

메타데이터를 이용한 분산 STEP 데이터의 웹 인터페이스

진연권*, 유상봉**

Web Interface for Distributed STEP Data using Metadata

Jin, Y. K.* and Yoo, S. B.**

ABSTRACT

Even though we have greater chances to accomplish successful collaborative design by utilizing recent proliferation of networks, current practices do not fully take advantage of the information infrastructure. There are so much data over the networks, but not enough knowledge about the data is available to users. The main objectives of the product data interface system proposed in this paper are to capture more knowledge on managing product data and to provide users effective search capability. We define the metadata model for product data defined in STEP AP 203 and manage the metadata from product data in a repository system. Because we utilize the standard formats such as STEP for product data and RDF for metadata, the proposed approach can be applied to various fields of industries independently on commercial products.

Key words : Product Data, STEP, Knowledge Management, Metadata, RDF, Web Interface

1. 서론

제품의 전 생명 주기 동안에 발생하는 제품데이터들에 대한 원활한 교환과 접근은 기업의 생산성을 향상시키는데 중요한 요소이다^[1,2]. 하지만 제품과 관련된 데이터들은 지리적으로 분리된 여러 부서나 조직에서 생성되고 있으며, 동일한 제품에 대해 서로 이질적인 데이터 포맷을 가지고 있어 제품데이터를 공유하고 교환하는데 어려움이 따른다. 따라서 이기종의 분산된 환경에서 생성되고 있는 제품데이터에 대한 접근을 위해서는 상이한 시스템들에 대한 통합과 동일 인터페이스를 통한 시스템에 대한 접근이 요구된다.

본 연구의 목적은 제품데이터의 효율적인 검색과 접근을 위해 제품데이터의 표준 포맷인 STEP과 메타데이터 정의 표준인 RDF를 이용함으로써 시스템 사이의 호환성을 제공하고 웹 인터페이스를 통한 분산 제품데이터의 용이한 검색과 접근을 제공하는데

있다. 분산 환경 지원을 위한 CORBA와 JAVA를 이용한 웹 기반의 인터페이스는 사용자들이 웹 브라우저가 설치되어 있는 곳이면 어디에서라도 손쉽게 제품데이터를 검색하고 제품에 관한 정보 수집을 가능하게 한다.

메타데이터의 사용은 특정 사용자나 소프트웨어 도구에서 빠르고 정확하게 데이터를 검색하고 접근하기 위한 방법을 제공한다. 자동차나 항공기와 같이 복잡한 제품데이터를 다루는 경우에는 원하는 데이터에 대한 정보를 얻는 것이 쉽지 않기 때문에 제품의 특징이나 구성 정보에 바탕을 둔 메타데이터의 사용이 요구된다^[3]. 하지만 각 시스템들마다 메타데이터의 정의 방법이나 저장 포맷이 달라 데이터를 교환하고 공유하는데 어려움이 있다. 이렇게 제각기 구축되어 있는 레포지토리 간의 통합을 위해서는 메타데이터의 정의와 저장 및 교환을 위한 표준이 필요하며, 이를 위해 표준적인 메타데이터 정의 메커니즘을 제공하는 RDF(Resource Description Framework)를 이용할 수 있다. RDF^[4,5]는 메타데이터 간의 상호 운용성 제공을 목적으로 W3C에서 제정된 표준으로, 메타데이터의 정의와 저장 및 교환을 위한 표준적인

*인하대학교 자동화공학과

**중신회원, 인하대학교 자동화공학과

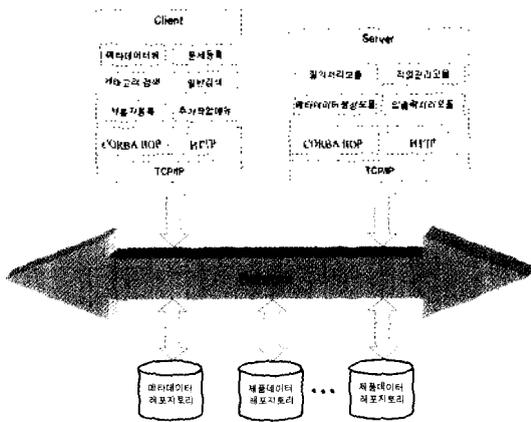


Fig. 1. Overall system architecture.

메카니즘을 제공한다. 특히 메타데이터의 저장과 교환을 위한 포맷으로 XML 신택스를 이용함으로써 웹을 통해 쉽게 접근할 수 있고 서로 다른 시스템간의 정보 교환 시에 표준적인 데이터 포맷을 제공할 수 있다.

본 연구에서 구현한 전체 시스템의 구성은 Fig. 1과 같으며, 각 모듈의 기능은 다음과 같다.

1) 서버 모듈

- 질의처리모듈: 클라이언트가 요청한 XML 포맷의 질의어를 파싱하여 질의어를 추출한 후 데이터베이스에 저장된 데이터를 검색하기 위한 SQL 문을 생성한다.
- 메타데이터생성모듈: 클라이언트 측에서의 문서등록 요청을 받아 STEP 파일의 위치를 확인하고 STEP 파일에 포함되어 있는 제품구성정보와 제품구조정보등을 추출한 후 RDF 메타데이터를 생성한다.
- 작업관리모듈: BOM정보, 등록정보, 메타데이터 정보 요구나 제품데이터 형상 뷰, 텍스트 뷰등 클라이언트의 추가 작업에 대한 요청을 처리한다.
- 입출력처리모듈: 클라이언트와의 CORBA 통신과 STEP 파일의 입출력을 처리한다.

