시맨틱 웹을 이용한 웹 변경 탐지 시스템

조 부 현[†] · 민 영 근[†] · 이 복 주[††]

요 약

시맨틱 웹은 정보검색과 웹 기반 시스템 분야의 새로운 추세이다. 본 논문은 시맨틱 웹과 온톨로지를 이용하여 웹 문서의 변경을 자동으로 사용자에게 알려주는 웹 변경 탐지 시스템의 개발에 관한 것이다. 기존의 웹 변경 탐지 시스템은 구문(syntax) 변화 중심의 변경 탐지인 반면 본 시스템은 의미(semantic) 변화 중심의 변경 탐지에 목표를 둔다. 즉 의미에 변화가 있는 경우만 찾아 알려주어 사용자에게 유용한 정보를 제공한다. 또한 특정 도메인에 중심이 된 변경 사항을 가정하여 사용자가 목표 사이트를 일일이 지정하지 않아도 변경 탐지가 가능하게 하였다. 이를 위하여 특정 도메인을 가정한(컴퓨터 관련 인물 정보) 온톨로지를 구축하고 웹 페이지를 이 온톨로지에 따라 변환한 다음 변경 전 페이지와 변경 후 페이지를 비교하는 방법을 사용하였다. 실험 결과는 구문 중심의 변경 탐지에 비해 의미 중심의 변경 탐지가 더 유용함을 보인다.

키워드 : 정보검색, 시맨틱 웹, 온톨로지, 웹 변화 탐지

Web Change Detection System Using the Semantic Web

Boohyun, Cho[†] · Youngkun Min[†] · Bogju Lee[††]

ABSTRACT

The semantic web is an emerging paradigm in the information retrieval and Web-based system. This paper deals with a Web change detection system which employs the semantic web and ontology. While existing Web change detection systems detect the syntactic change, the proposed system focuses on the detection of the semantic change. The system detects the change only when the web has semantic change. To achieve this,the system employs the domain-specific ontology (e.g., computer science professional person information in the paper). The Web pages regarding before and after change are converted according to the ontology. Then the comparison is performed. The experimental result shows the semantic-based change detection is more useful than the syntax-based change detection.

Key Words : Information Retrieval, Semantic Web, Ontology, Web Change Detection

1. 서 론

월드 와일드 웹의 발달로 인하여 개인은 정보 수집에 있어 웹에 많은 부분을 의존하고 있다. 이와 같이 웹을 통한 인터넷 정보 접근은 일반화 되고 있지만 엄청난 정보의 양과 다양성은 효율적인 정보 검색을 방해하는 요인이 되고 있다. 또한, 다수의 임의 사용자를 대상으로 하기 때문에 개인이 일일이 웹 페이지를 방문하여 변경된 새로운 정보를 확인하는 일은 큰 부담이 아닐 수 없다. 또한 만약 사용자가 중요한 정보가 변경된 웹 페이지를 놓쳤을 경우 손실은 매우 클 것이다. 또한 중요한 정보를 제공받기 위해 항상 그 페이지를 검색 해야 하는 불편함을 감수하기엔 부담이 매우 크다. 예를 들어 뉴스 속보 등의 빠른 기사 변화, 인터넷 경매 사이트에서의 특정 품목에 대한 경매 상황, 주식거래에 따른 실시간 시세 등 많은 사용자는 정보의 변화에 민감하게 반응하려 하지만 이에 따르는 부담감은 증가하고 있다.

웹 정보의 변화는 웹 문서의 내용이 변경 되는 경우도 있고 정보가 삭제 또는 추가 되는 등 여러 가지 상황이 발생하게 된다. 이러한 웹 페이지 변경의 자동 탐지에 관한 기존의 연구와 응용이 많이 있었으나[1-6, 8, 10] 기존의 시스템이 탐지 시점을 제대로 잡지 못하고 사용자가 관심 있는 페이지를 일일이 입력해 주어야 하는 번거로움 등 사용에 불편한 점이 많았다. 또한 기존의 시스템은 대상 웹 페이지의 과거와 현재 상태를 구문적으로 비교하여 사용자에게 알려주는 수준에 머물고 있다[4]. 예를 들면 웹 페이지 상의 의미가 있는 정보 변화보다 이미지 크기, 또는 글꼴 폰트의 변화 같은 HTML 문법의 변경 사항을 사용자에게 전달하는 경우가 빈번히 발생하고 있다. 결과적으로 기존의 시스템에서 보고 되는 사항은 사용자 입장에서 그리 중요하지 않은 구문적인 변화에 대한 부분이 많이 차지하고 있다.

