반으로 하는 설계 시스템의 경우에는 실제적인 기계요소의 제원 및 파라미터와 같은 특정 분야에 특화된 정보를 필요로 하는데, 이런 경우에는 기존의 표준 데이터 형식이 없어서 자체적으로 데이터 형식을 정의하는 경우가 대부분이다. 이와 같이 자체적으로 정의한 데이터 형식이 늘어남에 따라 광범위한 설계 데이터의 교환은 갈수록 어려워지고 있다.

이에 본 연구에서는 기존의 PDM 및 CAD 시스템과 수학적 해석 기반의 중소규모 설계 시스템을 포함하는 다양한 설계 시스템 사이의 통신을 원활히 하고 상이한 설계 데이터 형식을 통합하기 위하여 XML(eXtensible Markup Language)[2] 기반의 범용 데이터 교환 형식을 개발하고, 온톨로지(ontology)[3]를 이용하여 설계 데이터의 상호운용성을 향상시켜 컴퓨터가 설계 데이터의 의미를 이해하고 자동으로 처리할 수 있도록 하기 위한 프레임웍을 구축하는 것을 목표로 한다.

2. 연구 배경

2.1 온톨로지

온톨로지는 철학분야에서 처음 도입된 용어로 원래의 의미는 우주 안에 어떤 종류의 실체들이 존재하는가에 관한 연구 또는 관심을 말한다. IT 기술에서의 온톨로지는 각 분야의 지식체계를 컴퓨터가 자동적으로 처리할 수 있는 형태로 만들기 위해 해당 분야의 용어 사이의 관계를 정의하는 일종의 사전과 같은 의미로 해석된다. 온톨로지의 기본 목적은 해당 분야의 지식과 정보를 체계적으로 정리하여 시스템간의 정보와 지식을 공유하고 재사용하는 것을 목적으로 한다. 따라서 온톨로지를 이용하여 해당 분야의 상호 합의된 어휘를 공유함으로써 인간의 개입 없이 컴퓨터가 정보와 지식의 의미를 직접 이해하고 그에 따르는 융통성을 가질 수 있다.

2.2 RDF

RDF(Resource Description Framework)[4]는 Semantic Web[5]의 핵심 기술로 웹상의 메타데이터를 기술하고 의미론적 수준의 상호운용을 지원하기 위해 W3C에서 제안한 표준 메타데이터 형식이다. RDF는 자원(resource)의 의미를 기계가 이해할 수 있는 형태로 기술하는 것을 목적으로 하고 문서의 문법과 구조보다는 자원의 설명에 중점을 두고 있다. RDF는 자원의 의미를 기술하기 위해 Fig. 1과 같이 방향성 그래프를 이용하는 형식 모델을 기반으로 한다. RDF에서 자원은 URI(Uniform Resource Identifier)를 식별자로 가지고, 이러한 자원은 다시 속성과 속성 값을 가지는 RDF Description의 집합으로 기술된다.

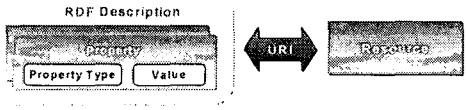

Fig. 1 Resource Description Framework Model

RDF는 기본적으로 XML 구문을 이용하는데 XML과는 달리 트리플(triple)의 집합으로 분석된다. 트리플은 자원을 나타내는 주어(subject), 속성을 나타내는 술어(predicate), 속성의 값을 나타내는 목적어(object)로 구성된다. 트리플 집합은 자원의 의미를 기계가 이해할 수 있도록 하고 지능적인 추론을 가능하게 한다.

3. XML 기반의 기계 설계 정보의 교환 형식 개발

3.1 GMDEF (Generalized Mechanical Data Exchange Format)

본 논문에서는 다양한 기계 설계 시스템의 통합을 위해 XML 기반의 범용 데이터 교환 형식을 개발하여 이를

GMDEF라고 명명하였다. GMDEF는 다음의 세 가지 목표를 달성할 수 있도록 설계되었다.

첫째, GMDEF는 기계요소에 중립적이어야 한다. 특정 기계요소에 국한되지 않고 일반적인 기계요소의 정보를 표현할 수 있도록 함으로써 단일 형식으로 다양한 기계요소의 정보를 표현할 수 있도록 한다.

둘째, 개방적인 구조여야 한다. 다양한 기계요소에 적용될 수 있어야 하기 때문에 정보의 추가 및 변경이 빈번하게 일어나게 된다. 따라서 정보의 추가나 변경이 발생해도 원래의 문서 구조가 변경되지 않으면서 확장될 수 있는 유연하고 개방적인 구조를 가져야 한다.

