

분산환경에서 표준제품모델(STEP)을 이용한 내용검색

손정모*, 유상봉**, 김영호***, 이수홍****

Content-search in Distributed Environment Using Standard Product Model (STEP)

Jung-Mo Sohn*, Sang-Bong Yoo**, Yeong-ho. Kim*** and Soo-hong Lee****

ABSTRACT

This paper presents a content-search system built on distributed environments using the open product model of STEP. The content-search system searches design data for given product descriptions such as part name and features. Distribute object interfaces has been defined by IDL and distributed product data are searched through CORBA protocols. Web interfaces are also provided for interactive user interfaces. Given a user request, a mediator interacts with distributed search servers and sends collected results back to the user. The mediator has such metadata as location, program name, and other information about product data stored on distributed systems. The search servers use SDAI interfaces to search STEP files or databases. The content-search system promotes the reuse of previous design within a company and the outsourcing of part designs.

Key words : 내용검색, STEP, 분산 시스템, CORBA, SDAI

1. 서 론

최근 제품 라이프사이클의 단축, 경쟁 업체의 증가, 국제 공동 개발 확산 등으로 인하여 기존의 제품 데이터를 활용할 필요성이 증가하고 있다. 여기서 제품 데이터는 회사 내부(in-house)와 외부(out-source)의 데이터를 모두 포함한다. 본 논문에서는 제품 데이터의 표준인 STEP¹⁾과 분산 객체 표준인 CORBA²⁾를 활용하여 개발된 분산환경에서 제품 데이터 검색 시스템을 설명한다.

제품 정보는 STEP으로 표시된 표준 제품 데이터와 그 외의 형태인 레거시(legacy) 데이터로 구분된다. 레거시 데이터로는 IGES, SET 등과 같은 기존의 중립 파일 포맷과 ACIS, AutoCAD 등과 같은 상용 시스템의 파일 포맷이 있다. 이와 같은 레거시 데이터는 주로 도면의 그래픽 정보를 전달하는데 주

안점을 두고 있어, 분산 네트워크 환경에서 이종의 자동화 시스템간 원활한 제품 데이터 교환을 위한 제품 전 수명주기에 걸친 폭 넓은 데이터를 제공하는 데 한계가 있음이 지적되어왔다³⁾. 이러한 문제점을 해결하기 위해 산업 전반에 걸친 다양한 제품의 설계, 해석, 생산, 그리고 관리에 필요한 모든 제품정보를 정의하는 STEP 표준화가 1983년부터 ISO (International Standard Organization)에 의해 추진되고 있다.

STEP의 기본적인 용도는 서로 다른 CAD 시스템간의 파일 교환을 가능하게 하는 중립 파일이다. 어느 기업체의 CAD 시스템이 거래 업체 또는 하청 업체가 사용하는 CAD 시스템과 다를 경우 이러한 중립파일의 기능을 이용하여 제품 데이터를 전달할 수 있다. 이와 같은 기본적인 파일 교환 기능 외에, 데이터 모델과 파일 포맷을 비롯한 모든 구현 기술이 표준으로 정의된 STEP의 특성을 이용하여 본 연구에서는 내용검색(content search) 기능을 제품의 설계 및 생산 과정에서 활용할 수 있도록 한다. 내용검색은 설계 파일에 포함된 조립 부품을 주어진 명세

*학생회원, 인하대학교 자동화공학과
**정회원, 인하대학교 자동화공학과
***정회원, 서울대학교 산업공학과
****정회원, 연세대학교 기계공학과

예를 들어, 부품 이름, 특성 등)를 이용하여 설계 데이터를 검색하는 기능이며, EXPRESS^[5] 스키마와 SDAI(Standard Data Access Interface)^[6]를 이용하여 구현된다. 이 기능은 설계 데이터베이스에 저장된 기존 설계 지식의 활용성을 향상시킨다.

2. 관련 연구

내용 검색은 멀티미디어 분야에서 유사한 이미지 또는 신호를 검색하는 것으로 널리 사용되고 있으며^[7], CAD/CAM 분야에서는 도면 관리 시스템과 PDM에서 주제어나 색인 등을 이용하여 기존 도면의 관리와 검색기능을 지원하고 있다^[8,9]. 본 연구에서 사용하고 있는 STEP 데이터는 표준 정보 모델이기 때문에 분산환경에서의 공개 솔루션(open solution)의 구현에 많이 활용되고 있다^[10,11].

NIIP(The National Industrial Information Infrastructure Protocols) 컨소시엄은 가상 기업(VE: Virtual Enterprise)의 구현에 실시간에서 정보 교환과 공동 작업을 위해 통신의 기능성과 서비스 측면에서는 인터넷을, 객체 기술 측면에서는 OMG의 CORBA를, 정보 모델링 기술 측면에서는 STEP를 핵심 기술로 사용한다^[11]. CORBA는 공동 작업이 가능한, 재사용 가능한, 그리고 이식성이 높은 소프트웨어 컴포넌트를 사용해서 기업의 자원을 통합 시킬 수 있다. STEP은 제품 생명 전 주기에 걸친 제품 정보의 교환에 대한 정보 모델을 제공함으로써 정보 교환과 모델링을 가능하게 한다.

STEP Tools 사는 ST-WebPublisher를 이용해 STEP 파일의 공유를 시도하고 있다^[12]. ST-WebPublisher는 Java 애플릿과 WWW(World Wide Web)에서 STEP 데이터를 사용할 수 있게 해주는 프로그래밍 툴이다. ST-WebPublisher는 Web 브라우저를 통해 볼 수 있는 web 페이지와 GIF 파일, 애플릿들을 만들 수 있다. 이를 통해 STEP 파일을 Web를 통해 공유 할 수 있다.