본 논문은 정보검색과 웹 기반 시스템 분야의 새로운 추세인 시맨틱 웹 기술을 이용하여 웹 문서의 의미상의 변경을 자동으로 사용자에게 알려주는 시스템을 개발하고자 한

※ 본 연구는 2004년도 단국대학교 대학 연구비로 수행되었음.
† 준 회 원 : 단국대학교 전자컴퓨터공학 석사과정
†† 정 회 원 : 단국대학교 전기전자컴퓨터공학과 교수
논문접수 : 2005년 10월 14일, 심사완료 : 2006년 1월 31일

다. 기존의 시스템의 구문적인 변화 탐지에서 벗어나 의미 중심의 변화 탐지를 목표를 맞춘다면 더욱 유용한 시스템이 될 것이다. 예를 들면 인물의 신상 정보(직장의 이전, 승진, 학회 활동, 새로운 논문 발표)에 변화가 있을 때 자동으로 감지하여 관심 있는 사용자에게 알려주는 시스템을 생각해 볼 수 있다. 이를 위하여 인물 신상 정보를 표준화한 온톨로지를 구축하고 이를 기반으로 하여 두 버전의 문서를 비교하고 결과를 사용자에게 알려주는 시스템을 설계 및 구현한다.

2. 기존의 웹 변경 탐지 방법

기존의 웹 변경 탐지 방법에 대한 연구와 응용을 특성과 역할에 따라 분류하였다. 먼저 각각의 버전간의 변화를 감지하여 찾아내는 Diff가 있고, 변화 감지된 정보를 사용자에게 알려주는 방식이 있다. 또한 시맨틱 웹을 이용하여 변화 탐지를 하는 연구가 있고 마지막으로 에이전트의 자동화된 서비스를 이용하여 변화탐지를 하는 연구가 있다.

첫째, 각각의 버전간의 변화를 감지하여 찾아내는 연구는 HTML 및 XML 문서의 변화 감지를 하는 알고리즘에 대한 기존 연구로서 XyDiff[1]는 두 버전의 XML 문서파일 사이에 발생한 모든 변화를 찾아내는 어플리케이션이다. 제시한 알고리즘은 속도 면에서 매우 능률적이다. 또한, 삽입 외에, 삭제와 업데이트, 그 서브트리에 변화 등을 찾아 낼 수 있다. 현재 Linux 또는 SunOS에서 작동된다. X-diff[2]는 구문 기반인 HTML과 의미 기반인 XML에 대한 문서의 변화를 찾아내는 효율적 알고리즘인 tree-to-tree correction 기술을 이용하여 XML의 구조에 대해서 보다 정확하게 변화를 찾아 낼 수 있다. 한편 HTMLDiff[3]는 HTML 문서 사이의 정보 비교뿐 아니라 태그까지 비교하여 결과를 얻어 낼 수 있다. 사용자가 원하는 시점에서 비교 할 두 가지 문서를 직접 입력하면 바로 결과를 얻어 낼 수 있는 장점이 있는 반면, 사용자가 직접 두 시점의 문서를 입력하여 비교하는 방식으로 인해서 사용자가 문서를 직접 관리하게 되는 번거로움으로 실시간의 비교에는 한계를 갖는다.

둘째, 변화 감지된 정보를 사용자에게 알려주는 방식은 현재 웹 페이지에 대해서 HTML 문서의 변화를 알려 주는 서비스를 운영하는 changedetection.com[4]과 ATS Consulting[5] 그리고 InfoMinder[6] 등이 있다. 위 서비스의 경우 사용자가 직접 받아볼 전자메일과 URL를 서버에 등록 시키면 탐지 간격은 최단기간이 일 단위이며 전자메일로 전송시켜 주는 방식이다. 이 방식은 현재 사용 되는 HTML 페이지 내용 전체를 대상으로 변화를 비교하고 통보하는 방식이다. 일정간격으로 사용자가 선택한 URL 의 페이지를 버전 간의 변화 감지 알고리즘을 통해 분석한다. 이때, 내용이 달라졌으면 변화된 사실을 사용자에게 통보하고 사용자로 하여금 웹 사이트에 재 접속을 하여 달라진 내용을 확인하게 한다. 서버 기반 방식을 취하고 있고 변경 통보 간격이 일 단위로 이루어진다. 또한 InfoMinder[6]의 경우도 HTML 문서의 변화를 알려주는 서비스를 하고 있다. 사용자가 웹사이트를 추적하고, 변경된 내용에 관한 정보를 얻는 것을 가능 하도록 하는 서비스이다. 모니터링 할 HTML 페이지의 URL을 지정하고 변화가 감지될 때 전자메일로 통보해 준다.