셋째, 다른 데이터와의 연동이 용이하고 상이한 시스템간의 통신이 쉽게 이루어질 수 있도록 해야 한다. 기존의 다른 데이터로 쉽게 변환하거나 반대로 다른 데이터로부터 GMDEF로의 변환이 쉽게 이루어질 수 있어야 한다. 또한 표준 기술을 활용함으로써 상이한 시스템간의 통신에서 데이터의 변환 및 해석에 드는 노력이 최소화되어야 한다.

이와 같은 목표를 달성하기 위해 GMDEF는 Fig. 2와 같이 조립품과 부품의 정보가 분리되어 있는 확장 가능한 구조를 취하고 있다.

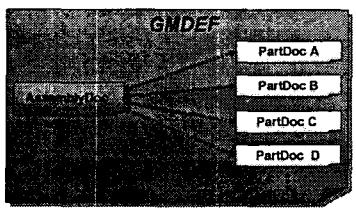

Fig. 2 Structure of GMDEF

GMDEF 문서 시스템은 PartDoc과 AssemblyDoc의 조합으로 구성되는데, PartDoc은 단일 기계요소의 정보를 기술하기 위한 문서이고 AssemblyDoc은 여러 개의 부품으로 구성된 조립품의 구성 및 연결 정보를 기술하는 문서이다. 이는 CAD 시스템에서 부품과 조립품 파일의 구조와 유사한 면이 있다. CAD 시스템에서 여러 개의 기계요소를 조합하여 하나의 완성된 조립품을 만들듯이 설계 데이터 역시 각각의 기계요소의 정보를 표현하는 PartDoc을 조합하여 AssemblyDoc을 생성하게 된다. 이러한 구조는 CAD 데이터의 구조와도 일치할 뿐만 아니라 물리적, 개념적인 조립품의 구조와도 일치하여 설계자의 이해도를 증가시키고 설계 데이터 구조의 일관성을 유지할 수 있다.

GMDEF의 문서 구조를 Fig. 2와 같이 정의한 것은 특정 기계요소에 국한되지 않는 일반적인 기계 조립품의 정보를 표현하기 위함이다. 단일 기계 부품은 일단 완성되면 그 자체가 확장되지 않는 하나의 개체로 생각할 수 있다. 한번 정해진 부품은 제원 정보 등이 바뀔 수는 있지만 새로운 정보가 추가되지는 않는다. 만약 새로운 정보가 추가된다면 그 부품은 기존에는 없었던 새로운 부품으로 간주되고 이 역시 하나의 개체로 생각될 수 있다. 따라서 하나의 부품을 단일 문서로 표현하는 것은 개념적으로도 부합할 뿐만 아니라 확장성이라는 측면에서도 문제가 되지 않는다. 조립품 문서는 부품 문서들의 정보가 직접 포함되지 않고 단지 부품 문서들의 위치와 관계 및 관리를 위한 정보만을 기술함으로써 특정 조립품에 국한되지 않는 일반성을 가진다. 따라서 부품 문서들만 존재한다면 어떠한 조립품이든지 표현이 가능하고 조립품 문서가 다른 조립품의 부품으로도 사용될 수 있어서 여러 개의 조립품으로 구성된 복잡한 조립품의 정보도 표현할 수 있다. 또한 문서를 물리적으로 분할함으로써 중복되는 정보를 최소화하여 문서의 크기를 줄이고 검색 속도를 향상시키는 효과를 얻을 수 있다.

GMDEF의 또 다른 특징은 XML과 RDF 구문을 혼합하여 사용하고 있다는 것이다. 필요에 따라서 XML과 RDF를 모두 사용함으로써 XML만 사용하는 기존의 데이터 형식에 비해서 보다 표현력이 풍부하고 서로의 단점을 보완할 수 있다.

3.1.1 PartDoc

PartDoc 문서에는 다음과 같은 정보들이 기술된다.