국내 연구로서 STEPShare는 Java, ROSE library, 그리고CORBA를 사용해 인터넷에서의 3D STEP 브라우저를 개발하였다^[13]. 이는 사용자가 다운로드 받은 애플릿이 실행 후 ORB를 통해 서버에 접속한 하여 STEP데이터를 사용할 수 있다. 이때 서버에서 ROSE를 이용해 STEP 데이터를 읽어 들인다. 한국 과학 기술원의 인터넷 검색기는 Oracle DB와 CORBA를 이용해 인터넷에서 STEP 데이터에 접근할 수 있게 하였다^[14]. ST-Oracle와 AP스키마를 사

용해 Oracle의 DDL(Data Definition Language)을 생성한 다음 이를 이용해 CORBA를 통해 애플릿으로 Oracle의 STEP데이터에 접근 할 수 있다. 또한 솔리드 모델에 대한 특징형상을 분해하여 이들간의 유사성을 평가해 형상 데이터베이스를 검색하는 연구 등도 이루어지고 있다^[15].

본 연구에서 개발한 내용 검색 시스템은 사용자가 제품 데이터 DB나 STEP 응용 프로토콜의 구조에 대한 지식이 없는 상태에서 단지 제품의 특성을 나타내는 단어를 사용하여 제품데이터를 검색할 수 있게 한다. 또한 미디어데이터를 제공하여 STEP 파일이나 DB의 위치도 사용자가 입력하지 않아도 되게 하였다. 최근 CAD/CAM 제품에서의 STEP 모듈의 지원이 증가하고 있어, 본 논문에서 제시한 방법을 이용하여 특정 벤더 제품에 독립적인 제품 데이터 검색 시스템을 실용화 할 수 있게 되었다.

3. 관련 기술 및 시스템 구조

3.1 관련 기술 및 표준

3.1.1 CORBA

CORBA는 분산 컴퓨팅 시스템의 개발을 위한 객체 지향 표준을 제정하기 위해 700여 이상의 관련 단체가 모여서 결성한 컨소시엄인 OMG가 이종의 분산된 환경 하에서 응용 프로그램들이 통합될 수 있는 표준 기술로써 제안한 OMA(Object Management Architecture) 중 일부이다. OMA는 ORB, 객체 서비스(Object Service), 공통 퍼실리티(Common facility), 응용 객체(Application Object)의 4가지 컴포넌트로 구성되어 있으며 그들 간의 관계는 Fig. 1에 표시되어 있다. ORB는 클라이언트가 발행하는 요청을 오브젝트로 보내 그 응답을 클라이언트에 반환하는 것이다. 객체 서비스는 분산 오브젝트 시스템을 구축하는 데 있어 필요한 기본적인 서비스의 집합이다. 공통 퍼실리티는 어플리케이션이 이용되는 범용적인 기능을 제공하는 클래스 오브젝트의 집

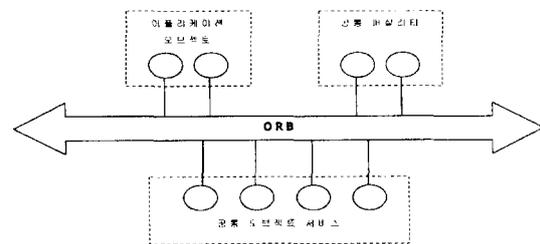


Fig. 1. OMA reference model.

합이다. 응용 객체는 그 이름대로 사용자 어플리케이션 고유의 오브젝트이다. CORBA는 IDL(Interface Definition Language)를 이용해서 오브젝트간의 클라이언트를 정의하게 된다. 이 클라이언트 정의로부터 클라이언트의 스텝(stub) 코드와 서버의 스켈러튼(skeleton) 코드를 생성하여 이들 간의 통신을 할 수 있다¹⁶⁾.

3.1.2 Java

Java는 인터넷의 확산과 더불어 발전해 왔고, 현재의 인터넷에서 가장 인기 있는 언어이다. Java는 가상 머신(VM: Virtual Machine)을 이용해 바이트 코드를 해석하여 실행함으로써 "Write once, Run every where"를 실현하였다. Java 언어로 프로그램을 작성한 후 Java 컴파일러를 통해 바이트 코드를 생성하게 된다. 이렇게 작성된 바이트 코드는 Java 가상 머신이 있는 컴퓨터에서 소스 코드를 다시 컴파일 할 필요 없이 실행될 수 있다. 따라서 Java는 인터넷을 통하여 애플릿으로 다운로드 될 수 있으며 Java를 지원하는 브라우저에서 실행될 수 있다. 이는 인터넷의 보급과 더불어 Java가 널리 사용되고 있는 이유라고 할 수 있다.

3.1.3 STEP

STEP은 제품데이터 즉 제품의 생산에 관계되는 데이터를 컴퓨터가 인식 가능한 형태로 완벽하게 표현하고 교환하기 위한 국제 표준이며, 그 목적은 제품의 전 생명 주기(Life Cycle)에 걸쳐서 제품 데이터를 표현할 수 있는 중립적인 메커니즘을 제공하는 것이다. STEP은 표준 자체의 구조를 반영하여 주는 파트들의 여러 그룹으로 나뉘어져 있다. 이러한 그룹에는 응용 프로토콜(application protocols, AP), 통합자원(integrated resources), 서술 방법(description methods), 구현 방법(implementation methods), 적합성 시험 방법 및 골격(conformance testing methodology and framework), 그리고 가상 시험 세트(abstract test suites)가 있다.

SDAI(Standard Data Access Interface)는 EXPRESS 엔티티들의 생성과 처리를 위한 프로그래밍 인터페이스(programming interface)라고 할 수 있다. SDAI는 응용 프로그램과 엔티티 인스턴스가 존재하는 환경 사이에 인터페이스를 정의한다. 그 환경은 저장소와 스키마 인스턴스 두 가지가 있다. 저장소(repository)는 식별 가능한 데이터의 저장 설비를 말하고, 스키마 인스턴스(schema instance)는 처리할 수 있는 엔티티 인스턴스의 집합으로 SDAI 모델의 논리적 모입이다.

SDAI를 통하여 STEP DB에서 사용자에게 제공하는 기능은 다음과 같다.

