Journal of Korea Multimedia Society Vol. 10, No. 11, November 2007(pp. 1496-1506)

웹 2.0 기반 RSS 데이터 수집 엔진의 설계 및 구현

강필구[†], 김재환[††], 채진석[†††], 이상준[††††]

요 약

기존의 웹 서비스가 정적이고 수동적인데 반해 최근의 웹 서비스는 점차 동적이고 능동적으로 변화하고 있는데, 이러한 웹 서비스 변화의 흐름을 잘 반영하는 것이 웹 2.0이다. 웹 2.0의 특징은 사용자가 능동적으로 참여하여 정보를 생산하는 것인데, 이렇게 되면, 생산되는 정보의 양이 지속적으로 증가하게 되므로 더 빠르고 정확한 정보를 공유할 필요가 있다. 이러한 필요성을 충족시키는 기술이 웹 2.0의 웹 신디케이션 기술과 태그 기술이다. 웹 신디케이션은 웹 사이트의 내용을 다른 사이트나 사용자가 받아볼 수 있도록 피드를 만든다. 태그는 정보의 핵심이 되는 단어로, 여러 인터넷 사용자들이 태그를 통한 검색으로 좀 더 빠른 정보의 공유를 가능하게 한다. 이 논문에서는 웹 2.0의 핵심 기술인 웹 신디케이션과 태그의 활용을 높이기 위한 방법으로 데이터 수집 엔진을 만들어 데이터를 효율적으로 관리하는 기법을 제안하였다. 데이터 수집 엔진은 데이터베이스에 저장된 사용자의 웹 사이트 정보를 이용하여 사용자의 웹 사이트에 접속하여 업데이트된 데이터를 수집한다. 이 논문에서 제안한 데이터 수집 엔진을 사용하여 실험한 결과 기존의 기법에 비해 검색 속도가 최대 3.14배 향상되었고, 연관 태그를 구성하는데 사용되는 데이터 건수가 최대 66%까지 감소함을 확인할 수 있었다.

A Design and Implementation of RSS Data Collecting Engine based on Web 2.0

Pilgu Kang[†], Jaehwan Kim[††], Jinseok Chae[†††], Sangjun Lee[††††]

ABSTRACT

The environment of web service has changed a great deal due to the progress of internet technology and positive participation of users. The established web service is static and passive, but the recent web service is becoming dynamic and active. Web 2.0 reflects current web service change well. The primary feature of web 2.0 is positive participation of users. Since the size of generated information is becoming larger, it is highly required to share the information fast and correctly. The technology to satisfy this need is web syndication and tagging in web 2.0. The web syndication makes feeds for another site or users to receive the content of web site. In addition, the tagging is the kernel of a information. Many internet users share rapidly the information through tag search. In this paper, we propose the efficient technique to improve the web 2.0 technology such as web syndication and tagging by using the data collection engine. Data collection engine has stored in a database, a user's Web site to use the information. and it has a user's Web site with access to updated data to collect. The experimental results show that our approach can improve the search speed up to 3.14 times better than the existing method and reduce the size of data up to 66% for building associated tags.

Key words: Web 2.0 technology(웹 2.0 기술), RSS Data Collecting Engine(RSS 데이터 수집 엔진)

1. 서 론

인터넷의 발달과 사용자의 적극적인 참여에 힘입어 웹 서비스 환경은 다양하게 변화하고 있는데, 기존의 웹 서비스가 정적이고 수동적인데 비해 최근의 웹 서비스는 점차 동적이고 능동적으로 변화하고 있

다. 이러한 웹 서비스 변화의 흐름을 잘 반영하는 것이 웹 2.0이다[1].

웹 2.0의 주요한 특징은 사용자가 능동적으로 참여한다는 것이다[2]. 사용자들은 인터넷 환경에 익숙해지면서 더 많은 정보를 요구하게 되었고, 스스로 가치 있는 정보들을 생산해 내기 시작했다. 그리고 이렇게 생산된 정보는 인터넷을 통하여 다른 사용자들과 공유되고 의견을 나누는데 사용되고 있지만, 그 양이 지속적으로 증가하고 있으므로 더 빠르고 정확하게 정보를 공유하는 기술이 필요하게 되었다. 현재 이러한 필요성을 충족시키는데 유용하다고 생각되는 기술이 웹 2.0의 웹 신디케이션 기술과 태그 기술이다[3].

웹 신디케이션은 웹 사이트의 내용을 다른 사이트나 사용자가 받아볼 수 있도록 피드(feed)를 만드는 것을 말한다. 사용자는 이렇게 생성된 피드를 통해 자신이 원하는 웹 사이트 내용을 구독할 수 있게 되었다.