● 문서 자체에 대한 정보

PartDoc 자체가 메타데이터로서 활용될 수 있도록 하기 위해 PartDoc에는 문서의 제작자나 설명, 형식과 같은 문서 자체에 대한 공통 정보들이 Dublin Core element set[6]을 이용하여 doc_info 요소(element)에 기술된다. Fig. 3은 doc_info 요소의 예이다

```
<doc_info>
  <dc:creator type="machine">
    http://gearlab.hanyang.ac.kr/gear/ShaftModelers
  </dc:creator>
  <dc:date>2003-4-21 15:10</dc:date>
  <dc:format>text/xml</dc:format>
  <dc:description>The partdoc of first stage pinion
  </dc:description>
</doc_info>
```

Fig. 3 Example of doc_info element

● 부품에 대한 정보

part_info 요소에는 치수 파라미터 정보뿐만 아니라 PDM 등 다른 시스템과의 연동을 위한 정보 와 같은 특정 부품과 관련된 여러 가지 데이터가 기술된다. 이러한 정보들은 part_type 요소에 기술된다. 부품의 종류(type)에 관한 정보는 부품의 분류 방법이 일정한 것이 아니라 필요에 따라 달라질 수 있으므로 4장에서 언급할 온톨로지와의 연동을 통해 각 적용 상황에 맞도록 확장 가능한 것이 특징이다. dimensions 요소에는 부품의 형상과 관련된 정보들이 기술되는데, 기존의 방식과 달리 형상을 기술하는 문법만을 제공하여 그 문법에 따르는 어떠한 데이터도 기술할 수 있도록 하였다. 따라서 기존의 CAD에서 사용하는 기하학적인 형상정보 뿐만 아니라 각 부품에만 존재하는 치수 등도 표현할 수 있다. 또한 특정 기계요소에 국한되지 않고 다양한 데이터를 표현할 수 있다. 이 외에도 해석 및 시뮬레이션 프로그램에서 필요로 하는 파라미터 등도 표현할 수 있도록 했는데, 마찬가지로 문법만을 제공하여 다양한 파라미터를 표현할 수 있다. Fig. 4는 part_info 요소의 예로 표준 스퍼기어의 기본적인 제원을 기술하고 있다.

이와 같이 각 기계요소에 따라 달라질 수 있는 부분에 대한 정보들에 대해서 이를 기술하는 문법만을 정의함으로써 PartDoc이 특정 기계요소에 종속되지 않도록 하고 있다. 또한 이러한 정보들이 정의한 문법을 따르고 있는지 검증하기 위한 방법도 제공하고 있는데 이는 3.2 절에서 다룬다.

```
<part_info>
  <part_number>gear001</part_number>
  <part_name>pinion001</part_name>
  <part_type
    xmlns="http://gearlab.hanyang.ac.kr/GMDEF/part_type">
    <level depth="1" value="gear"/>
    <level depth="2" value="spur"/>
  </part_type>
  <dimensions standard_type="ISO">
    <gear:module gear:unit="module" value="2"/>
    <gear:number_of_teeth unit="" value="10"/>
    <gear:pressure_angel SI:unit="degree" value="25"/>
    <gear:face_width SI:unit="mm" value="20"/>
  </dimensions>
  <cost currencycodes:currencycode="KRW" value="100"/>
  <material standard_type="ISO">
    <material_type>steel</material_type>
    <material_code>SM45C</material_code>
  </material>
  <parameters>
    <gear:parameter>
      <name>role</name>
      <value>pinion</value>
    </gear:parameter>
  </parameters>
</part_info>
```

Fig. 4 Example of part_info element

● 관련 자원에 대한 정보

관련 자원에 대한 정보는 resources 요소에 기술된다. 관련 자원에는 관련 문서를 비롯하여 도면, 3D CAD 모델 등 해당 부품과 관련된 모든 것이 될 수 있다. resoueces 요소에 관련 자원에 대한 정보가 포함됨으로써 PartDoc을 사용하는 시스템이 필요에 따라 관련 정보를 가져올 수 있다. 관련 자원에 대한 정보의 기술을 위해 XML과 RDF 구문을 모두 사용할 수 있는데, RDF를 사용하는 경우에는 RDF 구문 자체만으로 관련 자원의 위치를 기술하고, XML 구문을 이용하는 경우에는 XLink[7]를 이용해서 관련 자원의 위치를 표현한다. Fig. 5는 resources 요소의 예로 Fig.4에 기술한 스퍼기어와 관련된 3D 모델의 정보를 나타내고 있다.

```
<resources>
 <resource_rdf>
   <rdf:RDF>
     <rdf:Description rdf:about=
     "http://gearlab.hanyang.ac.kr/webpub/3dmodel/pinion001.sldprt">
       <dc:title>3D Model of first stage pinion</dc:title>
       <dc:description>
         This is the Solidworks part file of first stage pinion
       </dc:description>
       <GMDEF:associated_application>
         solidworks
       </GMDEF:associated_application>
       <GMDEF:extension>sldprt</GMDEF:extension>
     </rdf:Description>
   </rdf:RDF>
 </resource_rdf>
 <resource_xlink xlink:type="simple" xlink:href=
       "http://gearlab.hanyang.ac.kr/webpub/3dmodel/pinion001.wrl">
   <title>VRML of first stage pinion</title>
   <type application_name="vrml" extension="wrl"/>
   <description>
     This is the VRML file of first stage pinion
   </description>
 </resource_xlink>
</resources>
```