2) 클라이언트 작업 메뉴

- 메타데이터 뷰: RDF로 표현된 메타데이터(즉 RDF 스키마)를 트리 형식으로 보여주는 인터페이스이다.
- 문서등록: STEP 파일을 등록하기 위한 인터페이스로 등록자 ID와 패스워드를 통해 등록자 확인을 거치게 된다.
- 카테고리 검색: 특정 메타데이터 요소에 대한 값

을 입력하여 검색을 수행한다.

- 일반검색: 도면명, 제품명, 작성자, 회사명 등의 기본적인 키워드를 가지고 검색을 수행한다.
- 사용자등록: 사용자가 등록요청을 하면 그 요청을 받아 사용자 정보를 시스템에 등록시킨다.
- 추가작업메뉴: 반환된 결과에 대해 클라이언트는 추가적인 작업을 수행할 수가 있는데, 이러한 추가 작업으로는 BOM정보 요청, 문서등록정보 요청, 해당 문서의 메타데이터정보 요청, STEP 파일에 대한 형상 뷰, 텍스트 뷰 등이 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 설명하고, 3장에서는 제품데이터를 기술하기 위한 메타데이터 요소들과 그것들과 관련된 STEP 엔티티들을 설명하고, RDF를 이용하여 메타데이터를 정의하고 표현하기 위한 방법을 기술한다. 4장에서는 메타데이터의 생성, 질의처리, 클라이언트 메뉴 등에 관한 시스템 구현 사항에 대해 설명한다. 5장에서는 구현된 시스템의 실행 예와 시스템에 대한 평가를 하고 6장에서 향후연구과제 제시와 결론을 맺는다.

2. 관련연구

제품데이터와 그 제품데이터에 관련된 데이터의 통합과 공유를 위한 연구는 주로 STEP과 SGML의 연계 방안을 중심으로 진행되고 있으며, ISO TC184/SC4/WG10에서 STEP/SGML 연계를 공식적으로 진행하고 있다. 최근에는 차세대 웹 문서의 표준으로 떠오르고 있는 XML과 STEP과의 연계 방안에 대한 연구도 활발히 진행되고 있다. XML은 SGML의 부분집합으로서 SGML에 비해 간단하면서도 구조화된 문서 표현에 적합하고 웹을 통해 쉽게 공유가 가능하다는 장점이 있다. Daniel Rivers-Moore는 STEP EXPRESS 스키마에 대해 가능한 XML DTD의 표현을 제시하였고⁶⁾, Joint Electronic Commerce Program Office(JECPO) 지원하의 Product Data Interoperability(PDI) 프로젝트에서는 PDML이라는 XML DTD를 만들었다. PDML은 상업용 PDM 시스템들이나 JEDMICS와 같은 정부 시스템들 사이의 제품 정보 교환을 위해 디자인된 XML DTD로 인터넷, STEP, XML을 이용하여 제품 데이터 교환을 위한 패러다임을 제공한다⁷⁾.

메타데이터에 대한 연구는 여러 조직에서 각기 자신들의 응용 분야에 따라 연구되고 있다. 이러한 조

직들은 그 조직에 맞는 메타데이터 표준을 정의하고 있는데 대표적인 메타데이터 표준으로는 다음과 같은 것들이 있다.

- Government Information Locator Service(GILS) FIPS-192: 정부의 정보 리소스를 기술하기 위한 표준.
- Federal Data Geographic Committee(FGDC): 지리 정보 리소스를 기술하기 위한 표준.
- Machine Readable Card(MARC): 도서관 카타로그 레코드의 요소를 정의한 표준.
- Dublin Core(DC): 웹 리소스와 연관된 메타데이터를 정의한 표준.
- Consortium for the Interchange of Museum Information(CIMI): 박물관 정보와 연관된 메타데이터를 정의한 표준.

이렇게 상이한 메타데이터 사이의 상호운용성을 제공하기 위해 W3C에서는 RDF라는 표준을 제정하였다. RDF는 특정 응용에 종속적이지 않는 메타데이터 정의와 표현을 위한 메커니즘을 제공하고, XML 신택스를 이용하여 웹 기반의 다양한 메타데이터를 통합한다^[4,5]. RDF에 대한 연구로는 HTML 문서에서 분류자(Classifier)를 이용해 정보를 추출하고 자동으로 RDF 메타데이터를 생성하는 연구가 있었으며^[6], SGML/XML 문서를 효과적으로 검색하고 관리하기 위해 RDF 메타데이터를 통하여 의미 있는 단위로 SGML/XML 문서를 축소하고 이 축소된 정보를 기반으로 인덱스를 만들어 내용중복을 방지하는 방법을 제시한 연구가 있었다^[9].

현재 여러 업체에서 RDF를 지원하고 있는데, 대표적으로 Linux Packages Database와 넷스케이프사의 모질라가 있다. Linux Package Database에서는 Linux에서 이용가능한 RPM 패키지에 대한 정보를 추출하여 RDF 메타데이터를 생성한다^[10]. 이렇게 생성된 메타데이터는 패키지 설치시 필요한 패키지의 위치나 패키지 사이의 의존관계 등의 정보를 얻기 위해 이용된다. 넷스케이프의 모질라 프로젝트에서는 RDF 기반의 Aurora라고 사용자 인터페이스 위젯(UI widget)을 제공하는데 이는 사용자들이 적은 시간과 노력으로 유용한 정보를 찾을 수 있도록 도움을 준다^[11].