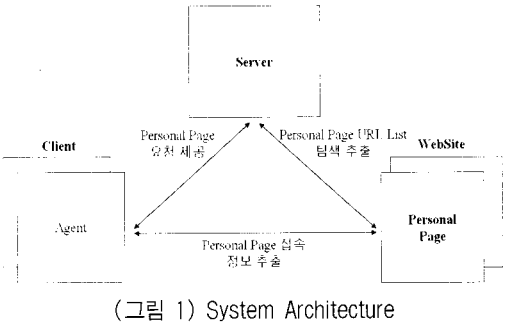
셋째, 시맨틱 웹을 이용하는 연구로서 시맨틱 웹의 비전은 W3C[7]에 의해 공유되고 있고, 아직 수많은 연구가 진행중에 있다. [8]의 논문에서는 웹에서 온톨로지를 효과적으로 이용 하려면, 온톨로지에서 변화를 관리하는 것과 온톨로지와 온라인 온톨로지 인증에 관한 버전들의 연관성을 이용하여 웹의 내용들을 주목함으로써 'ontology versioning'이란 주제를 분석했다. 사용자들의 편의를 위해 온톨로지를 관리하는 웹 기반 시스템의 디자인을 설명하고, 시스템은 온톨로지 사이의 변형뿐만 아니라 다른 버전들이 가진 개념들 사이의 개념적인 관계에 의지 함으로서 웹 기반의 온톨로지의 다른 버전들을 공동 이용이 가능하게 해준다. 차이점을 시각화하기 위해, 시스템은 차이점들을 찾아서 RDF[9] 기반의 온톨로지로 분류하는 융통성 있는 메커니즘을 이용한다.

마지막으로 에이전트를 통하여 사용자에게 자동으로 정보를 제공하는 연구가 있다. 한양 대학교 웹 브라우징 에이전트 IWeBA[10]는 웹 문서에서 새로 추가되는 항목이나 특정 웹 문서의 변화를 사용자에게 자동적으로 알려 주는 기능을 탑재 하고 있다. InfoMinder[6]와 changedetection.com[4] 등의 경우 미묘한 변화에 대해서도 반응하는 단점이 있는 반면 문서 내용의 변화에 중점을 주기 때문에 불필요한 변화에 대해서는 반응을 보이지 않는 특징이 있다.

3. 온톨로지를 이용한 웹 변경 탐지 시스템

(그림 1)은 본 연구에서 수행한 시맨틱 웹을 이용한 의미 중심의 웹 변경 자동 탐지를 위한 시스템 구조를 보이고 있다.

서버는 인물의 신상 정보 사이트의 URL를 찾아내고 이를 기반으로 Web page storage를 구축한다. 이를 통하여 인물정보 사이트의 URL을 제공하여 클라이언트는 보다 쉽고 정확하게 web page를 확보할 수 있다.클라이언트는 온톨로지 서버에 접속하여 인물에 대한 URL를 갱신하고 이를 통하여 인물정보 페이지에 접근하여 필요한 정보를 가져오게 된다. 이 정보(HTML)를 XML로 변화시켜 의미기반의 페이지로 변환하고, 문서간의 비교는 ontology prototype을 기반으로 한 RDF기반에서 의미 중심의 비교를 하게 된다.



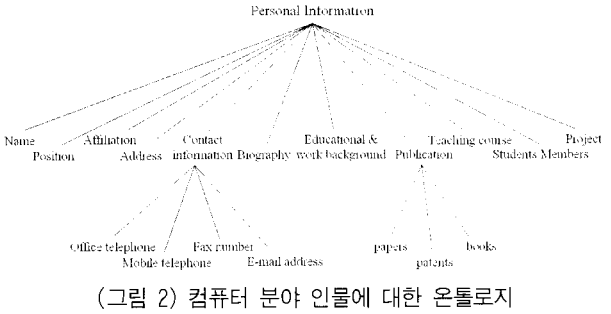
(그림 1) System Architecture

본 연구는 다음의 관점에서 기존 방식과 다르다. 첫째 의미 기반에 의해 변화를 감지한다. 기존의 방식은 HTML 문서 양식에 대한 변화 감지로 서비스를 이루고 있다. 현재 서비스를 제공하고 있는 업체들의 경우 두 버전의 HTML 문서에서 구문적으로 분석하여 변화를 전자 메일로 통보한다. 본 연구는 시맨틱 웹 기반으로 XML 문서 양식을 사용하여 Metadata만 분석함으로써 두 버전의 XML 문서 전체가 아닌 RDF 부분만 변화 감지 알고리즘이 동작하므로 의미 기반의 변화 탐지가 가능하다.

