

웹 서비스 기반의 영화 콘텐츠 검색 시스템 개발

곽길신, 주경수*

*순천향대학교 공과대학 컴퓨터학부
e-mail:phnewma@hotmail.com, gsoojoo@sch.ac.kr

A Development of A Movie Contents Retrieval System based on Web Service

Kwak Kil-Sin, Joo Kyung-Soo*

*Dept. of Computer Science and Engineering, College of
Engineering
SoonChunHyang University, Asan 336-745, Korea

요 약

디지털화 된 멀티미디어 데이터가 증가함에 따라 데이터의 효과적인 분류 및 검색 작업이 더욱 중요해지고 있으며, 네트워크와 하드웨어의 발전으로 영상 정보를 검색함에 있어 기존의 웹상의 클라이언트/서버 기반 검색만으로는 부족하게 되었다. 이에 따라 영상 정보를 웹상에서 뿐만 아니라 모바일 같은 이기종간의 검색이 필요하다. 그러나 이기종 간의 플랫폼에서 동일한 정보를 얻기 위해서는 상호운용성에 문제가 있다. 웹 서비스(Web Service)는 분산 애플리케이션에 대한 언어-중립적(language-neutral)과 벤더-중립적(vendor-neutral)[5]을 제시하고 있어 기존의 서로 다른 플랫폼간의 정보 이전 문제를 해결할 수 있다. 따라서 본 논문에서는 영화 콘텐츠 검색 시스템을 웹 서비스화 하였다.

1. 서론

지난 수십 년간 지속되어 온 아날로그 시대가 디지털이라는 새로운 기술 환경으로 바뀌면서 각종 정보가 디지털화 되어가고 있다. 게다가 영상데이터 압축 기술의 발전, 입출력 장치와 저장 장치의 발달, 컴퓨터 처리 속도의 향상, 통신 기술의 급속한 발전 등으로 멀티미디어 콘텐츠의 편리한 제작 및 전송이 가능해졌으며, 디지털화 된 데이터의 양도 과거와는 비교할 수 없을 정도의 속도로 증가하고 있다[2]. 이처럼 디지털화 한 멀티미디어 데이터가 증가하면서 데이터의 효과적인 분류 및 검색 작업이 더욱 중요해지고 있다.

네트워크와 하드웨어의 발전으로 영화 콘텐츠를 검색함에 있어 기존의 단순 클라이언트/서버를 기반으로 한 웹상에서의 검색만으로는 부족하게 되었다. 좀 더 발전된 방법으로 고려 할 수 있는 것으로 하나의 데이터베이스에 영화 콘텐츠의 정보를 담아 웹 뿐 아니라 모바일 같은 이

기종에서의 검색 기능을 구현하는 것이다. 그러나 이기종간의 플랫폼에서 동일한 정보를 얻기 위해서는 상호운용성에 문제가 있다[6].

최근 웹 서비스(Web Service)를 통해 분산 컴퓨팅 환경의 표준, 즉 분산 애플리케이션에 대한 언어-중립적(language-neutral)과 벤더-중립적(vendor-neutral)을 제시하고 있다[5]. 따라서 웹 서비스를 이용하면 기존의 서로 다른 플랫폼간의 정보 이전 문제를 해결할 수 있다. 본 논문에서는 영화 콘텐츠 검색 시스템을 웹 서비스화 하여 다양한 플랫폼에서 검색이 가능하도록 기반을 마련하였다

본 논문에서 2장은 웹 서비스화 할 영상정보의 메타데이터를 소개하고 웹 서비스와 관련된 기술을 간단히 언급하고 3장은 시스템 아키텍처 및 구현에 관하여 다룰 것이며 4장에서는 결론짓도록 하겠다.