- 1) EXPRESS로 정의된 데이터에 대한 접근과 조작
- 2) 단일 응용프로그램에 의한 복수의 데이터 저장소 접근
- 3) 응용프로그램에 의해 조작되는 데이터의 메타 데이터(dictionary)에 대한 접근
- 4) EXPRESS에 의해 정의된 제약조건(constraint) 확인 지시
- 5) 데이터간의 종속성 관계 관리
- 6) 다른 스키마로 생성된 데이터 사용

세션 안에서 SDAI 기능들을 사용하여 데이터의 생성과 처리를 할 수 있다. 세션은 하나의 응용 프로그램에서 SDAI 기능의 사용 시작과 끝 사이에 일어나는 오퍼레이션들이다. 실제 데이터에 접근하기 위해 요구되는 스키마에 대한 정보를 SDAI는 데이터 사전을 사용한다. 이러한 데이터 사전의 지원 없이 응용 프로그램을 사용할 수 있는 데, 이러한 경우 프로그래머는 스키마에 대한 완전한 지식이 필요하다. 다음 Fig. 2는 SDAI의 응용 데이터, 사전, 세션과 SDAI 개체군 조직 사이의 관계들과 응용 프로그램과 SDAI 구현을 보이고 있다.

SDAI는 언어 바인딩(language binding)을 지원한다. 언어 바인딩은 어떤 특정한 언어(예를 들어: C++, C, Java등)의 언어에 SDAI의 기능들을 정의한다. 언어 바인딩에는 초기 바인딩(early binding)과 후기 바인딩(late binding)의 두 가지 타입이 있다. 초기 바인딩은 스키마에 의존적인 인터페이스를 제공하는 것인데 이것은 응용 스키마의 특정 언어로의 변환을 의미한다. 후기 바인딩은 스키마에 관련 없이 독립적인 인터페이스를 제공하는 것이다. 초기 언어 바인딩을 통해 생성된 스키마 의존적인 코드와 결합하여 응용 프로그램에 제공될 SDAI라이브러리를 형성할 수 있다.

응용 프로토콜(AP: Application Protocol)은 각 산업 응용 분야의 정보 요구(information requirement)에 적

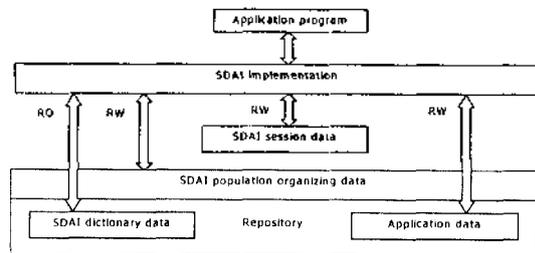


Fig. 2. Relationship among SDAI modules.

합한 정보 교환을 위한 메커니즘을 정의한 것이다. STEP의 응용 프로토콜을 개발하기 위해 먼저 해당 응용 분야에서 필요한 엔티티들을 모은 ARM (Application Reference Model)을 작성하고, 이 데이터 모델을 STEP 파트 번호 40번 대의 공통자원을 이용하여 AIM(Application Interpreted Model)을 개발한다. 결과적으로 STEP 데이터는 AIM에 따라 생성되며, 각 데이터의 의미와 활용을 이해하기 위해서는 ARM을 참조한다.

3.2 시스템 구조

본 시스템의 구조는 Fig. 3과 같고 이는 크게 네 부분으로 구성된다. 클라이언트의 프로그램(애플릿)과 뷰어(Viewer), 미디어이터, 서버 프로그램으로 구성된다. 클라이언트 프로그램은 사용자의 질의를 입력 받고 검색된 부품들의 목록을 보여주는 역할을 하게 된다. 미디어이터는 주어진 질의를 등록된 서버를 통해 검색한다. 서버는 주어진 질의에 해당되는 부품을 AP203파일에서 검색하게 된다. 이런 정보는 CORBA를 통해 클라이언트에 전달된다. ORB (Object Request Broker)사이에서 프로토콜인 IIOP (Internet Inter ORB Protocol)를 사용한다. 검색된 부품은 클라이언트에서 목록에 표시되고 뷰어를 통해서 볼 수 있게 된다. 뷰어를 통해서 사용자는 자신이 원하는 부품을 더욱 쉽게 가려낼 수 있게 된다. 다음은 이 시스템의 실행 순서이다.

- 1) 클라이언트 프로그램으로부터 사용자의 질의 받는다.
- 2) 질의를 미디어이터로 전달한다.
- 3) 미디어이터는 다시 등록된 호스트를 찾아 질의를 보낸다.
- 4) 호스트에서는 질의에 맞는 AP203 및 AP214 부품을 검색한다.
- 5) 검색된 결과를 사용자의 미디어이터에 전달한다.
- 6) 미디어이터는 취합된 결과를 사용자에게 전달한다.

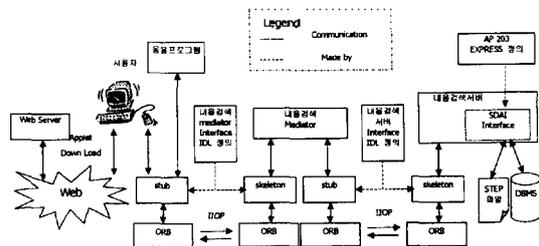


Fig. 3. System architecture.

7) 사용자는 이를 통해 검색결과를 볼 수 있게 된다.

4. 표준 제품 모델을 이용한 내용 검색 기능

4.1 STEP 응용 프로토콜을 이용한 내용 검색

내용검색은 모든 STEP 응용프로토콜에 대하여 적용할 수 있으나, 본 논문에서는 STEP에서 가장 널리 활용되고 있는 AP203^[17]과 AP214^[18]를 이용하여 그 기능을 설명한다. AP203은 구성 관리, 제품 구조, 규격, 기하학적 형상 등으로 구분 할 수 있다. 구성 관리는 제품 데이터 승인, 제품 개발을 진행하기 위한 작업 지원, 제품 모델의 구성 부품의 계획된 사용에 관한 정보, 부품이나 부품의 설계를 생성하는 조직에 관한 정보 등을 포함한다. 제품 구조는 BOM 데이터, 조립품, 부품 등에 관한 정보 등을 나타내며, 규격은 부품이나 제품의 품질과 관련된 정의, 공정 및 규칙 등을 포함하는 문서이다. 기하학적 형상은 고등경계 표현, 각진 경계 표현, 다양체 곡면, 위상 정보가 있는 와이어 프레임, 그리고 위상 정보가 없는 곡면 및 와이어프레임 등 6개의 형상을 포함한다.