둘째, 서버와 클라이언트의 역할 분담을 통해 변화 감지 간격을 단축시킬 수 있다. 기존의 방식은 서버 측에서 변화를 감지하여, 클라이언트에게 전자메일로 전송하는 방식이다. 클라이언트의 수가 증가함에 따라 서버에서 각각의 클라이언트 별로 변화 감지 알고리즘이 동작하기 때문에 서버의 부담이 증가한다. 따라서 변화를 감지하여 클라이언트에게 전송하는 간격이 증가하게 된다. 본 연구는 기존의 서버에서 동작한 변화감지 알고리즘을 클라이언트의 에이전트에 이식하여 변화 감지 간격을 최소화 한다.

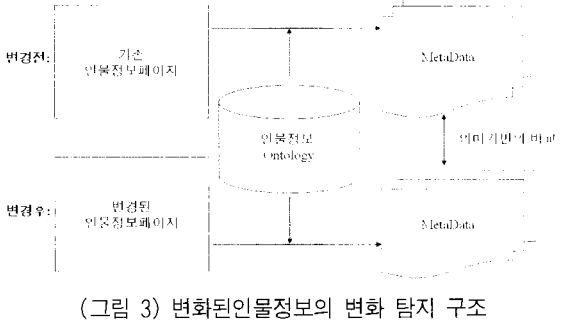
4. 본 시스템 세부 구조

(그림 2)은 인물에 대한 정보 도메인을 가정한 온톨로지 prototype이다.

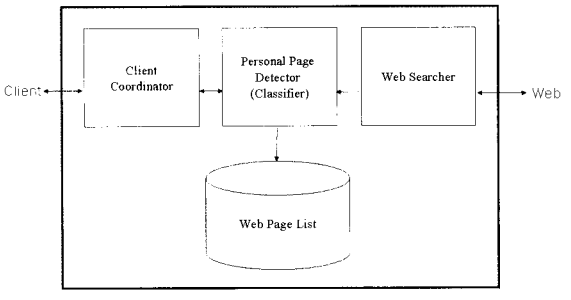


(그림 2) 컴퓨터 분야 인물에 대한 온톨로지

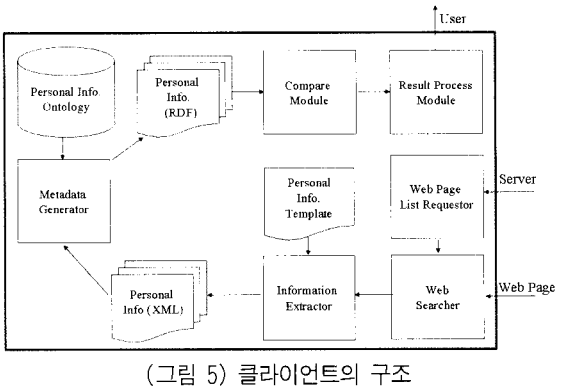
이 prototype을 기반으로 한 XML template을 두어 웹 사이트로부터 받아온 정보를 XML 문서로 변환하여 저장하게 된다. 이 때 버전 별로 구분을 두어 저장하게 되며 이 문서들은 변경 탐지를 위하여 N-Triple 형식의 RDF 문서로 변환된다.



(그림 3) 변화된인물정보의 변화 탐지 구조



(그림 4) 서버의 구조



(그림 5) 클라이언트의 구조

유사어 관계 표현 온톨로지
(Personal Data Synonym Ontology)
- 이름 = 성명 = Name ...
- 근무처 = 회사 = Position ...
- 우편번호 = Zipcode = Post number
- 주소 = 근무지 = Address ...
- 이메일 = 전자메일 = email = e-mail = electronic mail...
- 전화 = 전화번호 = 연락처 = Phone = Telephone = Tel...
- 휴대폰 = 휴대번호 = 휴대전화 = 이동전화 = cellular phone = cell phone = cell...
- ETC..

(그림 6) 유사어 관계 표현 온톨로지

(그림 3)과 같이 두 버전간의 변화를 탐지하는 방식은 다음과 같다. HTML로 되어 있는 페이지를 인물정보 ontology를 이용하여 metadata로 변환한다. 이 정보는 XML형식의 RDF문서로 변환되게되며, 기존에 저장되어 있는 XML형식의 RDF 문서와 의미기반의 변화를 비교 하게 된다. 자세한 알고리즘은 아래 클라이언트의 구조 부분에 기술되어 있다. 이를 통하여 기존 변화감지의 문제점인 정보와 관계없는 변화를 제거할 수 있다.

위의 (그림 4)와 같이 서버의 구조는 크게 세 가지의 모듈로 이루어지게 된다. Web Searcher는 Web에서 무작위로 탐색하여 페이지의 정보를 Personal Page Detector에게 제공하게 되고, 이 모듈은 web personal page를 분류 및 추출하여 Web Page List(Storage)에 저장하게 된다. 클라이언트의 요청에 의해 Client Coordinator는 storage 에서 변화 감지 대상 페이지의 URL를 제공하게 된다.

