웹 構造를 利用한 웹 質疑 시스템의 設計 및 具現

고성준,김상석,김응모
성균관대학교 전기전자 컴퓨터공학부
e-mail:gozila@ece.skku.ac.kr

Design and implementation of Web Query System Using Web Structure

Sung-Jun Ko, Sang-Suk Kim , Ung-Mo Kim
Electriacl & Computer Engineering, Sung-Kyun-Kwan University

요약

사용자 질의를 통한 검색엔진의 도움 없이 웹 데이터베이스의 개념을 도입하여, 웹을 직접 검색할 수 있는 웹 질의 엔진(Web Query Engine)을 제안한다. 특히 사용자가 자신이 원하는 질의를 할 수 있도록 기존의 질의 언어와 유사한 웹 질의 언어를 설계하고, 설계되어진 질의 언어를 사용하여 인덱스 서버에 정보의 요청 없이 네트워크 구조와 위상 기반의 질의를 할 수 있도록 하는 웹 질의 엔진을 설계 및 구현을 하였다.

1. 서론

최근 사용되고 있는 인터넷 정보 검색을 위한 방법은 크게 browsing 과 searching 이라는 두 가지 형태로 나누어 볼 수 있으며, Browsing의 경우 네트워크를 항해하는 부담을 사용자에게 요구하고 있으며, 사용자는 다음과 같은 일을 처리해야 한다.

● 사용자 스스로 자신이 원하는 정보를 얻기 위해 거대한 정보 공간을 찾아 다녀야 한다.
● 검색된 결과를 직접 구성하여 처리하여야 한다.
● 네트워크에 관한 많은 지식이 있어야 한다.
● 검색하고자 하는 정보에 대한 구체적인 명세를 알고 있어야 한다.

Searching의 경우 가장 널리 알려진 서비스로는 서치엔진이나 Index Robot를 사용하는 서비스와 사람이 직접 구성한 목록을 가지고 있는 두 가지로 나누어 볼 수 있다. 그러나 웹 검색엔진의 공통적인 문제점은 때때로 사용자의 기호에 부합되지 못하는 문서를 검색한다는 것과 다양한 검색엔진이 존재하기 때문에 사용자에게 혼란을 가중시킨다는 것이다. 이러한 문제점이 발생하는 이유는 다음과 같다.

● 다양한 종류의 분류가 있기 때문에 사용자는 자신이 원하는 정보를 얻기 위해 사용해야 하는 분류를 찾기 어렵다.
● 중앙 집중적인 인덱싱이기 때문에 정보 제공자의 정보가 바뀌었을 경우 갱신되는 것이 어렵다는 것이다.
● 복합적인 질의에 대한 병렬적인 검색이 어렵다는 것이다.

이러한 문제점을 해결하기 위해 본 논문에서는 사용자 질의를 통한 검색엔진의 도움 없이 웹 데이터베이스의 개념을 도입하여, 웹을 직접 검색할 수 있는 웹 질의 엔진(Web Query Engine)을 제안한다. 특히 본 논문에서는 사용자가 자신이 원하는 질의를 할 수 있도록 기존의 질의 언어와 유사한 웹 질의 언어를 설계하고, 설계되어진 질의 언어를 사용하여 인덱스 서버에 정보의 요청 없이 네트워크 구조와 위상 기반의 질의를 할 수 있도록 하는 웹 질의 엔진을 설계 및 구현을 하였다.

3. 웹에 쓰이는 데이터베이스 기술

데이터베이스 기술이 많은 양의 데이터를 다루는 가장 기초적인 기술이지만, 웹의 새로운 등장은 이런 기존의 방식보다 새로운 기술을 요하게 되었다.

웹 정보의 관리와 관련된 작업을 나누어 보면 다음과 같다.

● Modeling and querying the web

웹을 방향성을 가진 그래프로 가정하면, 노드는 웹 페이지를 나타내고 에지는 페이지들간의 링크를 나타낸다. 웹의 특정 페이지에서 정보를 뽑아내는데 있어서 우선적으로 고려해야 할 것은 명확한 질의어를 생성하는 것이며, 질의어는 질의되는 웹 페이지들의 내용과 그 페이지들 사이의 링크 구조에 따라 만들어져야 한다.