본 연구에서는 동시공학 및 기업통합 등의 기반이 되는 제품데이터의 웹 인터페이스를 통한 검색 및 공유를 지원하기 위하여, 제품데이터의 메타데이터를 정의하였고 메타데이터 레포지토리와 사용자 인터페이스를 구현하였다. 이러한 메타데이터 관리 시

스템은 제품데이터 외에 기업의 인사, 판매, 자원 등을 포함함으로써 전사적인 지식관리 시스템으로 발전할 수 있다.

3. 제품데이터의 메타데이터

하나의 제품은 여러 개의 어셈블리 부품으로 구성될 수 있으며 각 부품마다 그 자신의 도면과 관련문서를 가지게 된다. 제품에 대한 메타데이터는 제품 정보의 검색과 관리를 위한 데이터로 STEP AP203 파일에서 추출할 수 있는 정보를 바탕으로 생성되며, 이 정보들은 STEP 파일의 헤더 정보와 CC1에 해당하는 구성관리정보를 중심으로 추출이 된다. 이렇게 정의된 메타데이터는 RDF(Resource Description Framework)로 표현이 된다.

3.1 메타데이터의 요소정의

Fig. 2는 시스템의 요구사항을 표현한 유스케이스 다이어그램이다. 시스템의 사용자라 할 수 있는 액터로는 관리자(Administrator)와 사용자(User)가 있다. 관리자는 사용자가 등록요청을 하면 그 요청을 받아 사용자 정보를 시스템에 등록시킨다. 관리자는 사용자의 역할도 수행할 수 있는데, 이 관계는 스테레오타입(Stereotype) Extends로 표현될 수 있다. 사용자가 시스템에 접속하여 수행하는 작업으로는 크게 문서등록과 문서검색 그리고 검색 결과에 대한 작업 선택 등이 있다. 사용자가 문서등록을 요청하면 먼저 사용자 등록여부를 검사한 후 문서등록이 이루어진다. 검색에는 일반검색과 카테고리 검색이 있으며, 일반검색은 특정 키워드와 일치되는 문서를 검색하

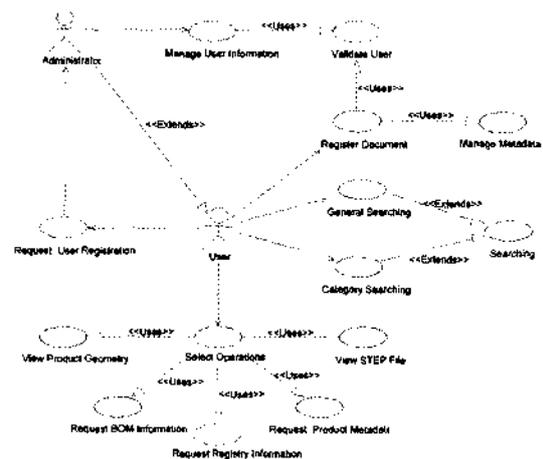


Fig. 2. Use case diagram of metadata system.

게 되고, 카테고리 검색은 하나 이상의 검색 필드에 대한 값을 입력하여 보다 정교한 검색을 수행하게 된다. 검색된 결과에 대해 사용자는 문서와 관련된 BOM정보, 등록정보, 메타데이터정보, 제품데이터에 대해 텍스트 형태의 뷰나 형상 뷰를 요청할 수 있다.

메타데이터의 요소는 STEP AP203 파일 자체에 대한 정보와 파일 내에 포함되어 있는 부품과 관련된 정보를 중심으로 정의된다. STEP 파일에 대한 정보는 파일의 헤더에 정의된 정보와 동적으로 생성되

는 파일크기, 파일위치 등이며, 부품과 관련된 정보는 STEP AP203의 구성관리, 제품구조, 규격, 기하학적 형상정보 중에서 구성관리와 제품구조정보를 바탕으로 Table 1과 같이 메타데이터를 정의하였다.

부품정보는 product 엔티티, 문서정보는 document 엔티티, 구성원 정보는 person 엔티티, 승인정보는 approval 엔티티를 중심으로 구성되며 이들 엔티티 간의 관계는 다음과 같은 엔티티를 이용하여 정의된다.

- person_and_organization_role: 구성원과 구성원의

Table 1. Element definition of metadata

항 목	요소 이름	요소 표시자	요소 설명
디자인정보 (Design)	파일이름	fileName	파일의 이름
	파일설명	fileDesc	파일의 설명
	전처리기명	preprocessor	파일을 생성한전처리기
	스키마명	schema	AP스키마의 이름
	파일위치	url	파일의 URL
	파일크기	fileSize	파일의 크기(단위는KB)
	등록정보	registryinfo	디자인의 등록정보
	bom정보	bominfo	디자인의 BOM 정보
	포함된부품	containedpart	디자인에 포함된 부품
등록정보 (Registry)	등록자식별자	registrarID	등록한 사람의 식별자
	등록자명	registrarName	등록한 사람의 이름
	등록자이메일	registrarEmail	등록한 사람의 이메일
	등록일자	registryDate	등록한 일자
BOM정보	부품레벨	level	어셈블리에서의 부품의 레벨
	partID	부품의 식별자	부품 식별자
	부품이름	partName	부품의 이름
	부품수량	quantity	부품의 수량
하위부품정보	subBOM	하위부품에 관한 정보	
부품정보 (Part)	부품식별자	partID	부품의 식별자
	부품이름	partName	부품의 이름
	부품설명	partDesc	부품에 대한 설명
	관련문서	associatedDoc	부품과 관련된 문서정보
	관련된구성원	associatedPerson	부품과 관련된 구성원 정보
	관련승인정보	approvalInfo	부품의 승인정보
문서정보 (Document)	문서식별자	docID	문서의 식별자
	문서이름	docName	문서의 이름
	문서설명	docDesc	문서에 대한 설명
	문서종류	docKind	문서의 종류
구성원정보 (Person)	구성원식별자	personID	구성원의 식별자
	구성원이름	personName	구성원의 이름
	구성원조직명	employedBy	구성원이 속한 조직명
	구성원역할	personRole	부품에 대한 구성원의 역할
승인 정보 (Approval)	승인상태	approvalStatus	현재 부품에 대한 승인 상태
	승인자	approvedBy	승인자의 정보
	승인내용	approvalRole	승인의 역할(종류)
	승인날짜	approvalDate	승인된 날짜