(그림 5)는 클라이언트 구조를 보여주고 있다. Web Searcher는 server에 접속하여 Client별로 인물정보 페이지 리스트를 가져오고 인물정보 페이지 주소(URL)와 비교하여 갱신한다. 이 갱신된 주소를 이용하여 HTTP 접속을 하고

페이지의 정보(HTML)를 추출하여 Information Extractor에 제공해 준다. Information Extractor 모듈은 아래 (그림 6)과 같은 유의어 온톨로지를 참고로 하여 원하는 property와 value를 쌍으로 추출하고, 이 정보를 XML template를 참조하여 XML 파일을 작성하여 저장한다. Metadata Generator 모듈은 인물정보에 대한 온톨로지를 기반으로 XML 파일을 RDF data로 생성을 하게 된다. RDF data는 버전 별로 생성되어 Compare Module을 통하여 이 버전들간의 Metadata를 뽑아내어 비교하게 된다. Diff 알고리즘을이용하여 찾아낸 변화를Result Process Module을 통하여 사용자에게 알리게 된다.

5. 시스템 구현

시스템 구현 플랫폼은Windows 환경을 기반으로 하여 Java 2dk1.4.2.08(J2SE) 와 Jena 2.2, XML1.0, OWL언어를 이용하였다. Server를 통하여 가져온 web page의 URL를 토대로 Web Searcher는 네트워크를 통하여 가져온 HTML 문서를 임시 저장하고, 프로세스 는Information Extractor 모듈을 통하여 파싱을 통하여 (<property>, <value>)의 형태로 값을 추출하게 된다. (그림7)은 personal data의 유사관계를 표현하고 있다. "sameAs"라는 관계로 유사관계를 가진 property 값들을 표현하며 rdf:ID="e_mail_catpin"인 대표어는 "<captin>" 항목에 true값을 갖게 하여 대표어를 구분해 준다.

```
••••••••••••••••
<Person rdf:ID="e_mail">
  <owl:sameAs>
    <Person rdf:ID="e_mail_captin">
      <owl:sameAs rdf:resource="#e_mail"/>
      <owl:sameAs>
        <Person rdf:ID="아메일">
          <owl:sameAs rdf:resource="#e_mail_captin"/>
        </Person>
      </owl:sameAs>
      <owl:sameAs>
        <Person rdf:ID="전자메일">
          <owl:sameAs rdf:resource="#e_mail_captin"/>
        </Person>
      </owl:sameAs>
      <captin rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >true</captin>
      <owl:sameAs>
        <Person rdf:ID="electronic_mail">
          <owl:sameAs rdf:resource="#e_mail_captin"/>
        </Person>
      </owl:sameAs>
    </Person>
  </owl:sameAs>
</Person>
••••••••••••••••
```

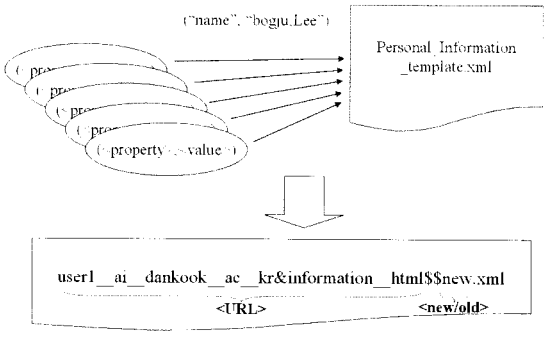
(그림 7) Synonym Ontology Instance

온톨로지에 정의되어 있는 <Property> 값은 Synonym Ontology에 아래와 같이 질의를 하여 대표어로 변경한다.

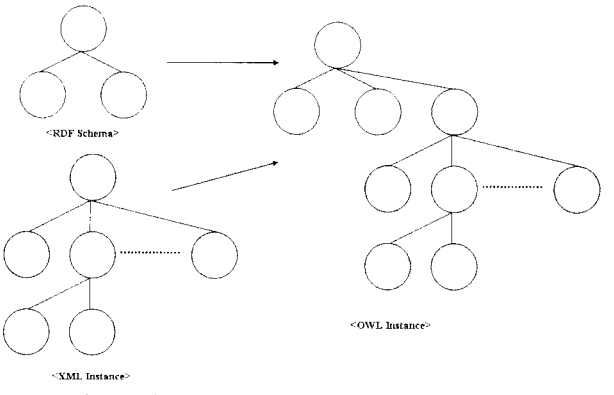
대표어를 찾아내는 Query

```
SELECT ?same WHERE(feat:"+inputname+" owl:sameAs ?same) ,+
"(?same feat:captin ₩"true₩")"        + "USING feat FOR " + namespace
```