● Information extraction and integration

웹사이트들은 단순하게 구조적 데이터를 저장하는 장소로서의 기능을 하는 페이지라기보다는 계층이 있는 (예를 들면, 계층별 튜플들의 집합 또는 객체의 집합) 그래프로서 보여질 수 있다.

● Web site construction and restructuring

웹사이트들은 데이터베이스 또는 구조화된 파일에 저장된 - 가공되지 않은 - 정보에서, 또는 이미 존재하는 웹사이트들을 재구성함으로써 만들어진다.

3.1 웹 데이터베이스를 위한 데이터 표현

3.1.1 그래프 데이터 모델

데이터들을 모델링하는 자연스러운 방법으로는 라벨이 있는 그래프 데이터 모델에 기초하여 모델링을 하는 것이며, 노드(Node)들은 웹 페이지들(또는 웹 페이지의 내부 요소들)을 표현하고, 아크(Arc)들은 페이지들 사이의 링크들을 표현한다. 아크의 라벨(Label)들은 애트리뷰트(Attribute)의 이름이며, 라벨이 있는 그래프 모델(Labeled Graph Model)을 기반으로 여러 질의어들이 개발되었다. 이러한 질의어들에게 있어서 공통적으로 나타나는 한가지 중요한 특징은 그래프를 질의할 수 있는 정규패스표현(Regular Path Expression)을 가진 질의어들을 생성할 수 있다는 것이다. 따라서 정규 패스 표현식을 가진 질의어는 그래프 구조에 대해 탐색이 가능하다.

3.1.2 반 구조적인 데이터 모델

반-구조적 데이터를 위한 모델은 라벨이 있는 방향 그래프(Labeled Direct Tree)를 기초로 하고 있으며[Abi97, Bun97], 반-구조적 데이터 모델에서는 그래프 내부의 노드에서 나오는 아크들의 집합과, 애트리뷰트 값의 타입에 제한이 없다.

3.1.3 웹 데이터 모델의 다른 특성

웹 데이터베이스 응용 분야에서 사용되어지는 모델의 다른 구별되어지는 특성은 데이터를 표현할 때 독특한 웹 구조를 가진다는 것이다. 예를 들면, 몇몇 모델들은 페이지들을 표시하는 단일 관계와 페이지들 사이의 이진 링크 관계를 구분하며, 뿐만 아니라 하나의 웹 사이트 내부에 있는 링크와 외부 링크를 구별한다는 것이다.

웹 응용분야에서 데이터를 질의하는 언어의 중요한 관점은 질의어에 대한 결과로서 복잡한 구조를 만들어질 수 있다는 것이다.

3.2 웹 모델링 및 질의

웹을 거대한 그래프 형태의 데이터베이스로 가정하면, 웹을 질의 할 수 있는 질의어는 현재의 검색엔진과 같이 웹 페이지의 내부 구조와 웹 페이지들을 연결하는 링크의 외부 구조로서 표현되고, 그 구조에 의해서 지원되는 기본적인 정보 검색 시스템을 훨씬 능가하는 질의어를 구현하는 것이 당연하다.

3.2.1 구조 정보 추출

● WebSQL

WebSQL은 웹을 Document와 Anchor의 2 개의 관계를 가지는 관계 데이터베이스로 모델을 제안한다. Document Relation은 웹 상의 문서마다 하나의 튜플을 가지고 있으며, Anchor Relation 역시 하나의 튜플을 가지고 있다. 이러한 웹의 추상화는 웹을 SQL 과 유사한 질의 언어를 사용하여 질의 할 수 있도록 하는 장점이 있다.

만약 Document와 Anchor가 실제 관계형이라면 그들을 질의하기 위해서 SQL을 사용할 수 있겠지만, 실제 이들은 가상의 테이블로서 구성되어 그들을 질의하기 위해서는 SQL과는 다른 새로운 질의 언어의 정의가 필요하다. 이러한 가상의 테이블을 질의하기 위한 언어를 WebSQL로 정의하고 웹을 질의하기 위해서 이 언어를 사용한다. WebSQL은 Path Regular Expression을 사용하며, 이러한 표현식을 통해서 웹의 구조를 탐색할 수 있는 언어이다.