2. 관련 연구

2.1 영상정보 메타데이터 기반의 영화 콘텐츠

내용량의 비정형 데이터라는 특성을 가지는 영상정보 데이터를 효율적으로 관리하기 위해서는 우선, 응용분야에 따른 사용자의 요구를 분석하고, 질의유형을 파악하여 영상정보가 포함하고 있는 내용의 특성을 기술할 수 있는 영상정보 메타데이터를 추출, 분류 및 모델링 하는 작업이 필수적이다[1]. 따라서 본 논문에서는 한국영상자료원의 지원에 의해 연구되어 데이터 관리 교환 표준을 위해 제안된 영상정보 메타데이터 표준 제시에 따른 메타데이터를 기반으로 개발한 검색엔진을 기반으로 웹 서비스화 하였다. 본래 영상정보 메타데이터는 영상과 관련된 모든 정보를 총칭하는 개념이지만, 제안된 연구 대상에서는 멀티미디어 자원 및 문화예술정보 자원 가운데 영화로 대표되는 영상정보에만 한정하였다. 성균관대학교 정보관리연구소에서 제안된 영상정보 메타데이터의 추출 결과 저작, 표현, 실현, 개별 자료와 같은 4가지 메타데이터 요소 세트가 추출되었다. 저작은 작품의 명확한 지적, 예술적 창작물을 나타내는 엔터티로 총 20가지의 요소가 추출되었으며 표현은 작품이 문자, 숫자, 뮤지컬, 안무, 사운드, 이미지, 오브젝트, 움직임 또는 여러 개가 조합된 형태로 나타나는 예술적 실현을 총 17가지의 요소로 나타낸다. 실현은 표현의 물리적 구체화로 총43가지의 요소가 추출되었다. 마지막으로 개별자료는 실현의 단일 제품으로 총 9가지의 요소가 추출되었다[1].

2.2 웹 서비스

자바(JAVA)는 1995년에 소개된 이후로 단순한 웹 페이지를 훨씬 동적으로 만들기 위한 언어에서 완벽한 플랫폼으로 변모하였다. 거의 대다수 애플리케이션 서버 업체들은 기업용 애플리케이션을 개발하는데 J2EE를 사용하고 있고 자바 기반의 J2SE의 스윙(Swing) 같은 사용자 인터페이스라이브리어나 네트워킹 기능들은 데스크탑 애플리케이션을 개발하는데 손색이 없다. 또한 J2ME를 사용하여 셋톱박스나 휴대전화 같은 작은 기기까지 자바 플랫폼을 확장하고 있다. 웹 서비스는 이러한 자바 환경에서의 상호연용 분산 시스템을 개발하기에 알맞다[6].

웹 서비스는 XML을 기반으로 표현하고 그 위에 웹 서비스를 기술하는 WSDL(Web Service Description Language), 웹 서비스를 나열하고 찾기 위한 UDDI(Universal Description, Discovery and Integration) 그리고 웹 서비스를 호출하고 메시지를 교환하기 위한 SOAP(Simple Object Access Protocol) 등을 사용하고

있어 기존의 분산된 서비스들을 쉽고 유연하게 묶을 수 있다.[4].

SOAP은 소프트웨어 서비스들 간 네트워크 통신을 위한 표준으로 분산 환경에서 정보를 교환하기 위해 제안된 XML 기반의 메시징 프로토콜이다. 또한, SOAP은 XML문서를 SMTP, HTTP, FTP 등 다양한 표준 인터넷 기술로 전송하는 표준 포장 구조(standard packaging structure)를 제공한다. 이에 따라 XML을 사용하는 애플리케이션이 동작하는 운영체제, 언어 그리고 객체 모델에 관계없이 애플리케이션 사이의 통신을 할 수 있게 된다.

WSDL은 표준화된 방식으로 웹 서비스의 인터페이스를 기술하는 XML 기술이다. WSDL은 웹 서비스가 호출된 입출력 변수, 함수구조, 호출의 성질, 서비스 프로토콜 바인딩을 위부에 표시하는 방식을 표준화한다. 또한, 서로 다른 클라이언트가 웹 서비스를 통해 상호작용하는 방식을 파악하게 해준다. WSDL 문서에는 웹 서비스 구현의 세부사항 정보도 포함되어 있다. 따라서 소비자는 WSDL 문서를 통해 웹 서비스의 여러 오퍼레이션과 URL정보를 얻을 수 있다.