AP203의 ARM에서 제품의 특성은 design_information에 포함된다. design_information은 다시 specification, usage_constraint, additional_design_information 등의 응용 객체를 포함하고 있다. 이 중 specification은 공정 중 또는 공정이 끝난 부품의 품질에 관계되는 정의, 프로세스 또는 규칙 등을 기술한다. 이들 각각의 specification은 design_specification, material_specification, process_specification, 또는 surface_finish_specification 중 하나 이다. design_specification은 부품에 대한 설계 요구조건을 정의한 것으로서, 무게의 한계, 크기의 한계, 내침도, 색깔, 투명도, 감촉, 모양 등을 기술한다. material_specification은 제품조립에 사용되는 반 조립 품, 혼합품, 그리고 기본 부품에 대한 속성들을 기술하고 있다. process_specification은 제품이나 물질들이 생산과정에서 필요한 공정을 정의한 것으로서 열처리, 용접, 포장, 표시 등을 기술한다. surface_finish_specification은 공정 중 또는 완성된 부품에 대해서 표면의 구조 또는 보호 코팅에 대한 속성을 기술한 것이다. 이상의 모든 설계 정보가 AP203에서 document에 저장되며, 내용검색에서는 사용자에게 의해 주어진 하나 이상의 특성을 이용하여 해당되는 제품 또는 부품을 검색한다.

필요한 부품의 명세 즉, 부품의 이름이나 색깔, 재

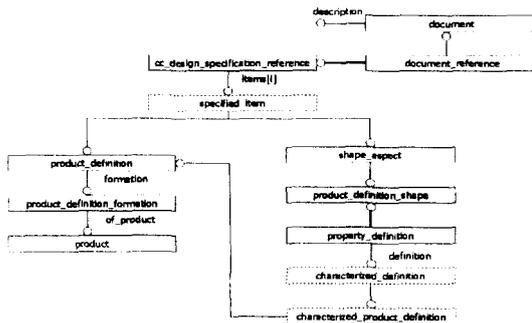


Fig. 4. Content-search schema of STEP AP203.

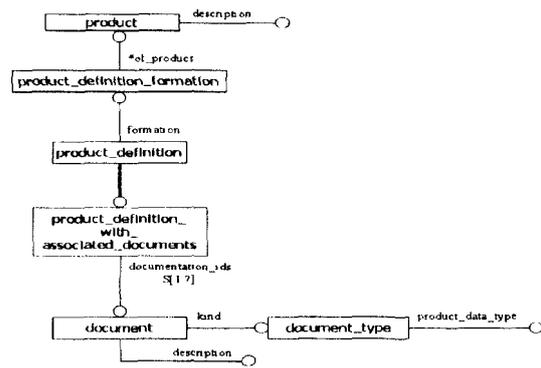


Fig. 5. Content-search schema of STEP AP214.

질 등의 특성을 이용해서 document의 description이나 product의 name, description등을 통해 검색할 수 있다. document의 description을 통해 검색할 경우에는 Fig. 4에서 보는 것과 같이 다음 경로를 이용해서 접근하고 검색할 수 있다. cc_design_specification-reference를 통해서 Part정보를 가지고 있는 product에 접근할 수 있게 된다. 해당되는 document의 cc_design_specification을 이용해 design_displine-product_definition의 정보를 가지고 있는 product_definition을 얻은 다음 product_definition-formation을 통해 Part의 정보를 가지고 있는 product에 접근해서 부품에 대한 정보에 접근 할 수 있다. 또는 cc_design_specification_reference에서 shape_aspect를 통해 product_definition_shape의 definition을 이용해 product에 대한 정보에 접근할 수 있다.

AP214는 자동차의 제품데이터 교환하기 위한 응용 프로토콜이다. 이 응용 프로토콜은 자동차 산업에서 요구되는 여러 가지 개발 단계의범위, 항목이나 정보 요구(Information requirements)등에 적합한 통합 자원(integrated resource)을 규정하고 있다. AP214는 제품 정의(product definition), 스타일링(styling), 디자인(design), 제품 평가(evaluation), 공정 계획(production planning), 툴 설계(tool design), 툴 제작(tool manufacturing), 품질 관리(quality control)에 대한 자동차의 전 생산 주기의 여러 공정들을 포함하고 있다. AP214는 자동차나 부품의 기계적인 부분(mechanical aspect)만을 포함하고 전기적 부분이나, 유체(공기, 물)관련 부분, 유한 요소 해석(FEA: Finite element analysis)은 포함하지 않고 있다.

AP214에서의 내용검색은 AP203과 같이 document의 description이나 product의 name, description 등을 통해 Fig. 5에서 보는 것과 같이 검색할

수 있다. document의 description을 통해 검색할 경우에는 product_definition_with_associated_documents를 통해 product데이터에 접근할 수 있다. 또한 document_type등을 통해 만약 document가 html파일이거나 텍스트 파일일 경우 해당 파일을 검색하여 더욱 다양한 검색을 할 수 있다.

4.2 내용 검색 기능 구현

내용검색은 SDAI(Standard Data Access Interface)^[6]를 이용해서 각 엔티티 들의 인스턴스를 생성, 처리한다. SDAI는 파일시스템 또는 데이터베이스 시스템의 종류와 무관하게 STEP 데이터의 접근 함수들을 정의한 API(Application Program Interface)의 일종이며, 여기서는 NIST에서 개발한 SCL(STEP Class Library)을 사용하였다. EXPRESS SCHEMA를 읽어 들여서 이것을 C++에서 사용할 수 있는 C++ Schema Library를 생성하게 된다. 이를 이용해서 C++ 프로그램으로 STEP 데이터를 처리할 수 있는 응용 프로그램을 작성할 수 있다.