● WebOQL

WebSQL에서 사용되어지는 주요 데이터 구조는 하이퍼트리(Hypertree)이며, 하이퍼트리는 내부(internal) 아크(arc)와 외부(external) 아크(arc)를 가지는 순서화된 아크와 레이블(ordered arc-labeled)을 가지는 트리이다. 내부 아크는 구조화된 오브젝트를 표현하기 위해 사용되어지고, 외부 아크는 오브젝트들 사이의 참조(references)를 나타내기 위해 사용되어진다.

4. 웹 질의 엔진

웹 질의 엔진(Web Query Engine)은 웹 기반의 환경에서 구성되어 있는 분산된 이 기종의 데이터베이스와 구조적

인 문서 그리고 웹 상에 산재해 있는 웹 페이지들을 웹 페이지를 기존의 검색 방식인 인덱스 서버를 통하지 않고 직접 검색하거나 사용자가 원하는 정보를 기존의 검색 엔진을 통해서도 검색을 할 수 있는 통합된 검색 방식을 제안한다.

이러한 기능을 수행하는 엔진을 개발하기 위해서는 우선되어져야 할 것은 첫째, 사용자가 질의하고자하는 웹 환경을 모델링(Modeling)하고, 둘째, 이러한 웹 환경에 적합하게 작성되어진 질의 언어(Query Language)와 이 질의 언어를 분석하는 분석기(Parser)가 필요하다. 마지막으로 분석기로 분석된 정보들을 이용하여 실제로 작업을 수행하도록 하는 실행 엔진의 설계 및 구현이 요구되어진다.

4.1 웹 질의 언어

4.1.1 데이터 모델

Infinite set D : 데이터 값의 무한 집합
Finite set T : 타입들의 집합

으로 가정하면,

Tuple types[a1:t1,...,a_n:t_n] (attributes a_i을 가지는 simple type t_i,i = 1...n)

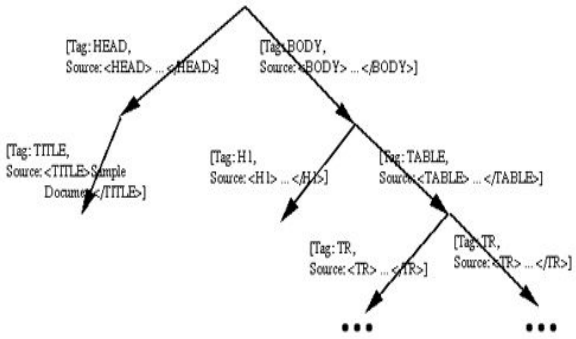
Tuple x에서 x.a_i 는 attribute a_i 과 연관되어지는 value v_i 로 표시한다.

Simple type Oid ∈ T 는 Object Identifier 이고 Node 와 Arc 2개 의 Tuple 타입,

Node = [id:Oid,...,a_i:t_i,...]
Arc = [from:Oid,to:Oid,...b_j:t_j,...]
를 가지고 있고, 첫 번째 tuples을 Node objects 로 두 번째 tuples을 Arc object로 부른다. 우리가 설계하는 모델에서는 웹 문서는 Node objects로 웹 문서들 사이의 하이퍼덱스트 링크들은 Arc Objects로 사상한다.