- same : 대표어를 받아오는 변수
- inputname : 추출해 온 <Property> 변수
- captin : 대표어 인식 기호



(그림 8) XML 파일 생성 과정



(그림 9) RDF Schema와 XML Instance Merge

inputname 변수에서 추출된 Property 정보의 sameAs 라는 관계를 가진 데이터 중 captin이 true(대표어를 의미함)인 데이터를 ?same 변수로 추출해온다. 이렇게 변경이 이루어진 정보는 (그림 8)과 같이 Template.xml을 기초로 하여 각각의 property 에 value를 삽입하여 XML 파일로 저장을 한다. 파일명은 URL과 문서 버전(new/old)으로 유일성을 갖추게 된다.

이렇게 URL별로 생성된XML 파일들은Java의 XML Parser(DOM)를 이용하여 (그림 9)의 각각의 node로 구성된 tree 구조로 구성하게 된다.

Metadata Generator를 통해서 위의 (그림 10)과 같이 RDF Schema와XML Instance를 사용하여 아래 (그림 10) 우측과 같이 RDF Ontology Instance를 생성하게 된다.

이렇게 Instance가 생성되면 기존의 최근 문서를 OLD로 분류되면서 NEW 문서로 저장한다. Compare Module 프로세스는 생성된 두 문서를 Jena를 이용하여 tree 구조로 된 객체를 생성하고 Personal Ontology Schema를 통해 아래와 같이 두 버전의 객체를 비교할 property를 추출한다.

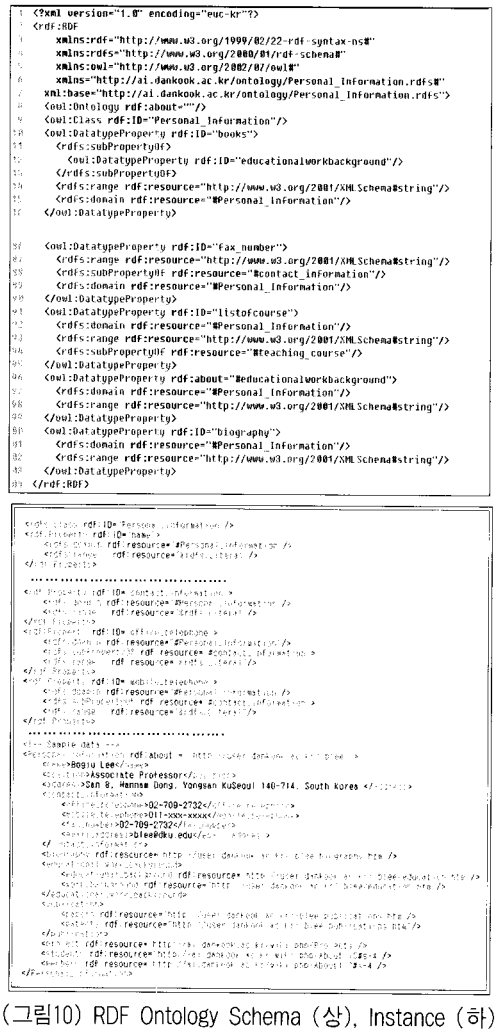
<Personal Ontology에서 Property추출 Query>

```
SELECT ?s WHERE (?s, ?p owl:DatatypeProperty)
```

이렇게 추출되어진 property를 기준으로 하여 두 객체의 property 별로 아래와 같은 질의를 통하여 정보를 추출하고 두 정보를 비교하게 된다.

〈객체의 Property별로 정보를 추출〉

```
"SELECT ?o WHERE
(s<"+namespace+(String)vc.get(i)+"> ?o)"
```