부품의 명세를 입력하게 되면 document의 description, 부품의 이름, 부품의 description 등을 참조하여 내용과 일치하는 부분을 검색한다. 이때 SDAI의 함수를 사용해서 속성을 읽어와서 문자열의 일치 부분을 검색하게 된다. 만약 검색사항이 존재하면 검색 서버는 존재하는 document의 목록을 CORBA를 통해서 미디에이터에 보내게 된다. 미디에이터는 검색 서버들의 결과를 목록으로 만든 다음 이것의 결과를 클라이언트에게 보내 주게 된다. 클라이언트에서는 결과를 보여주고 사용자가 원하는 부품의 형상을 뷰어를 통해 보여 준다.

4.2.1 IDL 클라이언트

네트워크에서 객체를 사용하기 위해 사용자의 응

용 프로그램이나 애플릿과 내용 검색 미디어어터 그리고 내용 검색 서버와 내용 검색 미디어어터 사이에 인터페이스를 정의해야 한다. 이 정의를 사용해 CORBA에서 스텝과 스켈러톤을 통해 통신을 하게 된다. 내용검색기의 미디어어터와 클라이언트 사이의 IDL의 정의는 다음과 같다.

```
struct Product_Data{
    string part_name;      // part의 이름.
    string part_description; // part에 대한 설명.
    string part_filename;  // part가 저장되어
                          // 있는 파일 이름.
    string part_hostname;  // host의 이름
    string ap_type;       // AP의 타입
};
interface GetPart{
//    FOR SEARCH.
    short getNextPart(out Product_Data p_data);
    short getResultCount();
    void resetPart();
    void putQuery(in string query);
...
}
```

여기서 사용된 putQuery()함수는 사용자에게서 주어진 질의에 해당되는 Product를 찾아 linked list에 저장하게 된다. resetPart()함수는 이렇게 저장된 linked list의 처음을 가르치게 하고 getNextPart()함수를 사용해 Product에 대한 정보를 가져올 수 있게 된다. 이 인터페이스 함수는 미디어어터와 내용검색 서버에도 그대로 적용된다.

미디어어터와 내용 검색 서버 사이에서는 미디어어터의 서버에 대한 정보와 부품이 어떤 응용 프로토콜을 사용하는 지에 대한 타입 정보를 가지고 있기 때문에 Product_Data 스트럭처에서 서버에 대한 정보(part_hostname)와 타입 정보(ap_type)가 제거된 Product_Data 스트럭처를 사용한다.

4.2.2 내용 검색 미디어어터

내용 검색 미디어어터는 사용자 프로그램에서 질의를 입력 받고, 자신에게 등록된 내용 검색 서버에서 검색 결과를 가져 오는 역할을 수행 한다. 따라서 미디어어터는 사용자의 질의를 받는 스켈러톤과 검색 서버로 검색을 요청하는 스텝이 있고 검색된 결과를 관리하는 부분으로 되어 있다. 결과를 관리하는 부분에서는 검색된 결과를 메타데이터를 이용해 메모리에 임시 저장한다. 다음 Fig. 6은 미디어어터

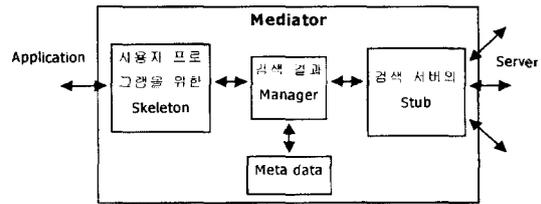


Fig. 6. Structure of mediator.

의 구조를 나타낸다.

사용자에게서 입력되어 지는 질의는 AND 및 OR 검색이 가능하다. 각각의 단어나 문장은 큰 따옴표 ("") 또는 괄호("(")로 구분된다. 예를 들어, (A와 B) 또는 C가 포함된 것을 찾는다면 ("A" AND "B") OR ("C") 라고 입력하면 된다. 괄호로 구분된 피연산자 뒤에는 괄호로 구분된 피연산자가 있어야 한다. 예를 들어, (A) AND "B"는 에러를 출력한다. (A) AND (B)라 하여야 한다. 한 단어나 문장일 경우는 "A", (A), 그리고 A와 같이 세 가지 방식으로 입력한다. 이렇게 주어진 질의는 다시 서버에게 주어져 처리된다.

미디어어터는 처리된 결과를 메타데이터에 임시로 보관하게 된다. 미디어어터의 메타데이터는 어느 호스트의 어떤 파일이 어떤 스키마로 질의에 맞는 부품이 존재하는 지에 대한 정보를 가지게 된다. 메타데이터의 정의는 다음과 같다.

```
class StepPartData extends java.lang.Object
{
    String name;
    String description;
    String filename;
    String hostname;
    String ap_type;
...
}
```

4.2.3 내용 검색 서버

내용 검색 서버는 NIST의 SCL3.1을 사용해서 구현하였다. SCL3.1은 STEP core Class Library와 Data Editor Library, 기타 유틸리티 Library로 구성되어 있다. 또한, 사용자에게서 주어진 EXPRESS 파일을 SCL3.1의 fedex_plus를 사용해 초기 바인딩을 하여 Schema Library를 만들 수 있다. STEP core Class Library는 Schema Library에 있는 모든 클래스들의 기본 클래스인 STEPentity, STEPattribute, Registry등의 클래스를 가지고 있다. STEPentity는

EXPRESS에 기술된 엔티티의 속성과 기술적인 정보에 대한 값을 포함하는, 엔티티 인스턴스의 정보에 접근할 수 있게 한다. STEPAttribute는 엔티티의 속성의 값과 기술적인 정보에 접근할 수 있는 방법을 제공하고, Registry는 스키마와 타입, 속성의 사전 정보를 제공한다. 실행 시에 한 프로그램에서 두개의 스키마를 사용하기 위해서는 Registry의 스키마를 바꿔주면 된다. Data Editor Library는 읽어 들인 인스턴스들의 관리와 개별적인 인스턴스에 대해 편집하는 기능들을 제공하고 있다¹⁾. 이러한 기능을 이용하여 AP203의 경우 알고리즘1을 사용하여 검색 결과를 찾을 수 있다.