UDDI는 웹 서비스를 온라인 디렉토리에 등록, 공개하기 위해 개발된 규약이다. UDDI는 웹 서비스의 전 세계적인 레지스트리를 제공하며 비즈니스, 웹 서비스, 규격 메타데이터, 웹 서비스 접근 위치를 나타내는 구조를 제공한다. 웹 서비스에서는 UDDI 명세를 통해 웹 서비스 정의의 공개(publish)와 발견(discovery)에 대한 사항을 정의하고 있다.

HTTP나 SMTP같은 표준 기반 인터넷 프로토콜을 통해 접근할 수 있다는 점을 고려하면 J2EE, CORBA, CGI 스크립을 웹 서비스 기술로 분류할 수도 있지만 그렇지는 않다. 이러한 기술들과 웹 서비스라고 하는 새로운 기술들과의 가장 큰 차이는 표준화에 있다. SOAP, UDDI, WSDL등의 새로운 기술들은 표준화된 XML을 기반으로 하여 전 세계적으로 주요 기술 회사의 지원을 받는다.

2.3 Apache Axis

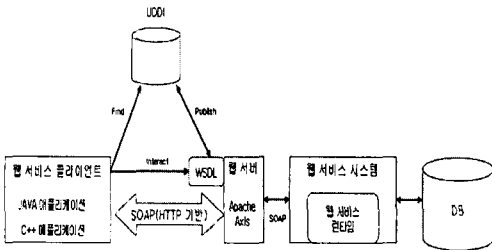
웹 서비스 애플리케이션에 필요한 API를 서버에 설치하기 위해 사용되는 Axis는 아파치의 전 프로젝트인 아파치 SOAP 2.2를 모태로 하여, 보다 나은 유연성 제공을 목표로 개발되었다[6]. Axis는 벤더-중립적(vendor-neutral)으로 제공되며 매크로미디어(Macromedia)와 같은 벤더들이 J2EE 서버(JRun)의 일부로 Axis 구현을 사용한다. Axis는 JAX-RPC의 구현을 제공한다. JAX-RPC는 자바 기반의 웹 서비스 솔루션의

제공자들이 그들의 제품에 틀을 제공하면서 정적인 호출 모델을 만들기 위한 방법을 사용하기 위한 표준 자바 API이다. 또한 Axis는 맞춤화에 있어 엄청난 유연성을 제공하는 확장 가능한 구현을 제공한다[5]. 기존의 아파치 SOAP은 DOM 방식으로 XML 메시지를 파싱하는 반면, AXIS는 SAX방식으로 파싱하여 메시지 처리 속도가 매우 향상되었다. AXIS는 SOAP구현을 재설계한 것으로 메시지를 받아들이는 전송 리스너와 메시지를 처리하는 메시지 핸들러로 이루어져 있다.

3. 시스템 설계 및 구현

3.1 웹 서비스 기반의 시스템 아키텍처

그림 1은 웹 서비스 시스템 아키텍처이다. 웹 서비스 시스템은 J2EE에서 말하는 컨테이너 개념과 유사한 것으로 웹 서비스를 실행하기 위한 런타임 환경을 제공하는 것으로서 SOAP 요청을 받아들이고, 이를 처리할 수 있는 적절한 자바 컴포넌트로 보내 그에 따른 처리 기능을 제공한다. 웹 서버는 SOAP 클라이언트의 요청을 서비스 제공자에게 전달하는 중요한 게이트웨이이며 WSDL 문서가 위치하는 곳이다. 웹 서비스 클라이언트는 웹 서비스의 소비자로서 플랫폼 독립적이기 때문에 어떤 언어로 작성된 클라이언트도 웹 서비스를 호출할 수 있다. 또한, UDDI에서 원하는 서비스의 정보를 참조하는데, 보통 UDDI에서 제시된 URL을 통해 WSDL 문서를 받아오며 이는 해당 웹 서비스의 웹 서버를 거쳐게



<그림 1> 웹 서비스 시스템 아키텍처

된다.