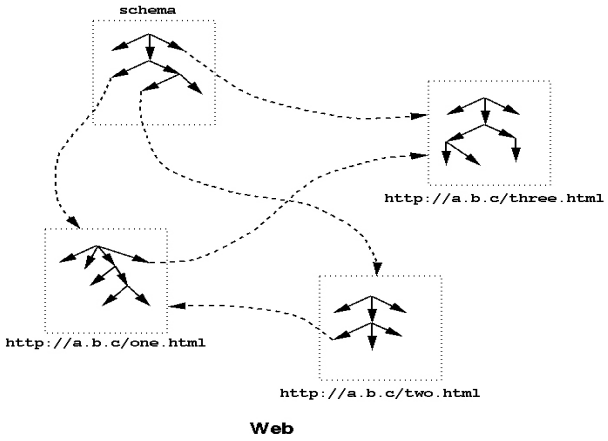
4.1.2 하이퍼트리 모델

Web Query Engine을 위한 스키마 구조는 하이퍼트리(Hyper-tree)를 이용한다. 하이퍼트리는 내부 링크(internal link) 와 외부 링크(external link)로 구성되는 계층적인 링크-레이블(link-label) 구조이다. 여기서 내부 링크는 문서 객체의 구조를 표현하며, 외부 링크는 이러한 객체들간의 참조 관계를 표현하며, 각 링크는 레코드들로 레이블화 되어 표현된다.



[그림 4-1]HTML 문서의 Hypertree 표현

[그림 4-1]은 HTML 문서의 구조를 하이퍼트리 형태로 표현한 것이다. 일반적으로 웹은 위와 같은 HTML 페이지들이 서로 하이퍼링크(Hyperlink)로서 연결되어 있는 형태를 가지고 있으며, 이는 [그림 4-3]과 같이 표현할 수 있다. 이와 같이 웹 상의 여러 페이지들은 외부 링크들로 구성된 하이퍼링크를 통해서 서로 연결되어 있고, 이 연결을 통해서 기존의 키워드 검색 및 웹의 구조를 이용한 여러 유형의 검색이 가능해 진다.



[그림 4-2]Web의 Hypertree 표현

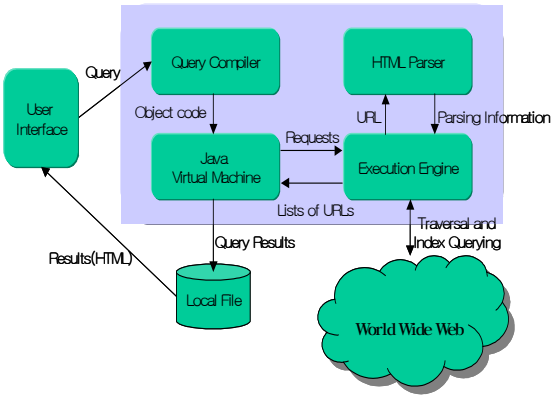
위와 같이 스키마 구조를 정의한 후, Web Query의 구조 및 기능을 설계 및 구현한다. 이를 위해 최근 발표된 Web Query 언어인 WebSQL[3,4,9] 및 WebOQL[7,10]에서 제공되는 기능 및 구문 구조를 분석하여, 이들을 본 시스템에서 제공되는 페이지 합성(restructuring) 및 동적 템플릿에 맞도록 새로운 BNF를 생성한다.

[그림 4-3]웹 질의 언어의 BNF

4.2 웹 질의 엔진의 구조

```
QueryList ::= ( Query ";" )* <EOF>
Query ::= <SELECT> Field <FROM> Location <WHERE> Condition
Field ::= <FIELD_VAR>
Location ::= ( <HTTP> )? <URL> | <INDEX_SERVER>
Condition ::= Boolterm ( <OR> Boolterm )*
Boolterm ::= Bool ( <AND> Bool )*
Bool ::= ( <KEY> <IS> Key )
        | ( <DEPTH> <EQUAL> Depth )
        | ( <TAG> <IS> Tag )
        | ( <TUPLE> <EQUAL> Tuple )
Key ::= <STR>
Depth ::= <NUM>
Tag ::= <STR>
Tuple ::= <NUM>
```

웹 질의 엔진이 동작하는 환경은 그림에서 보이는 것과 같이 핵심 부분인 (1)웹 질의 분석기(Web query Compiler), (2)HTML 분석기(HTML parser), (3)자바 가상 머신(Java Virtual Machine), (4)실행 엔진(Execution Engine)으로 구성되며, 실행 엔진은 또 다시 실행부와 Web 환경과 연동 부분인 Mediator, Wrapper로 구성되어 있다.