태그는 정보의 핵심이 되는 단어를 의미하는데, 사용자들은 태그를 통한 검색으로 좀 더 빠르게 정보를 공유할 수 있게 되었다. 기존의 웹은 게시판처럼 리스트로 보여주고 검색을 통해 결과를 추출해내며, 컴퓨터에 의해 자동으로 정보를 생산하거나 수집하고 있으므로, 리스트 구조의 로그로 생각할 수 있다. 하지만 웹 2.0에서 대부분의 정보는 사용자에 의해 생산되고, 사용자가 붙인 태그에 의해 분류되고, 연관성 있는 태그에 의해 서로 관계를 맺게 된다.

사용자는 게시판이나 블로그를 통하여 만든 정보를 태그를 이용하여 분류한다. 이는 기존의 단일 분류 구조를 벗어나 다중 분류 구조의 확장성을 제공한다. 하지만 다중 분류 구조의 확장성은 의미론적인 단점을 가지고 있다. 책을 소개하는 글에서 사용자는 분류 구조에 '책'이라고 태그를 붙일 수도 있지만, '도서', 'book' 같이 다양한 태그를 붙일 수도 있다. 이것들은 의미론적으로는 같은 태그일 수 있지만 엄밀히 말하면 서로 다른 태그가 되는 것이다[4].

이 논문에서는 웹 2.0의 핵심 기술인 웹 신디케이션과 태그의 활용을 높이기 위한 방법으로 데이터 수집 엔진을 만들어 데이터를 효율적으로 관리하는 기법을 제안하였다. 이 논문에서 제안한 데이터 수집 엔진을 사용하여 실험한 결과 기존의 기법에 비해 검색 속도가 최대 3.14배 향상되었고, 연관 태그를 구성하는데 사용되는 데이터 건수가 최대 66%까지 감소함을 확인할 수 있었다.

본 연구의 구성은 2장에서는 웹 2.0과 웹 2.0에 사용되는 웹 신디케이션과 태그에 대해서 살펴보고, 3장에서는 데이터 수집 엔진을 설계하고, 4장에서는 실험을 통한 성능 분석 결과를 제시하였다. 마지막으로 5장에서 결론을 제시하였다.

2. 관련 연구

2.1 웹 2.0

웹 2.0은 O'reilly사와 MediaLive사의 컨퍼런스 과정에서 탄생한 단어로, 이전의 웹과 단절된 것이 아니라 연속성을 가진 형태에서 웹의 진화 환경에 대해 이야기하는 도중 도출되었다.

O'reilly가 웹 2.0 컨퍼런스에서 다루고자 한 주제는 새로운 서비스의 흐름이었다. 미국의 실리콘 밸리에서는 기존의 웹과 다른 서비스들이 끊임없이 생겨나고 있었는데, 이는 과거 닷컴 거품 때 새로운 사이트가 우후죽순으로 생겨나는 것과는 다른 양상을 보이고 있었다. 전방위적으로 거대한 흐름을 형성하면서 기존 웹과는 다른 개념의 서비스가 등장하기 시작한 것이다. 이 흐름의 정체를 파악하기 위해 O'reilly 미디어에서는 웹 2.0 컨퍼런스라는 이름을 붙이고 인터넷 분야의 인사들을 초청해 자유 토론 형태로 최근 기술 동향과 서비스, 플랫폼에 대한 이야기를 나누었다. 그 결과 웹 서비스의 새로운 흐름을 웹 2.0이라는 용어 속에 정리하기 시작했고, 새로운 서비스의 특징을 하나씩 분류해가면서 웹 2.0의 모양을 만들기 시작했다. 따라서 웹 2.0은 신기술을 뜻하는 용어가 아

※ 교신저자(Corresponding Author) : 채진석, 주소 : 인천광역시 남구 인천대길 319(402-7491), 전화 : 032)770-8427, FAX : 032)773-8428, E-mail : jschae@incheon.ac.kr
접수일 : 2007년 8월 21일, 완료일 : 2007년 10월 30일
† 정회원, 아이티플러스(주)
(E-mail : kpg2976@incheon.ac.kr)
†† 학생회원, 인천대학교 컴퓨터공학과 석사과정
(E-mail : jhkim@incheon.ac.kr)
††† 종신회원, 인천대학교 컴퓨터공학과 부교수
†††† 종신회원, 숭실대학교 컴퓨터학부 조교수
(E-mail : sangjun@ssu.ac.kr)
※ 본 연구는 2006년도 인천대학교 자체연구비 지원에 의해 수행되었음.

니라 새로운 흐름 자체를 뜻하는 용어로 정착되기 시작했다. 새로운 개념, 새로운 서비스, 새로운 플랫폼 등을 웹 2.0 안에 포함시키고 있는 것이다[5].