소속, 그리고 구성원의 역할을 연결한다.

- next_assembly_usage_occurrence: 제품구조를 정의하는 엔티티로 상위 부품과 하위 부품에 대한 참조를 가지고 있다.
- shape_definition_representation: 부품에 관련된 형상데이터를 연결한다.
- cc_design_person_and_organization: 부품과 관련된 구성원을 연결한다.
- cc_design_approval: 부품과 관련된 승인정보를 연결한다.
- approval_person_organization: 승인자와 승인자의 소속, 승인 역할 정보를 연결한다.
- approval_date_time: 승인과 승인된 날짜 정보를 연결한다.

부품의 수량정보와 레벨 정보를 제외하고는 해당 엔티티의 인스턴스나 참조되는 엔티티의 인스턴스를 찾아 1:1 매핑을 통해 정보를 추출할 수 있다. 수량 정보와 레벨 정보는 해당 부품을 참조하는 next_assembly_usage_occurrence의 엔티티의 개수를 파악하여 추출한다.

3.2 RDF를 이용한 메타데이터의 정의

RDF는 메타데이터를 처리하기 위한 기본 프레임워크로 애플리케이션 사이의 상호운용성 제공을 주된 목적으로 하고 있다^[45]. 즉 특정 애플리케이션 도메인의 리소스만을 기술하기 위한 것이 아니라 도메인에 중립적이면서 어떠한 도메인의 정보도 묘사할 수 있는 메카니즘을 제공한다. 현재 W3C에서는 RDF와 관련하여 두 개의 명세서를 발표하였다. 이는 Resource Description Framework(RDF) Model and Syntax Specification과 Resource Description Framework(RDF) Schema Specification으로 현재 각각 Recommendation 단계와 Proposed Recommendation 단계에 있다.

RDF Model and Syntax Specification에서는 메타데이터를 표현하기 위한 모델과 그 모델의 저장과 교환을 위한 인코딩 신택스를 정의하고 있다. 모델이라는 것은 메타데이터의 사용과 정의를 위한 추상적이고 개념적인 프레임워크를 말하며 신택스는 이 모델을 저장하고 교환하기 위한 목적으로 인간과 기계가 읽을 수 있는 형태로, 물리적으로 인코딩되는 포맷을 말한다.

RDF를 이용한 메타데이터 정의는 다음과 같은 단계로 구분할 수가 있다.

1. RDF 스키마 모델링.

2. RDF 데이터 모델과 XML 신택스를 이용한 인코딩.

메타데이터 요소에 대한 의미와 관계를 정의한다는 것은 RDF Schema Specification에서 정의되어 있는 리소스들(클래스와 특성들)을 이용하여 RDF 스키마를 작성하는 것이다. 미리 정의된 리소스는 rdfs와 rdf라는 이름공간(namespace)을 갖는다. 이름공간이란 XML 문서에서 요소나 특성들의 이름의 일부로, URI로 식별되는 이름들의 집합이다^[12]. 이름공간이 URI로 식별된다는 것은 한 이름공간 안에 포함된 개별적인 이름들이 모두 고유한 이름이라는 뜻으로, 서로 다른 스키마(혹은 DTD)를 혼용해서 사용하더라도 같은 이름으로 인한 혼돈이 야기되지 않는다. rdfs는 RDF Schema Specification에서 정의된 리소스에 대한 접두어로 이용되며 rdf는 RDF Model and Syntax Specification에서 정의된 리소스에 대한 접두어로 이용된다.

Fig. 3은 Table 1에서 정의한 항목과 요소들을 바탕으로 정의한 RDF 스키마이다. Class 클래스의 인스턴스로는 Design, Registry, BOM, Part, Approval, Document, Person이 있다. 새로운 클래스를 나타내는 이 리소스들을 이용하여 실제 STEP 파일과 그 파일에 포함되어 있는 부품정보, 부품구조정보, 승인정보, 관련문서정보, 관련자정보 등을 기술하게 된다. 이러한 인스턴스들은 Class 클래스의 인스턴스 즉 rdf:type 값으로 rdfs:Class를 취하고, rdfs:Resource 클래스의 서브클래스가 된다. 본 연구에서 궁극적으로 기술하고자 하는 대상은 Design(도면)이므로 모든 클래스들이 Design 클래스와 직간접적으로 연결되어 있다. 보통의 타원은 특성을 나타내는 리소스들로 Property 클래스의 인스턴스에 해당한다. rdfs:subClassOf는 상위 클래스에 대한 서브클래스를 나타내고, rdfs:domain은 해당 특성으로 기술되는 리소스를 나타내며, rdfs:range는 특성이 취하는 값을 나타낸다. 모든 리소스는 rdf:type 속성을 가져야 하는데 Fig. 3에서 rdf:type에 대한 표현은 타원에 의해 표현되고 있다. Fig. 3에서 사용된 rdfs:Literal 클래스는 문자와 숫자로 이루어진 객체를 나타낸다. 이렇게 모델링 된 RDF 스키마는 XML 신택스를 이용해 인코딩된다.