[그림 4-4]웹 질의 엔진의 구성도

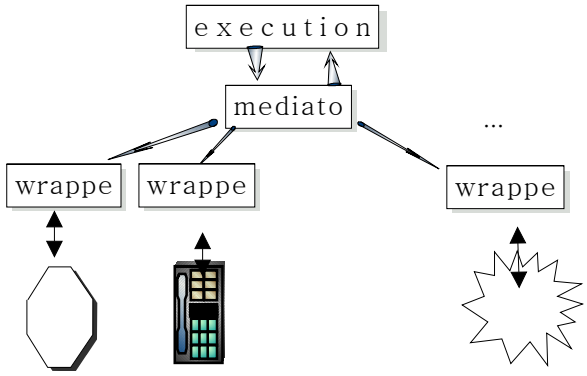
4.2.1 웹 질의 엔진의 세부 구조

● 웹 질의 분석기(Web query Compiler)
웹 질의 분석기는 웹을 검색하기 위해 사용자가 질의한 질의 언어를 컴파일하고 컴파일된 오브젝트 코드를 자바 가상 머신에 전달하는 역할을 한다.

● HTML 분석기 (HTML parser)
실행 엔진으로부터 요청된 URL을 분석하여 HTML 정보를 실행 엔진으로 전달하는 역할을 한다.

● 자바 가상 머신(Java Virtual Machine)
웹 질의 분석기에서 오브젝트 코드를 전달받아 실행계획을 세우고, 실행 엔진에게 실행을 요청한다. 웹 검색이 완료되어지면, 실행 엔진으로부터 검색 결과로 URL의 리스트를 받아 사용자에게 전달한다.

● 실행 엔진(Execution Engine)
실행 엔진은 웹 질의 엔진의 핵심이 되는 부분으로 실제 동작에 관한 전반적이 작업이 수행되어지는 부분으로 아래의 [그림 4-5]와 같이 세부적으로 나누어 볼 수 있다.



[그림 4-5]실행 엔진의 세부 구성도

실행 엔진은 세부적으로 실행부와 중재자인 Mediator 부분, 검색하고자하는 웹 소스에 따라 다르게 적용되는 Wrapper 부분으로 나누어진다. 실행부에서는 사용자의 질의를 수행하기 위해 구현되어진 부분이며, 웹 사용자가 질의를 할 수 있는 대상으로는 다른 사용자의 웹 페이지, 인덱스 서버, 데이터베이스 서버 등 다양한 소스(source)들이 있다. 이렇게 다양한 소스에서 질의를 수행하기 위해서는 각 소스마다 고유한 처리를 해주기 위한 기능이 필요하다.

4.3 웹 질의 엔진의 구현

4.3.1 구현에 사용되어지는 기술

● Java Servlet: Web 과 Database와의 연동기술

4.3.2 세부 모듈 구현

(1) 웹 질의 분석기(web query Compiler)

▶ 질의 분석기의 데이터 타입 선언 및 메인 함수 실행부
질의 분석기에서 사용되어지는 데이터 구조를 선언하고 메인 함수와 함수의 실행부를 작성하는 부분

▶ 질의 분석기의 토큰 정의 선언부
질의 언어의 구문을 나눌 수 있도록 각각의 의미가 있는 구분을 미리 선언하는 부분

▶ 질의 분석기의 실행부
질의를 토큰 정의 선언부에서 선언 된 토큰들로 나누는

것을 실제로 수행하는 부분

(2) HTML 분석기(parser)

HTML Parser 역시 위와 같은 방식으로 작성되며, 우선 HTMLParser.jj file을 작성하고, 이렇게 작성되어진 HTMLParser.jj file을 JavaCC로 컴파일을 하게되면 Java source file 들이 생성된다. 이렇게 생성되어진 Java source 파일들을 Java JDK로 컴파일 하게되면 HTML 분석기의 실행 가능한 오브젝트 코드가 생성된다.