웹 2.0에 대해서 O'reilly는 다섯 쪽의 긴 문서를 통해 설명했지만 정작 웹 2.0이 무엇이냐는 질문에 대한 명확한 정의는 내리지 못하고 있다. 아직도 웹 2.0에 대한 개념 정의는 계속 첨삭이 되면서 변화하고 있기 때문이다. 또한 웹 2.0이 기술적인 기준으로 구분하는 것이 아니고 웹 2.0에 사용된 기술과 개념이 이전부터 존재했던 것이기에 딱 잘라 무엇이 되면 웹 2.0이고, 그것이 아니면 웹 2.0이 아니라고 말하기도 어려운 것이 사실이다. 결국 당분간 웹 2.0을 명확하게 정의 내리기란 쉽지 않을 것이다.

다만 O'reilly는 웹 2.0의 용어를 직접 정의하기보다 닷컴의 붕괴 이전, 이후에 따라 각각의 특징별로 웹 1.0, 웹 2.0 서비스를 규정하고 예를 들어 설명했다.

한국어 위키 백과사전에서는 웹 2.0을 '월드 와이드 웹이 웹 사이트의 집합체에서 최조 사용자에게 웹 애플리케이션을 제공하는 하나의 완전한 플랫폼으로 진화하는 변화 양상에 대한 인식을 반영하는 의미로 종종 사용되는 용어' 라고 설명하고 있다. 웹 2.0이 구현되어 나타나는 대표적인 형태로는 블로그, 위키, Bit Torrents, Creative Commons, Google, IPO, RSS, Social Software, Web APIs, REST, XHTML/CSS 등이 있다.

표 1. 웹 1.0과 웹 2.0의 비교

웹 1.0	웹 2.0
DoubleClick	Google Adsense
Ofoto	Flickr
Akamai	BitTorrent
Mp3.com	Napster
Britannica Online	Wikipedia
Personal websites	blogging
evite	Upcorming.org and EVDB
Domain name speculation	Search engine optimization
Page views	Cost per click
Screen scraping	Web services
publishing	participation
Content management systems	wikis
Directories(taxonomy)	Tagging("folksonomy")
stickiness	syndication

2.2 웹 신디케이션

웹 신디케이션은 웹 사이트의 내용을 다른 사이트나 사용자가 받아볼 수 있도록 피드(내용의 요약정보나 링크)를 만드는 것을 말한다. 웹 신디케이션은 뉴스 사이트나 블로그에서 사용되기 시작했지만 점차 정보의 내용에 구애받지 않고 웹 사이트 내용의 업데이트를 전달하기 위해 사용되고 있다.

웹 신디케이션을 위한 피드 파일 형식으로는 HTML이나 Javascript와 같이 HTTP 프로토콜을 통해 전송될 수 있는 모든 형식이 가능하지만 일반적으로 XML 형식에 맞추어 만들어진다[6].

웹 신디케이션의 장점들로는 다음과 같은 것들이 있다.

- 선택적 구독 - 사용자가 원하는 주제와 정확히 일치하는 채널 선택
- 빠른 구독 - 동시에 다양한 채널 소스 접근
- 히스토리 관리 - 다양한 채널의 과거 기록들 보관이 가능
- 자동화된 컨텐츠의 편리한 연동 - 배급 / 수집
- 컨텐츠 재사용성 - 구조화된 XML 데이터로 손쉬운 변환 및 처리
- 커뮤니케이션 방식의 변화 - 1:1에서 1:N으로 동시 접속

웹 신디케이션의 확산 및 발전 배경으로 블로그를 빠트릴 수 없다. 블로그와 웹 신디케이션은 완벽하게 상호 보완적인 관계를 형성하며 서로를 발전시켜 왔다고 해도 과언이 아니다. 수많은 개인 블로그들의 정보를 웹 신디케이션 기능 없이 개별적으로 접속하고 활용해야 했다면, 오늘날처럼 폭발적으로 블로그가 확산될 수는 없었을 것이다.

특히 웹 신디케이션의 응용이 단순히 블로그의 컨텐츠 배급에만 한정되는 것이 아니라, 웹 신디케이션 기반의 광고 기법, 일정 및 스케줄 공유, 기업 홍보 및 마케팅 수단, 쿠폰 발행, 소프트웨어 배포, 오디오/비디오 컨텐츠의 배급, 기업 간 정보 공유 및 지식공유 수단 등의 응용들에까지 확산되고 있다[7].

웹 신디케이션의 대표적인 파일 유형으로는 RSS와 ATOM이 있는데, RSS는 웹 신디케이션에 적합한 파일 형식으로 제안된 것으로 현재 몇 가지 버전이 사용되고 있다[8]. 또한 최근에는 ATOM이라는 형식의 프로토콜이 제안되어 사용되고 있다[9].