Fig. 5 Example of resources element

3.1.2 AssemblyDoc

AssemblyDoc에서 조립품을 구성하는 PartDoc의 정보는 RDF 구문을 이용해서 기술된다. AssemblyDoc을 구성하는 각 PartDoc 사이의 관계는 일종의 링크(link)로 생각할 수 있는데, 링크 정보를 기술하는데 XLink를 사용하지 않고 RDF를 사용한 이유는 RDF가 자원들 사이의 연관 관계를 나타내는데 있어 보다 효율적이고 XLink보다 더 풍부한 표현력을 가짐과 동시에 더 간결한 구문을 제공하기 때문이다. 또한 RDF를 이용함으로써 각 PartDoc 사이의 관계를 시각화하는데도 용이해진다.

AssemblyDoc에서 구성 PartDoc 사이의 관계 정보는 components 요소에 mating_part라는 속성을 가지는 RDF 구문으로 기술된다. mating_part에는 해당 부품과 맞물리

는 부품들의 PartDoc 위치가 기록된다. components 요소의 정보를 이용하면 필요에 따라 해당 PartDoc 문서의 위치를 알아내어 필요한 정보를 가져올 수 있다. Fig. 6은 기어, 축, 키, 기어박스로 구성된 조립품의 연결 정보를 기술한 components 요소의 예이다. AssemblyDoc에는 정보 중에서 문서 자체에 대한 정보, 관련 자원에 대한 정보 및 조립품과 관련된 파라미터 정보도 기술되는데 이들의 표현 방법은 PartDoc과 동일하다.

3.2 GMDEF Schema

GMDEF Schema는 GMDEF 문서의 구조 및 내용을 정의하는 XML Schema[8] 기반의 Schema로, GMDEF 문서들의 유효성을 검증하고 GMDEF 문서를 활용하는 시스템 개발을 위한 지침서의 역할을 수행한다. GMDEF Schema를 이용하면 시스템이 읽어올 GMDEF 문서의 구조와 내용이 GMDEF에서 제안하고 있는 형식을 따르고 있는지 확인할 수 있다. 따라서 GMDEF 대응 시스템을 개발할 때 읽어올 GMDEF 문서의 내용을 검증하는 코드를 따로 작성할 필요가 없어지므로 시스템 개발을 보다 용이하게 할 수 있으며 GMDEF 문서로 통신하는 시스템들 사이에서 데이터의 무결성을 보장할 수 있다. 또한 GMDEF Schema는 일종의 명세서 역할도 수행하므로 GMDEF 대응 시스템을 개발할 때 참고 자료로도 활용될 수 있다.

```
<components>
  <rdf:RDF>
    <GMDEF:part
        rdf:about="file://c:/Repository/PartDoc/gear001.xml">
      <GMDEF:mating_part
        rdf:resource="file://c:/Repository/PartDoc/key001.xml"/>
      <GMDEF:mating_part
        rdf:resource="file://c:/Repository/PartDoc/shaft001.xml"/>
    </GMDEF:part>
    <GMDEF:part
        rdf:about="file://c:/Repository/PartDoc/shaft001.xml">
      <GMDEF:mating_part
        rdf:resource="file://c:/Repository/PartDoc/gear001.xml"/>
      <GMDEF:mating_part
        rdf:resource="file://c:/Repository/PartDoc/key001.xml"/>
      <GMDEF:mating_part
     rdf:resource="file://c:/Repository/AssemblyDoc/gearbox001.xml"/>
    </GMDEF:part>
    <GMDEF:part
        rdf:about="file://c:/Repository/PartDoc/key001.xml">
      <GMDEF:mating_part
        rdf:resource="file://c:/Repository/PartDoc/gear001.xml"/>
      <GMDEF:mating_part
        rdf:resource="file://c:/Repository/PartDoc/shaft001.xml"/>
    </GMDEF:part>
    <GMDEF:assembly
        rdf:about="file://c:/Repository/AssemblyDoc/gearbox001.xml">
      <GMDEF:mating_part
        rdf:resource="file://c:/Repository/PartDoc/shaft001.xml"/>
    </GMDEF:assembly>
  </rdf:RDF>
</components>
```