특정 도메인에 대한 메타데이터의 기술은 RDF 스키마에서 정의된 요소들에 기반하여 RDF 데이터 모델을 정의하는 것이다. 이렇게 정의된 RDF 데이터 모델은 XML 인코딩을 통해서 실제 애플리케이션에서 처리가 된다. XML 인코딩을 위한 신택스에는 시

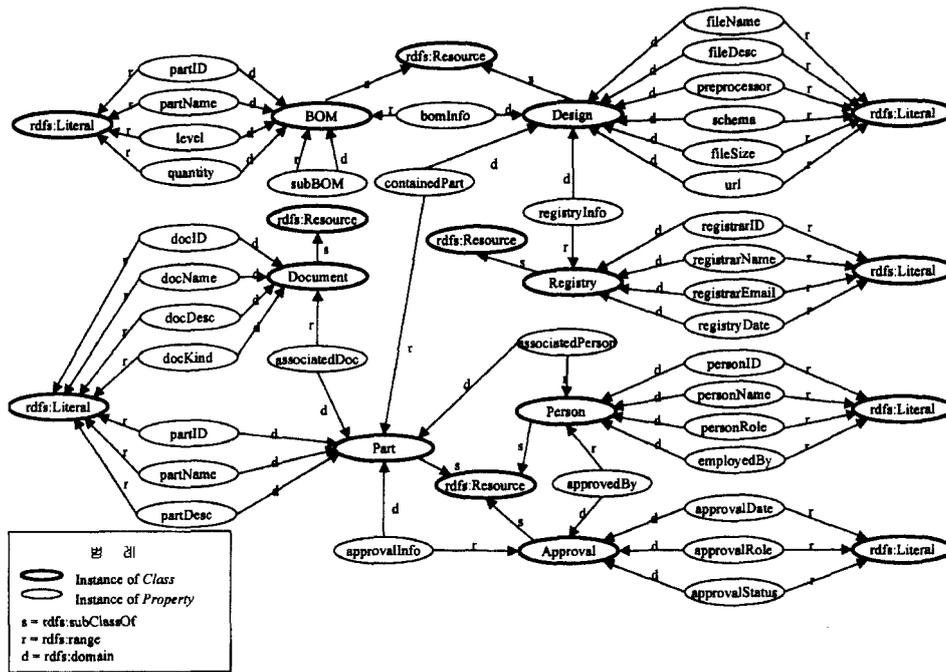


Fig. 3. Metadata elements defined in RDF schema.

리얼화(Serialization) 신택스와 축약형(Abbreviation) 신택스 두 가지가 존재한다. 시리얼화 신택스는 RDF의 모든 기능을 나타내는 정규 표현이며 축약형 신택스는 정규 표현의 간단한 형태라 할 수 있다. 본 논문에서는 모든 RDF 데이터 모델(RDF Schema 포함)을 축약형 신택스를 이용해 표현하였다.

축약형 신택스는 간단하면서도 직관적으로 클래스와 특성을 구분할 수 있다는 장점이 있다. 축약형 신택스에서는 시리얼화 신택스의 Description 요소 대신 기술되는 리소스의 타입을 요소명으로 사용한다. 즉 Description과 Description 대상의 타입을 나타내는 rdfs:type 값 대신에 rdfs:type 값을 요소 이름으로 사용하며, Description의 id나 about 속성은 그대로 유지한다. Fig. 4는 RDF 데이터 모델에 대한 축약형 XML 신택스를 이용한 인코딩의 일부이다. 위의 데이터가 실제로 애플리케이션에서 STEP 파일을 기반으로 동적으로 생성되는 데이터로 데이터베이스에 저장되기 전의 중간 포맷이라 할 수 있다.

4. 시스템 구현

본 논문에서 사용한 도구로는 Iona사의 Java ORB인 OrbixWeb3.0, Oracle XML파서, STEP Tools사

```
<Design about="http://islab.auto.inha.ac.kr/step-files/vh3148-ae22.stp">
  <fileName>V3148-XE22</fileName>
  <fileDesc/>
  <preprocessor>PRO/ENGINEER BY PARAMETRIC TECHNOLOGY
    CORPORATION.9745</preprocessor>
  <schema>CONFIG_CONTROL_DESIGN</schema>
  <fileSize>127K</fileSize>
  <url>http://islab.auto.inha.ac.kr/step-files/vh3148-ae22.stp</url>
  <registryInfo>
    <Registry>
      <registrarID>ykjin</registrarID>
      <registrarName>jin yeon kwon</registrarName>
      <registrarEmail>jin@islab.auto.inha.ac.kr</registrarEmail>
      <registrarDate>1998-11-16</registrarDate>
    </Registry>
  </registryInfo>
  <bomInfo rdfs:resource="#1b1"/>
  <containedPart rdfs:resource="#13006"/>
</Design>
<BOM id="1b1">
  <partID>V3148-XE22</partID>
  <partName>NOT SPECIFIED</partName>
  <level>1</level>
  <quantity>1</quantity>
  <subBOM/>
</BOM>
```

Fig. 4. Example of RDF data model encoded in XML.