알고리즘1: AP203데이터의 내용검색

입력: 미디어이터로부터 전달된 사용자 질의 (Query).

출력: 질의에 적합한 Product의 리스트.

- 1) SCL을 위한 Schema를 초기화 한다.
- 2) 각 엔티티에서 document나 product를 찾아 2.1, 2.2, 2.3을 반복한다.
 - 2.1) document나 product가 질의에 적합한지를 검사한다.
 - 2.2) 만약 document일 경우, 다음 2.1.1, 2.1.2, 2.1.3을 반복한다.
 - 2.1.1) cc_design_specification_reference로 product_definition을 알아낸다.
 - 2.1.2) product_definition으로 product_definition_formation을 경유해 product를 검색한다.
 - 2.1.3) product를 결과리스트에 추가한다.
 - 2.3) 만약 product일 경우, product를 결과리스트에 추가한다.

하나이상의 조건을 사용하여 질의를 입력 받았을 경우에 알고리즘2를 사용하여 사용자의 질의와 검색 항목 사이의 비교를 한다.

알고리즘2: 검색 조건 처리

입력: 사용자의 질의

출력: 적합성 여부 (TRUE, FALSE)

- 1) 만약 하나의 항목이 아닌 경우 다음 1.1, 1.2, 1.3, 1.4를 실행한다.
 - 1.1) 문장을 하나의 연산자가 둘이상 이고 피연산항목이 세 개 이상일 때,이들을 두개의 피연산자와 하나의 연산자로 나눈다. 예를 들어, "A" AND "B" OR "C" AND "D"일

경우("A" AND "B" OR "C") AND ("D")로 나눈다.

- 1.2) 첫 번째 피연산자를 추출하여 이를 파라미터로 현재 함수를 다시 호출한다.
- 1.3) 두 번째 피연산자를 추출하여 이를 파라미터로 현재 함수를 다시 호출한다.
- 1.4) 1.2와 1.3에서의 결과들을 연산자로 연산한 다음 이것을 반환한다.

2) 하나의 항목일 경우, 검색항목과 비교하여 같은 것이 있으면, TRUE를 없으면 FALSE를 반환한다.

질의가 "A" AND "B" OR "C" AND "D"로 들어올 경우 질의를("A" AND "B" OR "C") AND ("D")로 변형시킨 후 이것을 "A" AND "B" OR "C"를 인수로 다시 함수를 호출하여 결과를 얻고 "C"를 인수로 하여 함수를 호출하여 결과를 얻은 다음 AND연산을 통해 결과를 반환한다. Fig. 7은 이것을 그림으로 보여 주고 있다.

AP214의 경우에는 알고리즘3과 같은 방법으로 HTML파일이나 텍스트 파일의 내부를 검사하여 검색할 수 있다.

알고리즘3: AP214데이터의 내용검색

입력: 미디어이터로부터 전달된 사용자 질의 (Query).

출력: 질의에 적합한 Product의 리스트.

- 1) SCL을 위한 Schema를 초기화 한다.
- 2) 각 엔티티에서 document나 product를 찾아 2.1, 2.2, 2.3을 반복한다.
 - 2.1) document나 product가 질의에 적합한지를

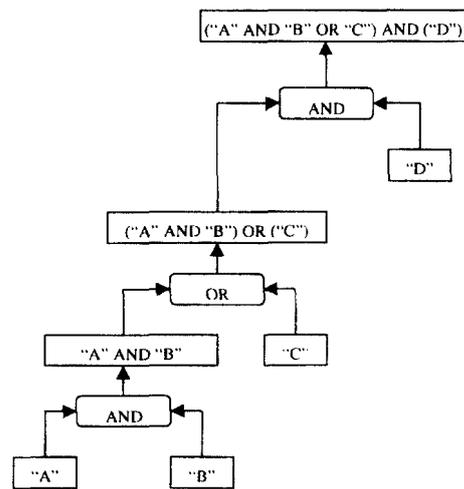


Fig. 7. Example of query operation.

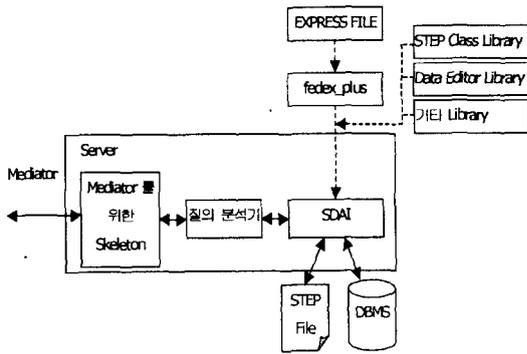


Fig. 8. Structure of content-search server.

검사한다. document일 경우 해당 document가 HTML파일이나 텍스트 파일일 경우 해당파일의 내용 중에 질의에 적합한지를 검사한다.

- 2.2) 만약 document일 경우, 다음 2.1.1, 2.1.2, 2.1.3을 반복한다.
 - 2.1.1) product_definition_with_associated_documents로 product_defintion을 알아낸다.
 - 2.1.2) product_definition으로 product_definition_formation을 경유해 product를 검색한다.
 - 2.1.3) product를 결과 리스트에 추가한다.
- 2.3) product일 경우, product를 결과 리스트에 추가한다.

다음 Fig. 8은 SCL3.1을 이용한 서버의 개발과정과 구조를 나타낸다.

미디어이터를 통해 새로 들어온 질의를 질의 분석기가 질의에 만족하는 지를 테스트한 다음 이것의 결과를 스텔러본을 통해 미디어이터에게로 보낸다.