Fig. 6 Example of component element

GMDEF가 다양한 기계요소에 대응할 수 있도록 개발된 만큼 GMDEF Schema 역시 GMDEF의 요구사항을 모두 수용할 수 있도록 여러 개의 물리적으로 분리된 XML Schema들을 조합 및 재정의하는 구조를 취하고 있다. 이러한 구조는 개체 지향 프로그래밍(OOP)의 개념과 비슷한 방식인데, 각 기계요소에 따라 구현한 XML Schema를 조합 및 재정의함으로써 Schema의 재사용성을 높이고, 항상 일관된 문서 구조를 취할 수 있게 해준다. GMDEF Schema는 PartDoc Schema와 AssemblyDoc Schema로 구성된다. Fig. 7은 PartDoc Schema의 구조를 보여준다.

PartDoc Base Schema는 Fig. 8과 같이 PartDoc 문서의 전체적인 구조를 정의하고 있다. PartDoc Base Schema에는 doc_info나 resources와 같은 공통 정보 및 문서 전체에 대한 구조를 정의하고 있는 PartDoc의 근간이 되는 Schema로 PartDoc Base Schema만으로도 PartDoc의 기본적인 유효성 검사가 가능하다. 각 기계요소마다 달라질 수 있는 부분인 dimensions나 parameters 요소는 any 형식으로 처리하고 있는데, 이는 이 요소들을 확장할 수 있도록 하기 위함이다. OOP에 비유하자면 PartDoc Base Schema는 부모 클래스(parent class)의 역할을 수행하고, 각 기계요소에 따른 PartDoc Instance Schema는 PartDoc Base Schema를 상속받는 자식 클래스(child class)가 된다.

각 기계요소마다 달라질 수 있는 dimensions와 parameters 요소의 내용은 외부 Schema인 Dimension Schema와 Parameter Schema를 이용해서 정의하고 있다. Dimension Schema는 각 기계요소에 대한 일종의 용어 사전 또는 어휘집과 같은 역할을 수행하며, 각 기계요소별로 존재할 수 있다. Dimension Schema에는 각 기계요소의 형

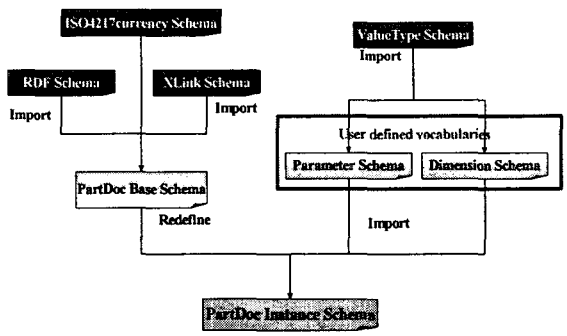

Fig. 7 Structure of PartDoc Schema

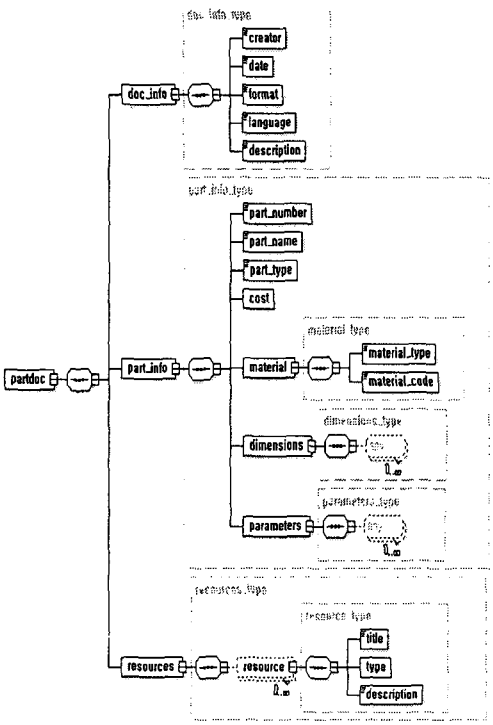

Fig. 8 Structure of PartDoc Base Schema

상과 관련된 용어를 XML 요소로 정의한다. 이때 정의한 용어가 가질 수 있는 값들은 ValueType Schema에 정의되어 있는데, XML Schema에서 사용할 수 있는 모든 형식을 지원하므로 기계 설계 시스템을 개발하는데 필요한 거의 모든 프로그래밍 데이터 형식을 사용할 수 있다. ValueType Schema는 여러 Schema 사이에서 공유되도록 하여 데이터 형식을 일관되게 정의 및 관리할 수 있도록 하였다. Parameter Schema는 기계요소의 해석에 필요한 여러 가지 파라미터를 기술하는 형식을 정의하고 있으며 Dimensions Schema와 마찬가지로 VaulType Schema를 이용하여 파라미터의 데이터 형식을 정의하고 있다.