의 Java SDAI 라이브러리와 데이터베이스 시스템으로는 마이크로소프트사의 SQL7.0을 이용하였다. 데이터베이스에 저장되어 있는 RDF 메타데이터와 STEP 파트21 포맷을 갖는 텍스트 파일 형태의 제품 데이터를 작업대상으로 하였다. 또한 데이터에 대한 접근 방법은 RDF 메타데이터의 경우 JDBC를 이용

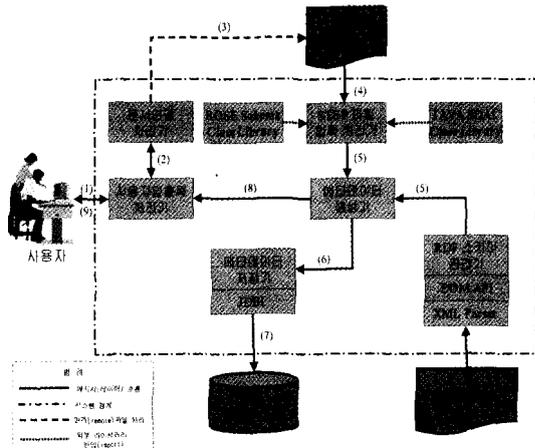


Fig. 5. Procedure of metadata generation.

하여 원격 데이터베이스에 접근하였으며, STEP 파일의 경우 HTTP 프로토콜을 통해 접근하였다.

4.1 메타데이터의 자동생성

메타데이터는 클라이언트에서 등록자가 문서등록 인터페이스를 통해 요청을 하면 서버에서 그 요청에 따라 해당 STEP 파일을 찾아 자동으로 메타데이터를 생성하고 저장하게 된다. 메타데이터 자동 생성을 위한 모듈 구성은 Fig. 5와 같으며 각 모듈을 설명하면 다음과 같다. Fig. 5에서의 숫자는 메타데이터의 생성 과정을 나타내는 순서이다.

• 사용자입력처리기

클라이언트측의 사용자가 문서등록창을 통해 서비스 요청을 하면 사용자의 입력을 분석하여 문서연결 처리기에게 STEP 파일의 URL을 전달해 준다. 사용자는 CORBA 인터페이스인 createMetadata를 통해 서비스를 요청하게 되며 성공적으로 메타데이터가 생성되었으면 True를 그렇지 않으면 False를 반환하게 된다.

• 문서연결처리기

사용자입력처리기에서 받은 URL을 분석하여 STEP 파일이 로컬에 있는지 원격에 있는지 판단한 후 원격일 경우 로컬로 가져와 임시파일 디렉토리에 저장한 후 STEP 파일입력처리기에 파일명을 반환한다.

• STEP파일입력처리기

임시파일 디렉토리에 저장된 STEP 파일을 AP203의 ROSE 스키마 클래스 라이브러리와 JAVA SDAI 클래스 라이브러리들을 이용하여 STEP 파일을 읽어

들이고 처리한다.

• RDF스키마관리기

미리정의 된 RDF 스키마를 읽어 들이고 메타데이터생성기에서 새로운 메타데이터 생성 시에 스키마에 맞는 메타데이터를 생성하기 위해 이용된다.

• 메타데이터생성기

RDF스키마관리기와 STEP파일입력처리기를 이용하여 STEP 파일에서 정보를 추출한 후 RDF 메타데이터를 생성한다. 메타데이터의 생성은 RDF스키마에 의존하여 동적으로 생성하게 된다. 생성된 메타데이터는 데이터베이스 스키마에 따라 변환된 후 데이터베이스에 저장이 된다.

• 메타데이터저장기

메타데이터생성기에서 생성된 RDF 메타데이터를 데이터베이스 스키마에 맞게 변환한 후 데이터베이스에 저장한다.

4.2 질의 처리

서버의 질의 처리는 클라이언트 측의 요청을 받아 실행되게 되는데, 클라이언트와 서버와의 통신은 CORBA IIOP를 이용하게 된다. 사용자의 질의 요청 시 클라이언트 측에서는 IDL 인터페이스의 구현 객체인 Query 객체의 executeQuery 메소드를 호출해 전달하게 된다. 질의어를 전달받은 서버는 질의를 분석하여 SQL 문장으로 변환한 후 데이터베이스에 저장되어 있는 RDF 메타데이터를 검색하게 된다. 검색 결과는 작업관리기에 의해 관리되고 getQueryResults 메소드를 통해 클라이언트에게 검색 결과를 반환하게 된다. IDL 인터페이스를 통해 검색결과에 대한 BOM정보, 등록정보, 메타데이터정보 등이 클라이언트에 전달되고 클라이언트에 STEP 파일을 디스플레이할 수 있는 도구가 있는 경우 형상 뷰를 통해 STEP 파일을 다운로드하여 형상을 볼 수 있다. 질의처리를 위한 모듈 구성은 Fig. 6과 같으며 각 모듈에 대한 설명은 다음과 같다.

• 질의생성기

사용자입력처리기로부터 사용자가 입력한 질의어를 전달받고 파싱한 후 SQL 문장을 생성하게 된다. 클라이언트로부터 전달되는 질의어는 XML형태의 문자열인데 이는 클라이언트 측에서 구성되어 전달되는 값이다.