3.2 Tomcat과 Axis의 연동

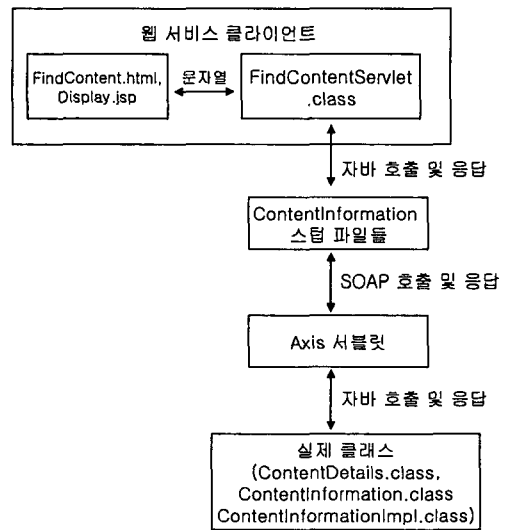
SOAP 엔진은 서버 쪽과 클라이언트 쪽의 SOAP 애플리케이션의 개발을 도와주는 자바 클래스들의 집합으로 본 시스템에서는 Axis.1-1과 쉐인 마이크로 시스템즈의 JWSDP(Java Web Service Developer Pack) 1.3을 사용하였다. JWSDP는 WSDL에서 정의하고 있는 타입을 자

바 타입에 대응하는 내용을 정의하고 있다.

HTTP 상의 원격 요청에 서비스를 하기 위해 웹 서버가 필요하다. 본 논문에서는 Axis에서 제공하는 SimplAxisServer 대신에 보다 강건하고 보안성, 확장성이 보장되는 Tomcat을 웹 서버로 사용하였다.

Axis의 디렉토리 중에 webapps 디렉토리 아래의 axis 폴더를 Tomcat이 설치된 디렉토리의 webapps에 복사만 하면 연동 준비는 끝난다. 중요 jar파일들은 Tomcat 디렉토리의 common/lib 디렉토리에 복사하면 될 것이다[9].

3.3 검색 시스템 구현



<그림 2> 검색 시스템 아키텍처

본 검색 시스템은 윈도우 2000pro, JSDK1.4.2, Tomcat1.4x, Axis 1-1버전 환경에서 구현되었다. 데이터 베이스는 오라클 9i release2를 사용하였다.

ContentInformation 클래스는 영상정보 검색을 위한 웹 서비스의 원격 인터페이스이다. ContentInformationImpl은 웹 서비스의 구현 클래스이며 FindContentServlet은 FindContent에서 요청한 값을 얻기 위해 ContentInformation 웹 서비스에 접근하기 위한 클래스이다. findcontent.html은 원하는 영상정보 데이터를 얻기 위한 폼을 제공하며 Display.jsp는 ContentInformation 웹 서비스가 반환한 값을 화면에 보여준다.

웹 서비스를 구현하기 위해서 먼저 자바 파일들을 컴파일시키고, 웹 서비스에 배치시키기 위해 WSDD(Web Service Deployment Descriptor) 파일을 생성한다. WSDD는 해당 웹 서비스의 배치에 관련된 세부적인 속성을 정의한 XML 문서로써, Axis는 WSDD 파일을 사용하여 웹 서비스를 배치한다. 배치가 성공적으로 끝나면 Axis의 메인 화면에서 생성된

WSDL문을 확인 할 수 있다.

웹 서비스 배포자가 자신의 웹 서비스를 XML 레지스트리에 등록할 때, 그 웹 서비스는 서비스의 인터페이스나 WSDL 정의를 같이 제공해야 한다. 이를 위해 JAX-RPC의 표준에서는 이러한 자바 타입과 WSDL에서 정의하고 있는 타입을 서로 대응하는 내용을 정의 하고 있다. 따라서 클라이언트가 XML 레지스트리에 접근하여 서비스를 받으려고 할 때, 실제로 클라이언트는 서비스로부터 WSDL 문서를 다운로드 받는다. 이것을 자바에서 사용하기 위해서 클라이언트는 동적이든 정적이든 적절한 자바 타입으로 변경해야 한다[6]. 이를 위해 본 논문에서는 JWSDP 1.3 버전의 wscompile을 사용하였다. wscompile는 JWSDP 1.1 에서는 xrpc로 제공되었으며, 여러 가지 옵션을 통해서 스텝과 타이, WSDL 문서 그리고 배치 중에 사용할 서버 환경 설정 파일을 자동 생성할 수 있다. 또한 이미 존재하는 WSDL 문서를 이용하여 웹 서비스에 접근 할 수 있다[7].