이와 같이 PartDoc Schema는 PartDoc Base Schema를 기본으로 하여 각 기계요소마다 달라질 수 있는 부분들을 사용자 정의 Schema로 분리하고, 이들을 가져오기(import) 함으로써 최종적인 해당 기계요소의 PartDoc Instance Schema를 구성하게 된다. 이와 같은 PartDoc Schema의 구조는 Schema의 재사용성을 극대화할 뿐만 아니라 다양한 조합을 통해 광범위한 확장성 또한 획득하게 된다. 예를 들어서 PartDoc에 여러 가지 기계요소의 정보를 섞어서 기술해야 하는 경우에도 각 기계요소들의 Dimension

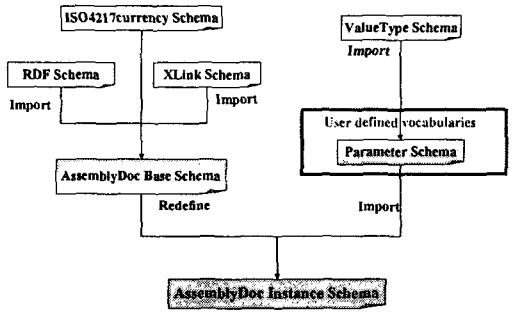

Fig. 9 Structure of AssemblyDoc Schema

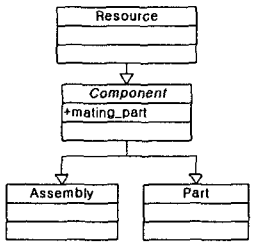

Fig. 10 UML diagram of RDF Schema for AssemblyDoc

Schema들을 PartDoc Instance Schema에서 참조하게 함으로써 간단하게 구현할 수 있다.

해당 기계요소에 대한 PartDoc Instance Schema의 작성은 매우 간단하여 적용하려는 기계요소의 용어사전 역할을 수행하는 Schema들을 참조하여 필요한 용어들의 요소를 나열하기만 하면 된다. 대부분의 내용들이 재사용되는 Schema에 들어있기 때문에 실제 Instance Schema의 내용은 매우 간단해지는 장점을 가진다.

Fig. 9는 AssmeblyDoc Schema의 구조를 보여준다. AssemblyDoc Schema 역시 PartDoc Schema와 마찬가지로 Base Schema를 근간으로 확장하는 구조를 취하고 있다. AssemblyDoc Schema에서 각 기계요소별로 달라질 수 있는 부분인 조립품의 파라미터 정보를 나타내는 Parameter Schema는 PartDoc Schema와 마찬가지로 외부 XML Schema를 참조하는 형식을 취하고 있어서 다양한 데이터를 표현할 수 있다.

AssemblyDoc에서 구성 PartDoc 사이의 연결 관계를 나타내기 위해 RDF 구문의 검증을 위한 RDF Schema를 개발하였다. Fig. 10은 RDF Schema의 UML(Unified Modeling Language)[9] 다이어그램을 보여준다. Fig. 10과 같이 RDF Schema에는 조립품과 부품의 부모 클래스인 component 클래스를 정의하고 component 클래스의

mating_part 속성에 연결 요소를 정의하고 있다. 따라서 부품간의 조합뿐만 아니라 하부 조립품들의 조합도 가능해져서 복잡한 조립품의 구성 요소들의 관계도 정의할 수 있다.

4. 온톨로지를 이용한 GMDEF의 상호운용성 증진

GMDEF Schema는 XML 기반의 뛰어난 상호운용성을 확보하고 있지만, GMDEF만으로는 내부에 기술된 정보의 의미까지 유지하는 의미론적 수준의 상호운용성을 갖지 못한다. 이에 본 논문에서는 대규모의 설계 데이터의 상호운용성 증진을 위해 온톨로지와의 연동을 통해 컴퓨터가 GMDEF 문서에 기술된 정보의 의미까지 이해할 수 있도록 GMDEF와 온톨로지의 연동에 대한 연구를 수행하였다.