• 질의처리기

질의생성기로부터 전달받은 질의어를 SQL 문장으로 변환한 후 데이터베이스에 저장되어 있는 RDF

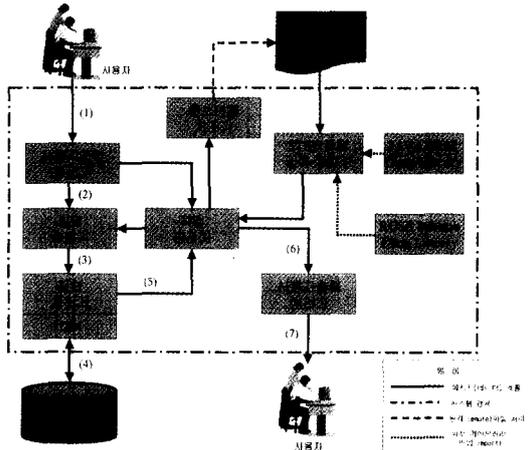


Fig. 6. Procedure of query processing.

메타데이터에 대한 검색을 수행한다. 질의처리는 AND 연산자만을 수행하며 검색된 항목에 대해서는 STEP 파일 자체에 대한 정보에 해당하는 Design 클래스의 특성들(fileName, fileDesc, schema 등)만을 추출하여 클라이언트에게 전달한다. 검색 결과에 대한 사용자의 BOM정보나 등록정보, 메타데이터정보 요구 시에도 작업관리기로부터 호출되어 이용된다.

• 작업관리기

작업관리기는 검색결과에 대해 사용자가 요청한 작업들을 관리하는 모듈이다. BOM정보 요청이나 등록정보 요청 등 검색결과에 대한 사용자의 추가 작업 요청 시에 사용자입력처리기로부터 해당되는 메시지를 전달받아 질의처리를 호출하여 요청되는 정보를 추출하고 클라이언트에게 전달한다. 클라이언트로부터 전달받는 메시지는 STEP 파일의 URL이나 STEP 파일명이며 이는 Design의 about 속성에 해당된다. 이 정보를 질의처리에 전달하여 질의를 수행하고 해당되는 메타데이터를 전달받은 후 클라이언트가 요청한 정보만을 추출하여 사용자출력기에게 전달한다.

5. 제품 데이터 등록 및 검색 예

사용자는 문서등록 인터페이스를 통해 제품데이터와 관련 문서를 등록하게 된다. 사용자는 먼저 ID와 패스워드를 입력한 후 등록하고자 하는 문서종류(STEP 파일 또는 일반 문서)를 선택하고 문서의 URL을 입력한다. 서버는 입력된 ID와 패스워드를 확인하여 등록이 된 사용자인가를 검사하게 된다. 만

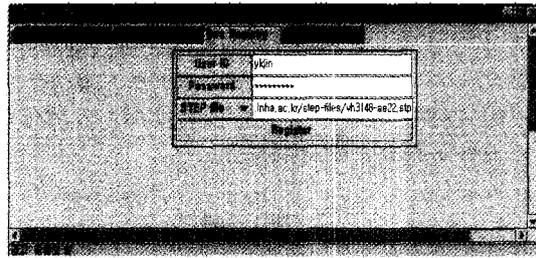


Fig. 7. User interface for registering product data.

약 등록되지 않은 사용자의 경우에는 에러 메시지와 등록을 요구하는 메시지를 띄우며, 등록된 사용자임을 확인하였을 경우에는 문서에 대한 등록요청을 처리한다. 일반 문서의 등록요청에 대해서는 본 연구에서는 고려하지 않았으며, 향후 시스템 확장을 위해 인터페이스만을 제공하였다. Fig. 7은 제품 데이터 등록을 위한 사용자 인터페이스이다.

검색은 카테고리검색과 일반검색, XSL Pattern을 이용한 검색이 있다. 카테고리검색과 일반검색은 서버 쪽에서 이루어지며 XSL Pattern을 이용한 검색은 클라이언트에서 수행된다. 모든 검색 인터페이스에서는 메타데이터 뷰를 제공하여 사용자가 이미 구축되어 있는 메타데이터의 구조를 알 수 있게 해준다. 카테고리 검색을 선택하면 6개의 카테고리가 나타나며 각 카테고리마다 검색항목이 있다. 검색하고자 하는 항목을 활성화시키고 질의어를 입력하여 질의를 요청한다. 여러 항목들이 동시에 선택되었을 경우에는 AND 연산자가 적용되어 검색을 수행하게 된다. Fig. 8은 카테고리검색을 이용한 검색 화면이며, 특정 검색 결과를 선택한 후 마우스의 왼쪽 버튼을 누

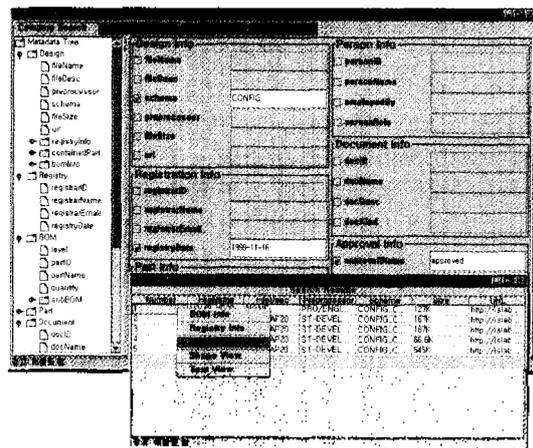


Fig. 8. Search interface and results.