2.3 태그

2.3.1 태그의 정의

태그는 일반적으로 학생들의 이름표, 수화물의 딱지, 제품의 상표 등을 뜻하는데, 웹에서도 태그는 어떤 글이나 자료에 붙여 놓은 추가 정보를 뜻하고 있다. 태그는 하나의 정보에 부가적인 설명을 기록할 수 있다는 점에서 카테고리와 유사하지만, 태그는 여러 단계를 거칠 필요가 없고 특별한 규칙이 없다는 장점을 가지고 있다. 또한 태그는 사용자가 마음대로 지정한 태그가 중요한 정보가 되고 태그와 태그 사이에서 입체적으로 다양한 정보들과의 관계를 형성하고 서로 사용할 수 있다는 점에서 키워드와 카테고리와는 다른 장점을 가지고 있다[10].

2.3.2 연관 태그와 대표 태그

태그가 웹 2.0에서 가장 주목 받는 이유는 글과 글, 태그와 태그 사이의 연관 관계를 맺을 수 있다는 것이다. 그림 1과 같이 글과 글 사이에는 연관성이 없지만 같은 태그를 입력 하였을 경우 태그를 통해서 글과 글 사이의 연관 관계를 맺을 수 있다. 그림 2와 같이 하나의 글에 여러 개의 태그를 입력하는 경우, 하나의 글에 입력된 여러 개의 태그는 서로 연관성을 맺게 된다. 이렇게 연관성을 맺게 된 태그를 연관 태그라고 부르는데, 연관 관계가 맺어진 태그들 중에서 중복된 태그의 개수가 많을수록 정보의 연관도도 높아진다고 생각할 수 있다. 그리고 연관 태그 중에서 가장 높은 연관성을 가지고 있는 것이 대표 태그가 될 수 있다[11].

2.3.3 태그 구름과 태그 맵

태그 구름(Tag Cloud)은 태그들을 아무런 분류 없이 나열한 것으로, 나열된 모양이 구름과 같다고 하여 태그 구름이라고 부른다. 태그 맵(Tag Map)은 서로 연관성 있는 태그들을 태그 구름 형식으로 모아 놓은 것을 의미한다. 관련된 연관 태그를 링크로 묶고, 태그의 규모를 적용시켜 하나의 큰 태그 구름을 만들 수 있다. 태그 구름은 한눈에 모든 태그들의 경향을 살펴볼 수는 있지만 전체적인 태그의 연관성을 표현하기는 부족하다. 태그 맵은 단순한 경향을 살펴보는 것뿐만 아니라 태그와 태그 사이의 연관 관계를 표현하고, 태그를 탐색해 나가는 중요한 방식 중의 하나가 될 수 있다[12].

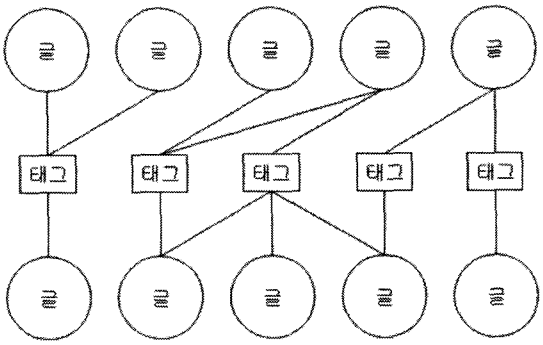


그림 1. 태그를 이용한 글과 글 사이의 관계

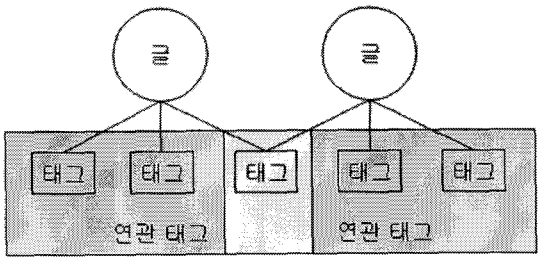


그림 2. 글과 글 사이의 연관 태그

2.4 기존의 RSS 데이터 수집 연구

기존의 RSS 데이터 수집 연구는 RSS를 이용하여 웹의 다른 문서자료를 수집하는 연구와 RSS 자료를 수집한 뒤 그 해당 자료에서 사용자가 원할 만한 자료를 선별하여 보여주는 연구가 나와 있다. 다른 형식의 문서자료를 RSS로 변환하여 얻는 이점으로는 현재 RSS가 문서의 배급과 수집의 표준 포맷으로 널리 사용되고 있으므로 다수의 사용자에게 배포하기 쉽다는 점을 들 수 있다[13,14]. 자료를 선별하여 보여주는 연구는 수집한 RSS 피드의 게시물의 수, 갱신률, 갱신 주기 등을 고려하여 각 RSS 채널에 순위를 부여하여 사용자에게 보여준다[15]. 그러나 위의 연구들은 RSS 피드에서 제공하는 중요한 정보 중 하나인 태그에 대해서는 다루지 않는다.