And now... Some Services

- AdminService (wsdl)
 - AdminService
- Version (wsdl)
 - getVersion
- ContentInformation (wsdl)
 - finalize
 - getContentDetails

<그림 3> 웹 서비스 배치

Content Finder Results



<그림 4> 영상정보 검색 화면

FindContent.html을 통해서 입력 받은 값은 FindContentServlet 클래스에 전달 되고 FindContentServlet은 JWSD의 wscompile을 사용하여 생성된 stub파일들을 참조하여 ContentInformation과 ContentInformationImpl을 통해 받은 값을 반환하여 Display.jsp를 통해 값을 보여주게 된다.

4. 결론

웹 서비스의 정의는 보는 시각에 따라 다양하다. IBM의 경우는 '표준화된 XML 메시지를 사용하여 네트워크를 통해 접근할 수 있는 프로그램의 모음을 기술하는 인터페이스'라 하였고 마이크로소프트의 경우에는 '인트라넷이나 인터넷에서 문서를 쉽게 공유할 수 있으며, 확장성 있고 신뢰성 있는 웹 기반 응용 프로그램을 배포할 수 있으며, 레거시 데이터와 응용 프

로그를 웹에서 제공하고 회사에서 내부 및 외부 비즈니스 프로세스를 재정의할 수 있도록 한다.' 라고 정의 하였다[8,10]. 결국 웹 서비스는 산재해 있는 많은 기술을 표준화 하여 상호 운용성을 극대화 하려는 개념임을 알 수 있다.

최근 웹상에서 제공되는 영상정보의 양이 급증하고 이용자들의 문화적인 수요가 높아진 것에 비해 이를 검색하는 시스템은 부족한 실정이다. 또한 네트워크의 속도와 하드웨어 소프트웨어는 하루가 다르게 발전하고 있다. 새로운 컴퓨팅 패러다임에 맞추어 영상정보 검색도 기존의 웹상에서 단순 HTTP 프로토콜과 프로그램언어 기반뿐만 아니라 핸드폰, PDA와 같은 이기종간에서의 정보교류를 통한 검색도 필요하게 되었다. 이를 위해서 본 논문에서는 웹 서비스를 기반으로 한 영화 콘텐츠 검색 시스템을 설계 및 구현 하였다.

참고문헌

- [1] 고영만, 김소형, "영상정보 메타데이터 요소 표준 개발을 위한 연구", 성균관 대학교 문헌정보학과, 2003년 12월 9일.
- [2] 박선영, 용환승, "XML을 이용한 내용기반 이미지 데이터베이스의 설계 및 검색 시스템 구현," 한국정보과학회 논문지, 제27권 4호, 2000년 12월, pp. 572-584.
- [3] 김상욱 외, "멀티미디어 데이터베이스", 정보과학회지, 제14권, 제9호, pp.31-42, 1996년 9월.
- [4] 문창모, "웹 서비스 적용 방안", 한국정보처리학회, 9권 4호, 2002.
- [5] 앙리 베케 외 3인 공저, "Begging 자바 웹 서비스", 정보문화사
- [6] Gadzik, Lomei, Santry, Zang "JAVA Web Services for Experienced Programmers", DEITE
- [7] Yawns.com, [www.theyawns.com/Moving%20from%20xrpc%20to%20wscompile.doc]
- [8] Sun,[http://java.sun.com/webservices/docs/eal/tutorial/index.html]
- [9] Web Services - Axis, [http://ws.apache.org/axis/]
- [10] ARINDNE, [http://www.ariadne.ac.uk/issue29/gardner/]