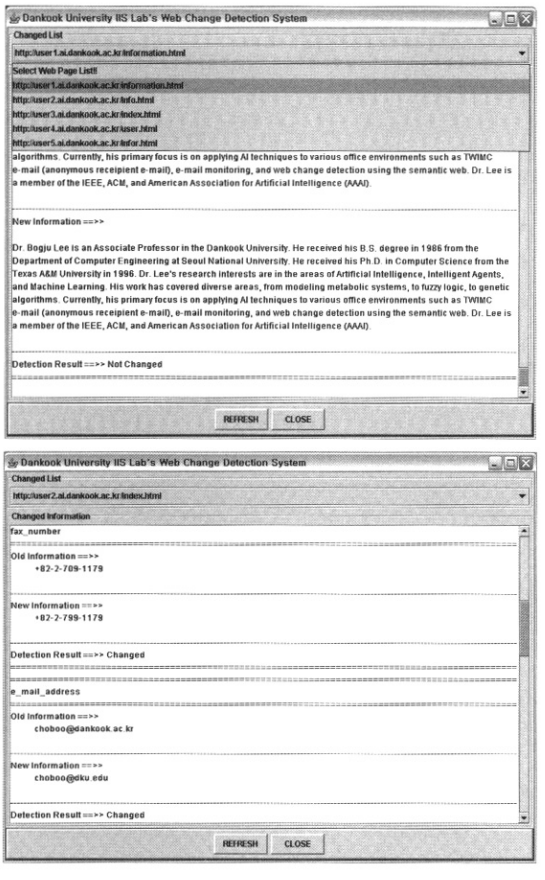


(그림10) RDF Ontology Schema (상), Instance (하)

정보간의 비교 시 발생하는 상태는 아래 <표 1>과 같이 초기상태(Initial), 변화없음(Not Changed), 기존정보가 최근 문서에서 삭제된 경우(Deleted), 최근 문서에 추가된 경우(Inserted), 두 문서 모두 정보가 존재하지 않는 경우(Not Exist), 마지막으로 정보가 변경 된 경우로 구분되며 <표 1>와 같이 코드로 구분하였으며, 2, 3, 5코드의 경우가 정보가 변경된 경우에 해당된다.

〈표 1〉 코드로 표현한 상태표

Code Value	Status
0	Initial
1	Not Changed
2	**Deleted**
3	**Inserted**
4	Not Exist
5	**Changed**



(그림 11) 사용자 인터페이스 변화된 웹 페이지 리스트 (상), 변화페이지 상세보기 (하)

Result Process Module을 통해서 변화가 감지된 URL 정보 값들만 (그림 11)의 좌측과 같이 사용자의 Select List에 보여주게 되고 사용자는 원하는 정보를 선택할 수 있다. (그림 11)의 우측과 같이 선택에 따른 결과 값을 이전정보와 새로운 정보를 비교하여 <표 1>의 Status 값이 출력하게 된다.

6. 실험 결과

제안된 본 시스템의 성능 실험을 위해 서버의 경우 인텔 1.60GHz CPU, 512MB, 클라이언트의 경우 인텔3.0GHz CPU, 1GB RAM를 사용하였다. 실험에 사용된 테스트 웹 페이지는 <표 2>와 같이 www.changedetection.com을 대조군으로 사용하였으며, URL별로 각각의 페이지를 작성하여 등록 하였다. 또한 본 시스템이 제안했던 의미적 변화에 대해 초점을 두고 의미적 변화 즉 사용자에게 필요한 정보들의 변화에 대한 비교와HTML 문법의 변화 같은 의미 없는 정보 등의 구문적 변화를 동시에 주어 실험해 보았다. <표 2>와 같이 http 프로토콜을 통해 전송되는 전체 정보를 대상으로 체크하는 대조군의 경우는 의미적 변화와 구문적 변화에 반응하여 결과를 보여주는 반면, 본 시스템은 의미적 변화에는 반응하는 것을 알 수 있다. <표 3>에서의 종합 실험 결과를 보면 본 시스템은 대조군에 비해 의미적 변화만

탐지하였다. 하지만 총100회의 변화 횟수에 2회의 의미적 변화는 탐지하지 못하였다. 그 원인은 HTML 문서에서의 작성 방식에 따라서Information Extractor에서 property 를 제대로 추출해내지 못하는 데 있었다.

〈표 2〉 샘플 인물 정보 웹 페이지의 변화에 따른 결과

Detection URL	구문적 변화			의미적 변화		
	변화 횟수	변화 횟수	변화 횟수	변화 횟수	changedetect.com	본 시스템
http://user1.ai.dankook.ac.kr/infor.html	0	0	0	0	0	0
http://user2.ai.dankook.ac.kr/infor.html	5	5	0	5	5	5
http://user3.ai.dankook.ac.kr/infor.html	10	10	0	10	10	10
http://user4.ai.dankook.ac.kr/infor.html	15	15	0	15	15	15
http://user5.ai.dankook.ac.kr/infor.html	20	20	0	20	20	20
http://user6.ai.dankook.ac.kr/infor.html	0	0	0	0	0	0
http://user7.ai.dankook.ac.kr/infor.html	5	5	0	5	5	5
http://user8.ai.dankook.ac.kr/infor.html	10	10	0	10	10	10
http://user9.ai.dankook.ac.kr/infor.html	15	15	0	15	15	14
http://user10.ai.dankook.ac.kr/infor.html	20	20	0	20	20	19