3. RSS 데이터 수집 엔진

3.1 시스템 구성도

그림 3은 데이터 수집 엔진의 시스템 구성도를 보여주고 있다. 그림 3에서 보는 것과 같이 이 시스템은 블로그의 RSS 문서를 수집하는 데이터 수집 엔진과

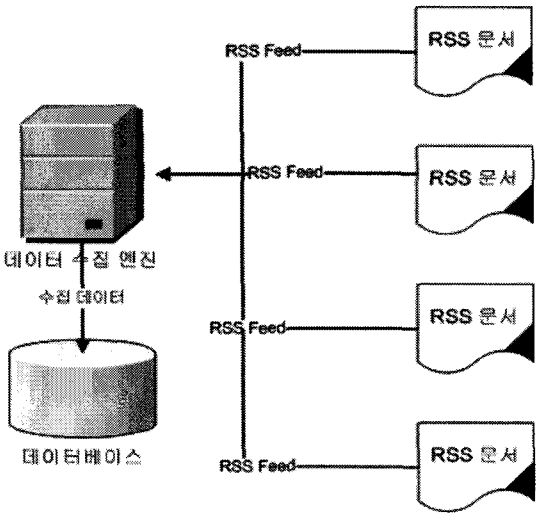


그림 3. 전체 시스템 구성도

엔진에서 수집한 데이터를 저장하는 데이터베이스로 구성되어 있다.

3.1.1 데이터 수집 엔진

데이터 수집 엔진은 데이터베이스에 저장된 사용자의 웹 사이트 정보를 이용하여 사용자의 웹 사이트에 접속하여 웹 사이트에 업데이트된 데이터를 수집한다. 데이터 수집 엔진은 그 역할에 따라 프로퍼티 컴포넌트, 로그 컴포넌트, 데이터베이스 컴포넌트, 피드 파싱 컴포넌트의 4개 컴포넌트로 구성되어 있다. 그림 4는 데이터 수집 엔진의 구성도를 보여주고 있다.

데이터 수집 엔진의 컴포넌트 중 프로퍼티 컴포넌트는 데이터 수집 컴포넌트 환경 설정 파일을 관리한다. 프로퍼티 파일은 데이터베이스 접속 정보 및 로그 정보를 관리한다. 프로퍼티 파일에 적용된 내용은 프로퍼티 엔진에서 해시 테이블로 관리되어 실시간으로 적용된다.

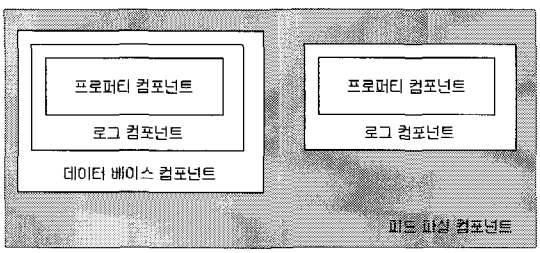


그림 4. 데이터 수집 엔진 구성도

데이터베이스 컴포넌트는 데이터베이스와 관련된 모든 명령을 관리하는 컴포넌트이다. 생성자가 생성될 경우 프로퍼티에서 정보를 데이터베이스 접속 정보를 받아와서 자동으로 접속한다. 생성자가 소멸하기 전까지 데이터베이스 접속 상태를 유지하고 있으며, 데이터베이스의 SELECT와 DML 명령을 수행한다.

로그 컴포넌트는 데이터 수집 엔진에서 발생하는 정보, 에러, 쿼리에 대한 로그를 관리한다. 프로퍼티 파일에서 지정된 위치에 {로그 파일명}{로그 파일 포맷}{시퀀스} 형식으로 로그 파일을 생성한다. 생성된 파일이 지정된 크기를 초과 하면 시퀀스를 증가 시켜 새로운 파일을 생성한다. 로그 파일에 기록되는 정보는 프로퍼티에 지정된 로그 레벨에 따라서 각각 다르게 적용된다.

피드 파싱 컴포넌트는 피드 문서를 통해 불러온 XML 문서를 탐색하고 수집 엔진이 필요한 요소들을 배열에 저장하고 데이터베이스에 입력하는 역할을 수행한다.

3.2 데이터베이스 모델링

일반적으로 태그를 사용하는 웹 사이트에서는 표 2와 같은 테이블 구조를 가지게 된다[16].

사용자가 입력하는 글에 대한 정보는 ENTRY_INFO 테이블에 저장되고, 태그에 대한 코드 정보는 TAG_INFO 테이블에 저장된다. 그리고 사용자가 입력한 글과 태그를 연결시키기 위해 사용하는 TAG_RELATION 테이블에서는 ENTRY_INFO 테이블의 SEQ 컬럼과 TAG_INFO 테이블의 SEQ 컬럼을 이용하여 관계를 구성한다. 따라서 사용자가 하나의 글에 N개의 태그를 입력하였을 경우 TAG_RELATION에는 $\{N\}$ 개의 데이터가 생성되며, 이렇게 생성된 데이터는 연관 태그로 활용된다. 그림 5는 태그를 사용하는 웹 사이트의 일반적인 스키마의 구조를 보여주고 있다.