르면 검색 결과에 대해 수행할 수 있는 추가작업 항목이 팝업 메뉴로 뜬다. 본 검색 시스템은 클라이언트와 서버간의 전송 메시지로 XML을 이용하기 때문에 검색된 결과에 대한 클라이언트에서의 활용이 유연하게 확장될 수 있다.

6. 결 론

제품데이터의 효율적인 검색 및 접근은 제품에 대한 개발이 글로벌한 형태가 되면서 그 중요성이 더해가고 있다. 본 논문에서는 효율적인 검색을 위해 메타데이터를 이용하고 데이터에 대한 접근과 표현의 호환성을 위해 RDF, XML, CORBA, STEP 등의 국제 표준을 이용한 웹 기반의 애플리케이션을 구현하였다. 제품데이터의 중요한 특징을 메타데이터는 이용하여 저장함으로써, 검색에 보다 정교하고 신뢰성 있는 정보를 제공하였다. 또한 이러한 메타데이터의 생성은 서버 측에서 자동으로 생성됨으로써 손수 입력을 위한 시간과 노력을 피할 수 있고 실수로 인한 잘못된 정보 제공을 방지할 수 있다.

메타데이터와 제품데이터에 대한 표현이 표준 메카니즘을 따름으로써 기존의 시스템과의 정보 교환시 호환성을 유지할 수 있고, Java와 CORBA는 웹상에서의 애플리케이션이 이기종의 분산 시스템에 접근이 가능하도록 한다. 이렇게 인터넷 상에 분산되어 있는 제품데이터에 대한 메타데이터를 중앙에서 관리하고 웹을 통해 시스템에 접근 가능하게 함으로써 시스템 확장에 대한 구축비용을 절감할 수 있고 시간과 공간적인 제한을 극복할 수 있게 된다. 클라이언트와 서버간의 CORBA IDL 인터페이스를 통한 통신과 전송 메시지로써 XML을 이용함으로써 시스템의 확장과 변경에 큰 유연성을 제공할 수 있다.

향후 연구 과제로는 본 연구에서 고려하지 않은 사용자 관리, 체크 인, 체크 아웃 등의 기능추가와 STEP AP202, AP214 등으로의 확장, 그리고 STEP 파일뿐만 아니라 제품과 관련된 일반 문서에 대한 메타데이터 생성에 관한 연구가 필요하겠다. 또한 RDF 메타데이터를 효율적으로 관계형 또는 객체지향데이터베이스에 저장할 수 있는 방법이 심도있게

연구되어야 할 것이다.

감사의 글

본 연구는 1997년도 특정기초연구지원사업의 지원에 의하여 연구되었습니다(과제번호: 97-02-00-09-01-3).

참고문헌

1. Hamer, P. V. D. and Lepoeter, K., "Managing Design Data: The Five Dimensions of CAD Frameworks, Configuration Management, and Product Data Management", *Proceedings of The IEEE*, Vol. 84, NO. 1, Jan 1996.
2. Willson, D. J., *A powerful solution to bring your legacy data online*, 1999 <http://www.sea-initiative.org>
3. Spooner, D. L. and Hardwick, M., "Using Views for Product Data Exchange", *IEEE Computer Graphics and Applications*, 1997.
4. Lassila, O. and Swick, R. R., *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, Feb 1999. <http://www.w3.org/TR/REC-rdf-syntax/>.
5. Brickley, D. and Guha, R. V., *Resource Description Framework(RDF) Schema Specification*, W3C Proposed Recommendation, March 1999. <http://www.w3.org/TR/PR-rdf-schema/>.
6. ISO TC184/SC4, *XML representation for EXPRESS-driven data*, 1998.
7. Burkett, W. C., *Product Data Markup Language*, 1999. <http://www.pdit.com/pdml/pdmlintro.html>.
8. Jenkins, C., Jackson, M. et, *Automatic RDF Metadata Generation for Resource Discovery*, 1998. http://www.scit.wlv.ac.uk/~ex1253/rdf_paper/.
9. 오동현, 김규태, 정희경, 이수연, "RDF 메타데이터를 이용한 인덱스 기반의 XML/SGML문서 검색 방법에 관한 연구", 한국정보과학회 추계학술발표회, 1999.
10. Veillard, D., *Linux Packages Metadata Mirroring Proposal*. <http://rufus.w3.org/linux/fpnm2html/mirroring.html>.
11. Netscape, *Next Generation 'Aurora' Overview by Netscape Marketing*, <http://home.netscape.com/browsers/future/aurora.html>.
12. Bray, T., Dave Hollander, Andrew Layman, *Namespaces in XML*, W3C Recommendation, 1999. <http://www.w3.org/TR/REC-xml-names/>.



진 연 권

1998년 인하대학교 자동화공학과 학사
2000년 인하대학교 자동화공학과 석사
2000년-현재 (주) 삼진베리클 근무
관심분야: STEP, CALS/EC, XML, CORBA



유 상 봉

1982년 서울대학교 제어계측공학과 학사
1989년 Arizona 주립대학교 전기및컴퓨터공학과 석사
1990년 Purdue대학교 전기및컴퓨터공학과 박사
1989년 AT&T Bell 연구소 연구원
1990년 삼성전자 컴퓨터부문 선임연구원
1992년-현재 인하대학교 자동화공학과 부교수
관심분야: 데이터베이스, STEP, CALS/EC, 지식관리, Web데이터처리