〈표 3〉 위의 실험에 따른 종합 결과

	구문적 변화 (감지/변화)	의미적 변화 (감지/변화)	구문적 변화 감지율(%)	의미적 변화 감지율(%)
changedetection.com	100/100	100/100	100%	100%
본 시스템	0/100	98/100	0%	98%

7. 결론 및 연구 방향

본 논문은 시맨틱 웹이 제안하는 온톨로지를 기초로 하여 웹 페이지의 의미기반의 변화를 감지하고, 빠른 탐지 및 성능 향상을 위해 서버와 클라이언트의 역할을 분할하는 시스템을 구현하였다. 실험 결과에서 본 시스템이 의미기반의 변화에만 반응하여 사용자에게 유용한 정보를 제공함을 확인하였다. 실험에서 대조군의 경우 탐지 반응 간격이 1회/일로 반응하여 1일 동안 3회 이상의 변화한 경우 마지막 변화에만 결과를 보여주었다. 반면 본 시스템의 경우는 시스템이 동작상태에서는 즉시 반응하였다. 향후 본 연구는 서버의 Personal Page Detector 모듈에서 무작위로 탐색한 페이지의 정보를 분석하고 기계 학습을 이용하여 인물 정보 페이지를 자동으로 인식할 수 있도록 하여 보다 자동화 되고 신뢰성 있는 저장소를 구축하는 것이다. 또한HTML에서의 property 및 value 쌍을 추출하는 방식에 대한 문제점을 해결하기 위한 연구가 필요하다. 인물 정보 페이지가 아닌 증시, 뉴스, 쇼핑, 경매 등의 빠른 정보를 얻어야 하는 도메인에 대하여 본 시스템에서 사용한 방법을 응용한다면 좀더 빠르고 정확한 결과를 제시해 줄 수 있을 것이다.

참 고 문 헌

[1] Gregory Cobena, Serge Abiteboul, INRIA Rocquencourt, France Amelie Marian, Columbia University, NY "Detecting Changes in XML Documents," March, 2002. ICDE 2002 (San Jose), http://www-rocq.inria.fr/~cobena/cdrom/www/xydiff/eng.htm

[2] Yuan Wang, David J. DeWitt, Jin-Yi Cai, University of Wisconsin - Madison "X-Diff:A Fast Change Detection Algorithm for XML Documents," March, 2003.

[3] E. Berk "HTMLDiff:A Differencing Tool for HTML Documents", Student Project, Princeton University, http://www.HTMLdiff.com

[4] FreeFind.com, http://www.changedetection.com/monitor.HTML

[5] ATS Consulting AS, http://www.watchthatpage.com/index.jsp

[6] iMorph's InfoMinder, http://www.infominder.com/webminder/index.jsp

[7] SemanticWeb.org, http://www.w3.org

[8] Michel Klein, Dieter Fensel Vrije Universiteit Amsterdam, Atanas Kiryakov, and Damyan Ognyanov OntoText Lab., Sirma AI Ltd. "Ontology versioning and change detection on the Web" http://gunther.smeal.psu.edu/klein02ontology.html

[9] RDF (Resource Description Framework), http://www.w3.org/RDF

[10] 김태훈, 최중민, 한양대학교 전자계산학과, "An Intelligent Web Browsing Agent for User-Oriented Internet Information Retrieval", Journal of Korea Information Science Society (B), Vol.25, No.7, pp.1064-1078, 1998.

조 부 현

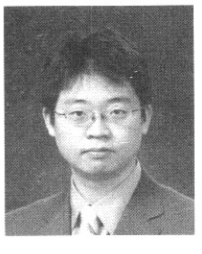
e-mail : choboo@dku.edu

2004년 단국대학교 전기전자컴퓨터공학과 (학사)

2006년 단국대학교 전자컴퓨터공학(석사)

2006년 현재 동양 시스템즈㈜ 근무

관심분야 : 시맨틱웹, 기계학습, 정보추출



민 영 근

e-mail : minykreva@nate.com

2005년 단국대학교 전기전자컴퓨터공학과 (학사)

2005년 현재 단국대학교 전자컴퓨터공학 석사과정

관심분야 : 시맨틱 웹, 기계 학습

이 복 주

e-mail : blee@dku.edu

1986년 서울대학교 컴퓨터공학과(학사)

1992년 University of South Carolina 컴퓨터학과(석사)

1996년 Texas A&M University 컴퓨터학과 (박사)

1997년~1999년 AT&T

2000년~2001년 한국정보통신대학교(ICU) 조교수

2001년~현재 단국대학교 전기전자컴퓨터공학과 조교수/부교수

관심분야 : 기계 학습, 데이터마이닝, 시맨틱 웹