표 2. 태그를 사용하는 웹 사이트의 일반적인 테이블 구조

테이블명	비 고
ENTRY_INFO	사용자가 입력한 글
TAG_INFO	사용자가 입력한 태그 정보
TAG_RELATION	글과 태그 사이의 관계

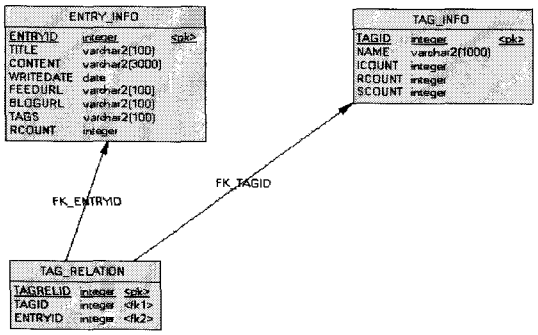


그림 5. 태그를 사용하는 웹 사이트의 일반적인 스키마

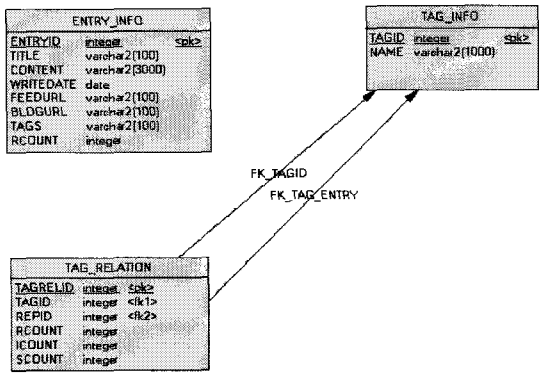


그림 6. 본 논문에서 제안한 스키마

본 논문에서는 태그 검색 속도를 향상시키고, 연관 태그 누적 데이터를 줄이기 위해서 표 3과 같은 스키마 구조를 사용한다.

일반적으로 태그를 사용하는 테이블과 비교했을 때 내용은 큰 차이가 없지만 데이터를 저장하는 방법에서 큰 차이가 있다. 일반적인 웹 사이트의 테이블에서는 ENTRY_INFO 테이블에서 태그를 관리하지 않고 TAG_RELATION 테이블에서 태그를 관리한다. 또한 TAG_INFO 테이블에서는 태그에 대해 조회수, 입력수 등을 관리하여 태그를 검색시 연관 태그의 정렬에 대한 우선순위를 관리한다. 하지만 제안한 테이블의 경우 ENTRY_INFO 테이블에서는 글과 태그 사이의 관계를 설정하기 위해서 하나의 컬럼에서 ","를 구분자로 하여 태그를 관리한다. 그리고 기존의 글과 태그 사이의 관계를 설정하여 연관 태그를 구성했던 TAG_RELATION 테이블은 사용자가 입력한 태그의 아이디 값을 대표 태그로 설정하여, 각각 연결된 태그의 정보를 모두 나열하고 관리하여 태그와 태그 사이의 관계를 설정한다. 마지막으로 TAG_LIST 테이블에서 관리되던 조회수, 입력수 등 정렬의 우선순위가 되는 컬럼 역시 TAG_RELATION 테이블에서 관리가 된다. 그림 6은 본 논문에서 제안한 스키마의 구조를 보여주고 있다.

태그와 태그 사이의 관계를 설정하는 TAG_RELATION 테이블에서는 사용자가 하나의 글에 N 개의 태그를 입력하였을 경우 TAG_RELATION은 $\{N^2 - \textit{중복된 태그}\}$ 만큼의 데이터가 추가된다.

데이터 수집 엔진에 의해 수집된 데이터를 저장하는 ENTRY_INFO 테이블은 TAG_INFO 나 TAG_RELATION 테이블과 직접적인 연관관계를 맺지 않고 TAGS 컬럼을 통하여 간접적인 연관관계를 맺는다.

표 3. 본 논문에서 제안한 테이블 구조

테이블명	비고
ENTRY_INFO	사용자가 입력한 글
TAG_INFO	사용자가 입력한 태그 정보
TAG_RELATION	태그와 태그 사이의 관계

3.3 피드 문서를 이용한 데이터 수집

피드 문서는 웹 사이트에 대한 기본적인 정보와 웹 사이트에서 업데이트된 정보를 XML 형태로 가지고 있다. 메타 사이트에서 필요한 데이터를 수집하기 위해서는 피드 문서의 XML을 분석하여 필요한 데이터를 수집해야 한다. 하지만 피드 문서는 다양한 버전을 가지고 있기 때문에 해당 버전에 따라서 데이터를 가져오는 방식을 다르게 해야 한다. 버전에 따라서 XML이 포함하는 정보의 내용이 변경되는 것이 아니라 XML 문서를 구성하는 요소와 속성의 이름이 변경되기 때문에 각각 버전에 맞게 XML을 파싱하여 데이터를 수집해야 한다.