GMDEF Schema는 다양한 기계요소 및 조립품의 정보를 동일한 구조로 표현할 수 있도록 할 뿐만 아니라 각 기계요소의 어휘를 미리 정해놓고 이를 참조하여 사용기 때문에 동일한 어휘의 사용을 보장하고, 따라서 GMDEF 문서를 교환하는 시스템간의 상호운용성을 증진시킨다. 그러나 기계요소의 어휘를 포함하는 XML Schema는 어휘의 문법 또는 철자를 정의할 뿐 그 의미까지 정의하지는 못하므로 모호성이 발생할 수 있다. 따라서 GMDEF 문서를 해석하는 시스템마다 다른 의미로 이해하거나 아예 분석할 수 없는 상황이 발생할 수도 있다. 이는 XML 기술의 상호운용성의 한계로 인하여 발생하는 문제이다. XML에서는 대소문자 구분이나 띄어쓰기 및 공백 처리 방식에 따라서 시스템마다 하나의 정보가 다른 방식으로 해석될 수 있다. 예를 들면 XML 문서는 대소문자를 구분하므로 "Length"와 "length"는 다른 정보로 해석될 수 있다. 또한 같은 의미라 할지라도 비슷한 뜻의 다른 용어로 기술될 수도 있다. 일반적인 스퍼 기어에서 기어 이의 크기를 나타내는 단위인 module은 헬리컬 기어에서의 traverse module과 구분하기 위하여 normal module로 사용되기도 하는데 두 용어의 의미는 스퍼 기어에서는 동일하다. 또한 같은 의미의 용어라 할지라도 적용대상이 다른 경우도 있는데, 축의 길이(length)와 키의 길이(length)는 같은 용어와 같은 뜻을 나타내지만 적용 대상이 축과 키로서 서로 다르다. 통신하는 시스템 사이에 용어의 정의 및 표현 방식에 대한 상호합의가 존재하지 않으면 이와 같은 모호성이 발생할 수밖에 없다. 온톨로지는 이러한 모호성을 제거하고 GMDEF에 정의된 제원 및 파라미터의 의미를 컴퓨터에게 정확하게 알려주는 역할을 수행한다.

본 논문에서 구축한 온톨로지에는 기계요소의 분류 및 계층 구조를 정의하고 관련 용어들을 해당 객체에 속성으로 추가된다. 이 온톨로지는 해당 기계요소 분야의 전문가의 지식이 종합되어 일종의 용어 사전과 같은 역할을 수행한다. 온톨로지 구축을 위한 온톨로지 편집기로 스탠포드 대학에서 개발한 Protégé 2000[10]를 사용하였고, 온톨로지 저장 형식으로는 RDF Schema와 RDF를 이용하였다.

이렇게 구축한 온톨로지를 이용하여 Fig. 11과 같이 GMDEF와 온톨로지의 연동방법을 개발하였다. 기계요소의 분류에 대한 정보는 GMDEF Schema에서 온톨로지의 계층구조를 직접 참조하는 방식으로 이루어진다. GMDEF Schema는 온톨로지에 접근하여 기계요소의 객체의 계층구조에 따라 기계요소 분류를 위한 ID를 생성한다. 이 ID는 온톨로지에 접근하는 모든 시스템에서 같은 의미로 받아들여지므로 컴퓨터가 해당 기계요소의 종류를 정확하게 알 수 있다. 기계요소의 용어는 GMDEF Schema에서 온톨로지에 정의된 용어를 가져와서 XML Schema 문법으로 변환하는 방식을 취한다. 즉 GMDEF Schema에서 온톨로지에 정의된 용어를 참조하여 GMDEF 문서의 용어를 검증하기 위한 XML Schema 구문을 자동으로 생성한다. 이렇게 생성된 GMDEF Schema를 이용하여 GMDEF 문서의 유효성을 검증하게 된다.

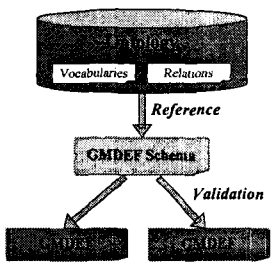

Fig. 11 Relation between Ontology and GMDEF Schema

온톨로지를 최상위 어휘집으로 이용하고 여기에서 자동으로 GMDEF 스키마의 어휘집 스키마를 생성하게 함으로써 대규모 단위에서의 어휘 공유를 실현할 수 있다. 또한 기계요소의 분류의 경우에도 온톨로지에 정의된 기계요소의 계층에 따른 URI를 기계요소의 형식으로 정의하면 유일무이한 기계요소의 형식을 정의할 수 있을 뿐만 아니라 다양한 기계 분류법을 사용할 수 있어서 특정 분류법에 제한되지 않는 개방적인 형식 판별이 가능해진다.