4.3 시스템의 개발과정

본 시스템의 구현환경은 다음과 같다. Sun Solaris를 운영체제로 하는 Unix 머신에서 NIST에서 제공하는 SCL3.1과 Orbix2.3을 사용하여 서버를 구성하였고, C++ Solaris 버전을 통해 컴파일 하였다. 미디어이터는 Java 컴파일러인 JDK1.1.7과 OrbixWeb3.0을 사용하여 작성하였다. 클라이언트는 JAVA를 이용해서 작성하였고 JDK1.1.7과 Swing1.1를 사용하여 웹에서 애플릿으로 실행 가능하도록 하였다. 본 시스템의 개발과정은 다음과 같으며, 이를 도시하면 Fig. 9와 같다.

- 1) 내용검색 미디어이터와 클라이언트 응용 프로그램 또는 애플릿과의 인터페이스를 정의한 IDL파일과 미디어이터와 내용검색 서버사이의 인터페이스를 정의한 IDL파일을 작성한다.
- 2) 1)에서 작성된 IDL파일로 각각의 Skeleton코드와 Stub코드를 IDL컴파일러를 통해 생성한다. 클라이언트 응용프로그램이나 애플릿과 미디어이터 사이의 IDL파일은 자바코드를 생성하고, 미디어이터와 내용검색 서버 사이의 IDL로는 Stub은 자바코드로 Skeleton은 C++코드로 생성한다.
- 3) 프로그래머가 작성한 클라이언트 응용 프로그램 또는 애플릿의 코드와 2)에서 생성된 내용검색 미디어이터와 클라이언트간의 Stub코드를 사용해 클라

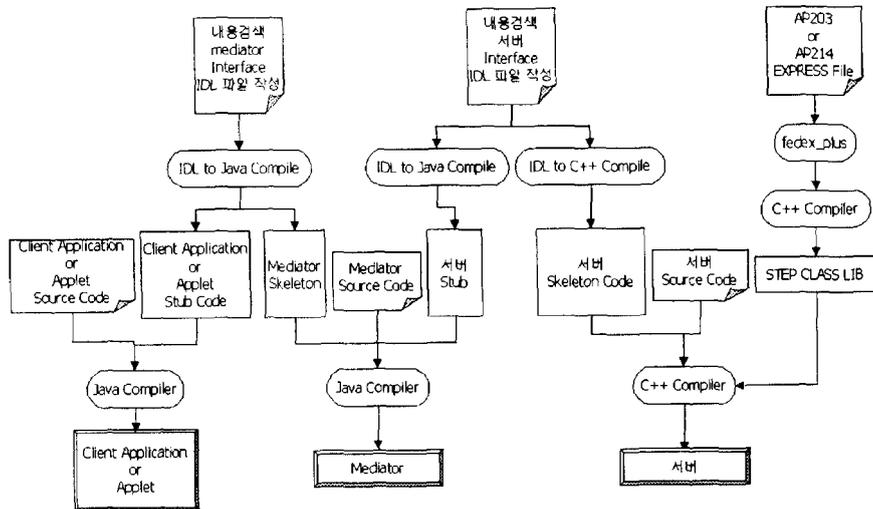


Fig. 9. Development procedure.

이엔트 응용 프로그램 또는 애플릿을 자바 컴파일러를 통해 만든다.

4) 클라이언트와 미디어이터 사이의 Skeleton코드와 내용검색서버와 미디어이터 사이의 Stub코드, 그리고 프로그래머가 작성한 미디어이터 코드를 자바 컴파일러로 컴파일하여 미디어이터를 만든다.

5) AP203 또는 AP214의 EXPRESS 파일을 SCL3.1에서 제공하는 fedex_plus를 통해 컴파일한다. 이렇게 하면 fedex_plus는 초기 바인딩을 하여 C++에서 STEP 데이터에 접근할 수 있는 C++코드를 생성하게 된다.

6) 5)에서 생성된 코드를 다시 C++컴파일러로 컴파일하여 STEP Class 라이브러리를 만들 수 있다.

7) 2)에서 생성된 내용검색서버와 미디어이터 사이의 Skeleton 코드와 6)의 STEP Class 라이브러리, 그리고 프로그래머가 작성한 코드를 C++ 컴파일러를 통해 내용검색 서버를 생성한다.

5. 시스템의 구현환경과 실행 예

본 시스템의 동작은 처음에 사용자가 애플릿이 포함된 웹 페이지에 접속하여 애플릿이 실행하게 됨으로써 시작된다. 사용자는 애플릿에서 미디어이터를 결정하게 되고 애플릿은 사용자가 입력한 질의를 미디어이터로 CORBA를 사용해 접속하게 된다. 접속 후 사용자가 자신이 원하는 부품의 명세를 입력하면 해당 부품의 리스트와 어떤 서버에 데이터가 존재하는지를 미디어이터를 통해 사용자의 애플릿에 보여지게 된다. 사용자는 원하는 부품을 선택해서 더욱 세부적인 명세를 얻을 수 있고 원한다면 부품의 형상정보를 보여 주게 된다.

Fig. 10은 사용자가 원하는 gripper의 부품 세부 명세와 형상정보를 보여주는 것이다. 사용자의 입력은 ORB를 통해 서버의 SCL이 STEP파일을 읽어

들여 주어진 명세에 맞는 부품을 찾는다. 이렇게 해서 검색된 부품을 사용자에게 보여주는 화면이다.

6. 결 론

본 논문에서 기술한 내용검색 기능은 사용자 또는 다른 응용프로그램이 사용하여 파일 이름이나 도면 ID 없이 위에서 설명한 제품 또는 부품의 다양한 특성을 이용하여 데이터베이스에 저장된 기존 설계 정보를 찾을 수 있도록 함으로써 설계 데이터의 재사용을 활성화시킨다. 표준 포맷으로 표현되지 않은 레거시 데이터는 다양하고 공개되지 않은 파일 구조를 갖고 있기 때문에 이와 같은 내용검색 기능을 구현하여 활용하기가 곤란하다. 이와 같은 기능은 표준 제품 모델의 개방성을 제품의 설계 과정에서 활용한 것이다.