웹 사이트 정보와 웹 사이트에서 업데이트된 데이터를 수집하는 방식이 각각 다르기 때문에 수집 엔진은 다음과 같은 두 개의 기법을 사용하고 있다.

3.3.1 웹 사이트에 대한 정보 수집

웹 사이트에 대한 정보는 XML 문서 상단에 단순한 형태로 구성된다. 따라서 그림 7과 같이 버전에 따라서 XML 문서를 각각 다르게 파싱하고, 파싱된 데이터를 해당하는 배열에 저장, 데이터베이스에 반영하는 단순한 과정을 거치게 된다.

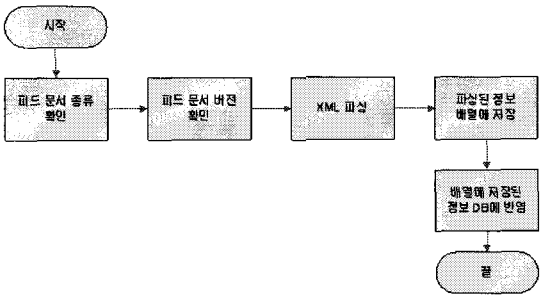


그림 7. 웹 사이트 정보 수집 순서도

3.3.2 웹 사이트에 업데이트된 데이터 수집

웹 사이트에 업데이트된 데이터를 수집하고 데이터베이스에 반영하는 방식은 웹 사이트 정보를 수집하는 방식보다 조금 더 복잡하다. 데이터를 수집하는 기본적인 방식은 웹 사이트 정보를 수집하는 방식과 동일하다. 하지만 그림 8과 같이 XML 문서에 대한 파싱 작업 후 태그 정보의 입력 유무에 따라 태그 입력 처리 단계가 추가 된다. 또한 XML 문서에서 웹 사이트 정보 데이터는 한 개의 노드를 구성하지만 웹 사이트에 업데이트된 데이터는 N 개의 노드를 구성한다. 따라서 데이터 수집시 N 개의 노드에 해당하는 만큼 반복을 한다.

3.4 태그 수집

웹 사이트에서 제공하는 피드 문서에는 태그에 대한 정의가 없기 때문에 웹 사이트 정보와 업데이트된 데이터를 가져오는 방식처럼 피드 문서의 특정 요소를 가져와서 데이터베이스에 저장하는 방법을 사용할 수 없다. 따라서 태그를 수집하는 방법에 대한 표준이 정해져 있지 않기 때문에 태그를 제공하는 웹 사이트마다 수집하는 방식이 다르게 된다.

3.4.1 웹 크롤

웹 크롤(web crawl) 방식은 일반적으로 가장 많이 사용하는 방식이다. 사용자가 입력한 태그는 웹 페이지를 통하여 화면에 보여줄 때 다음과 같은 방식으로 보여주게 된다.

```
<a href="URL" rel="tag">태그</a>
```

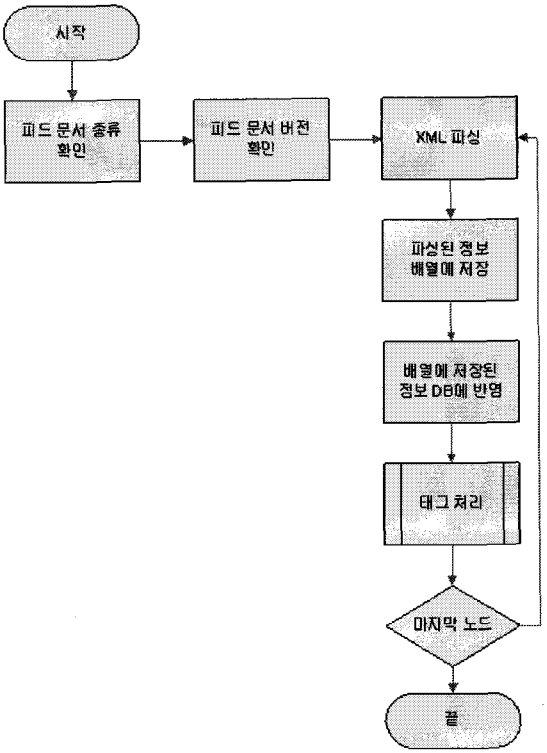


그림 8. 웹 사이트에 업데이트된 정보 수집 순서도

따라서 웹 페이지에서 태그를 추출하기 위해서 웹 사이트 페이지를 크롤해서 <a> 태그 속성 중 rel= "tag" 가 들어간 단어를 검색하여 추출한다.

3.4.2 피드 문서 검색

수집된 피드 문서에서 웹 사이트에 업데이트된 데이터를 표시하는 요소의 자식 요소 중 <CATEGORY> 요소가 2개 이상일 경우 2번째부터 태그로 판별하여 추출하는 방식을 사용한다.