5. 활용 방안

GMDEF와 GMDEF Schema는 다양한 기계요소의 설계 데이터의 기술 및 연동에 적용될 수 있다. GMDEF의 물리적으로 분할된 문서 구조로 인하여 정보의 중복이 최소화되고 확장이 용이해질 뿐만 아니라 XML과 RDF 같은 표준 기술을 기반으로 함으로써 기존의 데이터와의 연동 및 상호 변환이 매우 용이하다. GMDEF 문서는 XSLT(XSL Transformations)[11]와 같은 기술을 통해 기존의 XML 데이터와의 상호 변환이 용이하다. 또한 데이터베이스로부터 GMDEF 문서를 생성하는 방식을 통해 데이터베이스와도 연동 또한 쉽게 구현할 수 있다. Fig. 12는 시스템 통합을 위한 중간자로서의 GMDEF의 역할을 보여준다. 다양한 시스템이 데이터를 주고받아야 할 경우 GMDEF를 이용함으로써 교환 데이터를 하나로 단일화할 수 있다. 특히 다른 분야에 속하는 시스템, 다른 정보를 필요로 하는 시스템들이 혼재되어 있는 경우에 GMDEF의 장점이 더욱 부각된다. GMDEF는 범용적인 설계 데이터를 기술할 수 있으므로 여

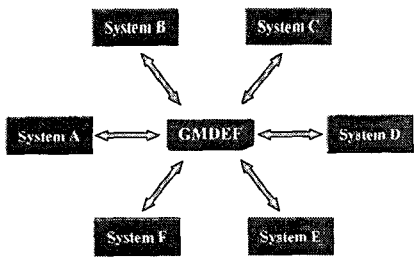

Fig. 12 Role of GMDEF as a mediator

러 가지 설계 데이터가 혼재되어 있는 환경에서 GMDEF를 표준 교환 형식으로 이용하면 데이터의 변환에 필요한 노력이 최소화되고 시스템간의 통신이 용이해진다. GMDEF를 지원하지 않는 기존의 시스템이 존재할 경우에는 GMDEF로의 데이터 변환을 통해 GMDEF 기반의 통합 설계 시스템 프레임웍에 참여할 수 있다.

6. 결론

본 논문에서는 다양한 설계 데이터의 상호운용성 증진을 위해 높은 확장성과 표현력을 가지는 GMDEF 문서 시스템을 개발하였다. GMDEF는 설계 데이터를 조립품과 부품으로 나누어 기술하도록 함으로써 다양한 조립품에 적용 가능하며, 각 기계요소마다 달라질 수 있는 데이터를 기술하는 형식을 제공함으로써 특정 기계요소에 종속적이지 않는 특징을 확보하였다. 또한 보다 높은 수준의 상호운용성을 획득하기 위해 온토롤지와 XML Schema를 연동한 GMDEF Schema를 개발하였다. 특히 GMDEF Schema는 여러 개의 물리적으로 분리된 XML Schema를 참조 및 조합함으로써 GMDEF 문서의 무결성을 검증할 수 있으며, 각 XML Schema의 재활용성을 극대화함으로써 각 기계요소에 적용하기 쉽도록 하였다.

본 논문에서 개발한 GMDEF를 이용하면 설계 시스템간의 데이터 교환 및 연동이 한가지로 통일되므로 연동에 드는 비용과 시간을 감소시킬 수 있으며, 대규모의 통합 설계 프레임웍 구축을 보다 용이하게 할 수 있다.

후기

"이 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2003-041-D00089)."

참 고 문 헌

(1) ISO, 1992, "Product Data Representation and Exchange-Part1: Overview and Fundamental Principles," ISO Documentation, ISO DIS 10303-1

(2) XML, Extensible Markup Language, http://www.w3c.org/XML

(3) Guarino, N., 1998, "Formal Ontology in Information System," ISO Press

(4) RDF, Resource Description Language, http://www.w3c.org/RDF

(5) Decker et al., 2000, "The Semantic Web: Rot of XML and RDF," IEEE Internet Computing,

(6) Dublin Core, http://dublincore.org

(7) Pinnock, Jon and Hunter, David, 2000, Beginning XML, Wrox Press

(8) Jon Duckett, et al, 2001, Professional XML Schemas, Wrox Press

(9) Sturm, Jake, 1999, Professional VB UML, Wrox Press

(10) Protégé 2000, http://protege.stanford.edu.

(11) XSLT, XSL Transformations, http://www.w3c.org/TR/xslt.