(3)실행 엔진(Execution Engine)

실행 엔진 부분은 사용자에 의해서 질의되어진 문장을 웹 질의 분석기가 분석을 하고, 얻어진 정보를 가지고 자바 가상 머신에서 실행 계획을 세워 Mediator와 Wrapper를 통해 수행되어진다. 이러한 작업을 웹 상에서 구현하기 위해서 웹과 연동 가능한 프로그래밍이 이루어져야 하며, 이를 해결하기 위한 방법으로 Java 의 웹 서버를 위한 기술인 Java Servlet을 사용하여 구현을 한다.

[알고리즘1 - 웹 문서 트리 저장]

이 알고리즘은 주어진 URL에서 사용자가 질의한 레벨만큼의 문서들을 트리로 구성하고, 임의의 저장장소에 저장하여 차후에 조건 검색을 위한 정보로서 사용되어지도록 작성한 알고리즘이다.

curLevel: 현재 링크의 레벨
Level: 문서 검색 레벨
URL: 웹 URL(Uniform Resource Locator)
file: URL로부터 얻어온 문서의 내용을 저장
parse_information[index]: 파싱 정보 저장소
index: 배열의 인덱스

```
-------------------------------------------------
-----------------------
index = 0;
while(curLevel < Level) do
{
        file = get_target_url(URL);
        insert_document_depositary(file);
        parse_information = parse_HTML(file);
        for(i=0;parse_information[i]!=null;i++) do
        {
                file                                    =
get_target_url(parse_information[i]);
                insert_document_depositary(file);

insert_anchor_depositary(parse_information[i]);
        }
        URL = select_anchor_url(index);
        index++;
        curLevel++;
}
```

[알고리즘 1 끝]

(4) 사용자 인터페이스(User Interface)

입력 형식은 대부분의 일반적이 웹 질의를 더 쉬운 형태의 질의로 만들어준다. 만약 더 복잡한 질의를 원한다면 추가적인 텍스트 필드를 추가할 수 있다. 질의를 입력한 후에 질의 버튼을 클릭하게 되면, 질의는 질의 분석기로 보내지고, 구문을 분석하고, 오브젝트 코드를 생성한다. 이 오브젝트 코드는 실행 엔진에서 실행되어진다.
결과는 HTML 문서 형태로 사용자 브라우저로 보여지게되며, HTML 문서의 URL의 링크 형태로 관련된 사이트로 이동하기 쉽게 구성되어진다.

4.4 웹 질의 엔진의 동작과정

[단계 1] 사용자는 웹 브라우저를 통해 질의를 한다.
[단계 2] 질의문을 질의 분석기에서 분석한다.
[단계 3] 질의 분석 테이블에서 검색이 시작될 위치(URL)를 얻어온다.
[단계 4] 얻어진 URL(Uniform Resource Locator)을 네트워크를 통해서 문서를 얻어온다,
[단계 5-1] 요구한 내용이 문서에 존재하면 문서를 데이터베이스에 저장한다.
[단계 5-2] 요구한 내용이 문서에 존재하지 않으면 문서를 데이터베이스에 저장하지 않는다.
[단계 6] 얻어온 HTML 페이지를 HTML 분석기로 보내 파싱을 한다.
[단계 7] HTML 문서의 파싱 정보를 데이터베이스에 저장한다.
[단계 8] 검색할 페이지를 현재 페이지의 하위 링크에 이동시킨다. 하위링크는 저장된 HTML 페이지의 정보를 이용하여 얻는다.
[단계 9] 질의한 링크 레벨에 이를 때까지 4에서 7을 반복한다.
[단계 10] 결과를 웹 페이지에 출력한다.

4.5 웹 질의 엔진의 검색 이외의 기능

(1) Dead link Detection
인덱스 서버 혹은 검색 엔진을 통한 검색 시, 일반적으로 검색 결과 중 다수의 웹 페이지들의 링크(link)의 값이 변경되었거나, 소멸되어 있는 경우가 많이 발생하게 된다.

(2) Link Level Search

Web Query Engine은 사용자가 웹의 구조를 통해 지정한 위치에서 링크의 Depth level을 가지고 검색할 수 있는 기능을 제공한다.