본 논문에서 기술된 시스템은 미디어이터에 포함된 메타데이터의 확장을 통하여 AP 203외의 다른 응용 프로토콜을 포함할 수 있다. 그리고 내용 검색 서버가 설치되지 않고 파일 시스템이나 일반 DBMS에 저장된 제품 데이터의 검색을 위하여서는 ODBC 등의 인터페이스가 필요하다. 현재 구현된 방법은 전체 제품 데이터를 대상으로 검색하므로 많은 검색 시간이 필요하다. 이러한 문제점은 제품 데이터를 모듈화 하여 데이터웨어하우스를 구축함으로써 해결될 수 있다.

감사의 글

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

참고문헌

1. ISO, Industrial Automation Systems and Integration Product Data Representation and Exchange Part I: Overview and Fundamental Principles, Geneva, 1994.
2. STEP 연구회, 제품 모델 정보 교환을 위한 국제 표준(ISO 10303) STEP, 성안당, 1996.
3. The Common Object Request Broker: Architecture and Specification, OMG, 1995.
4. D. N. Chorafas, The Engineering Database, Butterworths, 1988.
5. ISO, Industrial Automation Systems and Integration Product Data Representation and Exchange Part II: Description Methods: The EXPRESS Language Refer-

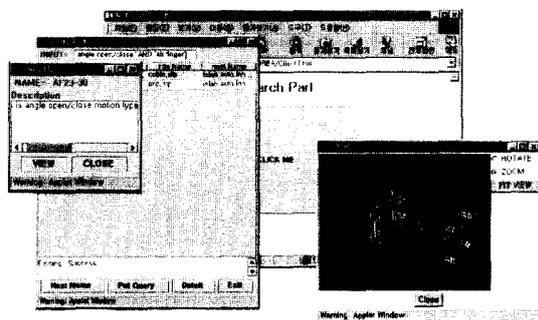
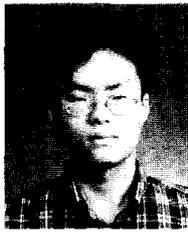


Fig. 10. Example of content-search.

ence Manual, Geneva, 1994.

6. ISO, Industrial Automation Systems and Integration Product Data Representation and Exchange Part 22: Implementation Methods: Standard Data Access Interface Specification (Final Draft International Standard), Geneva, 1998.
7. Prabhat K. Andleigh, Kiran Thakrar, Multimedia system design, Prentice Hall, 1995.
8. 강석호, 김영호, 황영현, 김대환, "PDM 시스템의 평가에 관한 연구", 한국 CALS/EC 학회지, 제 1권, 제 1호, pp. 209-226.
9. CIMdata, Product Data Management: The Definition, 1997.
10. Martin Hardwick, David L.Spooner, Tom Rando and K. C. Morris, "Data Protocols for the industrial Virtual Enterprise", IEEE Internet Computing, pp. 20-29, Jan-Feb 1997.
11. "The NIIP Reference Architecture: Concepts and Guidelines", <http://www.niip.org>, 1995.
12. STEP Tools Inc, ST-WebPublisher, <http://www.step-tools.com/products/stwebpub/>.
13. 최 영, 양상웅, "STEPShare: 웹 환경에서의 CORBA 기반 3D STEP 브라우저", '98 한국 CAD/CAM 학회 학술발표회 논문집, pp. 17-23, 1998. 2.12.
14. 김준환, 한순홍 외, "CORBA를 통한 STEP 데이터베이스의 인터넷 검색", '98 한국 CAD/CAM 학회 학술 발표회 논문집, pp. 24-29, 1998. 2.12.
15. 김철영, 김영호, 강석호, "기계부품의 형상검색을 위한 유사성 평가 방법", '99 한국 CAD/CAM 학회 학술 발표회 논문집, pp. 381-386, 1999. 2. 9.
16. H. Onozawa, 분산 오브젝트 지향기술 CORBA, 홍릉과학출판사, 1997.
17. ISO, Industrial Automation Systems and Integration Product Data Representation and Exchange Part 203: Configuration controlled design, Geneva, 1994.
18. ISO, Industrial Automation Systems and Integration Product Data Representation and Exchange Part 214: Core data for automotive mechanical design process, March 1997.
19. Katherine C. Morris, David Sauder, and Sandy Ressler, "Validation Testing System: Reusable Software Component", National Institute of Standards and Technology, October 1992.



손 정 모

1998년 인하대학교 자동화공학과 학사
 1998년-현재 인하대학교 자동화공학과 석사 과정
 관심분야: STEP, CALS, CORBA, JAVA, CAD/CAM



유 상 봉

1982년 서울대학교 제어계측공학과 학사
 1986년 Arizona 주립대학교 전기및컴퓨터공학과 석사
 1990년 Purdue대학교 전기및 컴퓨터공학과 박사
 1989년 AT&T Bell 연구소 연구원
 1990년 삼성전자 컴퓨터부문 선임연구원
 1992년-현재 인하대학교 자동화공학과 부교수
 관심분야: 공학 데이터베이스, STEP, CALS/EC, PDM, Internet Computing, 동시공학



김 영 호

1985년 서울대학교 산업공학과 학사
 1987년 서울대학교 산업공학과 석사
 1988년-1990년 한국과학기술연구원 연구원
 1993년 North Carolina 주립대 산업공학과 공학박사
 1993년-1995년 전북대학교 산업공학과 전임강사
 1995년-현재 서울대학교 산업공학과 조교수
 관심분야: 정보시스템, 동시공학, 인터넷 응용, Internet 기반 PDM개발, Work flow 관리 시스템 개발, 의료정보 시스템 등



이 수 홍

1981년 서울대 기계공학과 학사
 1983년 서울대 기계공학과 석사
 1991년 Stanford 대학 Design Division Concurrent Engineering 전공, 박사
 1991년-1992년 Lockheed Missile and Space Co. Cable Harness Design System개발 Post-Doc.
 1983년-1994년 KIMM CAD/CAM실, 선임 연구원
 1994년-현재 연세대학교 기계공학과, 부교수
 관심분야: 동시공학설계, 지식기반시스템 설계, DFM