(3) Web Page Restructuring

Web Query의 요청에 의하여 웹 페이지의 일부분을 새로운 페이지로 생성할 수 있으며, 여러 페이지에서 사용자가 원하는 부분을 추출하여 새로운 페이지를 재구성하는 기능을 제공한다.

6. 결론

본 논문에서는 웹 기반의 환경에서 사용자 질의를 통해 웹을 검색할 수 있도록 질의 언어를 설계하고, 이 질의 언어를 바탕으로 질의 엔진 시스템을 제시하였으며, 기존의 데이터베이스 검색 방식인 내용 기반(content based)의 검색을 대체 할 수 있는 기능을 제안하였다.

제안한 질의 엔진은 사용자(End user)를 위한 응용 프로그램이기보다는 또 다른 시스템을 구성할 수 있는 미들웨어로서 사용될 수 있으며, 웹 기반의 응용 프로그램을 좀 더 빠르고 편리하게 사용하도록 돕기 위한 것이다.

향후과제로는 발전하는 웹 환경에 맞추어 좀 더 향상된 시스템을 구성하기 위해, 차세대 웹 문서 포맷인 XML(eXtensible Makeup Language)을 처리할 수 있고, 웹 문서를 구조적으로 질의를 할 수 있도록 DOM(Document Object Model)을 이용하는 웹 질의 엔진에 관한 연구가 필요할 것이다. 또한 웹 환경에서 멀티미디어를 통한 정보의 제공 및 이동이 자유롭고, 대부분의 웹 브라우저들이 멀티미디어를 지원하므로 이에 상응하는 멀티미디어 데이터를 질의할 수 있는 질의 언어의 설계 및 멀티미디어 데이터를 저장하고 질의할 수 있는 시스템에 대한 연구를 해야 할 것이다.

참고문헌

[1] 이경미, 데이터 속성 집합에 기반한 반 구조적 데이터 집합의 구조화 방법론, 1998

[2] 서동렬 외 4인, Ajax-1: 객체지향 데이터베이스를 이용한 웹 자원 관리 시스템

[3] A. Mendelzon, G. Mihaila and T. Milo, “Querying the World Wide Web”, in Journal of Digital Libraries, 1997.

[4] A. Mendelzon and T. Milo, “Formal Models of the Web”, Proceedings of ACMPODS Conference, Tucson, Arizona, June 1997.

[5] D. Quass, A. Rajaraman, Y. Sagiv, J.D. Ullman, and J. Widom, “Querying Semistructured Heterogeneous Information”, Proceedings of the Fourth International Conference on Deductive and Object-Oriented Databases, p319-344, Singapore, December 1995.

[6] Daniela Florescu, Alon Levy and Alberto Mendelzon, “Database Techniques for the World-Wide Web: A Survey”

[7] G. Arocena, “WebOQL: Exploiting Document Structure in Web Queries”, Master's Thesis, University of Toronto, 1997.

[8] G. Arocena and A. Mendelzon, “Viewing Web Information Systems as Database Applications”, Communications of the ACM, July 1998.

[9] G. Mihaila, “WebSQL-an SQL-like query language for the WWW”, MSc. Thesis, University of Toronto, 1996

[10] Gustavo O. Arocena, and Alberto O. Mendelzon, “WebOQL: Restructuring Documents, Databases, and Webs”, Electronic Edition

[11] J. McHugh, J. Widom, S. Abiteboul, Q. Luo, and A. Rajaraman, “Indexing Semistructured Data”, Technical Report, January 1998.

[12] J. McHugh and J. Widom, “Query Optimization for Semistructured Data”, Technical Report, November 1997.

[13] J. McHugh, S. Abiteboul, R. Goldman, D. Quass and J. Widom, “Lore: A Database Management System for Semistructured Data”, SIGMOD Record, 26(3) September 1997

[14] R. Goldman and J. Widom, “DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases”, Proceedings of the Twenty-Third International Conference on Very Large Data Bases, p436-445, Athens, Greece, August 1997.참고문헌

[1] Roger S. Pressman "Software Engineering A Practlti ners' Approach" 3rd Ed. McGraw Hill