3.5 태그 정보 생성

태그가 생성 된다는 것은 웹 사이트에 업데이트된 정보를 수집하는 것을 의미한다. 예를 들어 사용자가 다음과 같은 태그를 입력하였다고 가정하자.

- 글 1 : 웹 2.0에 대해서 {웹 2.0, 블로그, 태그}
- 글 2 : 블로그와 RSS {블로그, RSS}

사용자가 입력한 글 1, 글 2에 대한 정보는 표 4와

같이 저장된다. 사용자가 입력한 글은 시퀀스(SEQ)를 사용하여 각각 고유한 키 값을 가지게 된다. 또한 사용자가 입력한 태그는 표 5에서와 같이 입력한 태그에 대해서 시퀀스를 부여하여 관리한다. 마지막으로 일반적인 테이블의 경우 표 6과 같이 TAGID 와 ENTRYID 값을 이용하여 연관 태그를 관리하고, 제안한 테이블의 경우 표 7과 같이 각각의 태그를 ANCESTOR의 대표 태그로 설정하여 해당 태그와 연결되는 모든 태그를 입력한다.

표 4. ENTRY_INFO 테이블

ENTRYID	TITLE
1	웹 2.0에 대해서
2	블로그와 RSS

표 5. TAG_INFO 테이블

TAGID	NAME	HCOUNT
1	WEB 2.0	1
2	블로그	2
3	태그	1
4	RSS	1

표 6. 일반적인 블로그의 TAG_RELATION 테이블

TAGRELID	TAGID	ENTRYID
1	1	1
2	2	1
5	2	2
3	3	1
4	4	2

표 7. 논문에서 제안한 TAG_RELATION 테이블

TAGRELID	REPID	TAGID
1	1	2
2	1	3
3	2	1
4	2	3
7	2	4
5	3	1
6	3	2
8	4	2

일반적인 웹 사이트의 TAG_RELATION의 경우 $\{N\}$ 개의 데이터가 증가된 반면에 논문에서 제안한 TAG_RELATION 테이블은 $\{N^2 - \text{중복된 태그}\}$ 개의 데이터가 증가된다. TAG_RELATION에 데이터가 적을 경우, 일반적인 웹 사이트의 TAG_RELATION 구성이 데이터를 적게 가지고 있으나, 데이터의 양이 많아질수록 중복되는 태그가 많아지므로 논문에서 제안한 TAG_RELATION 테이블 구성이 데이터를 적게 가지고 있다.

4. 실험 결과 및 분석

4.1. 테스트 환경

표 8은 데이터 수집 엔진 시스템의 개발, 운영 환경을 보여주고 있다.

태그 검색 속도의 성능 측정을 하기 위하여 현재 운영되고 있는 메타 사이트인 이올린에서 제공한 정보를 참고하여 동일한 환경을 구성하였다[17]. 표 9는 2007년 4월 19일을 기준으로 하여 최근 데이터 10만 건의 각종 비율을 보여주고 있다.

태그 검색 속도와 연관 태그를 구성하기 위한 누적 데이터의 양을 측정을 하기 위하여 사용자가 하나의 글에 1~9개의 태그를 입력한다고 가정하고 랜덤

표 8. 데이터 수집 엔진 시스템 개발, 운영 환경

개발 OS	Microsoft Windows XP Pro
운영 OS	Microsoft Windows Server 2003 Enterprise Edition
DBMS	Oracle 10g (10.2.0.2)
개발 툴	Microsoft Visual C#.Net Microsoft Visual Basic 6.0 Microsoft Visual C++ 6.0

표 9. 2007년 4월 19일 기준 최근 데이터 10만 건의 각종 비율

내 용	값
태그를 하나 이상 가지고 있는 글의 비율	72.92%
태그를 가지고 있는 글의 평균 태그 수	4.63개
다른 사람이 쓴 적이 없는 태그를 사용한 비율	10.62%

으로 입력된 태그를 기존 스키마와 제안 스키마로 나누어서 입력하였다. 태그는 XXX 형식으로 대문자 3자리로 구성되었으며, 각 자리에 대문자 26자를 랜덤하게 배열하여 총 17,576 개를 생성한다고 가정하고 성능 측정을 수행하였다. 위 방식으로 생성된 데이터와 그에 따른 평균 태그 수, 중복 비율은 표 10과 같다.

4.2 태그 검색 속도

본 논문에서 제안한 스키마 구조의 속도와 일반적인 스키마의 태그 검색 속도는 표 11과 같다. 그림 9는 표 11의 내용을 그래프로 표현한 결과를 보여주고 있다.

그림 9에서 보는 것과 같이 제안 스키마의 검색 속도가 일반 스키마에 비해 빠른 것을 확인할 수 있다. 제안 스키마와 일반 스키마를 비교했을 때, 제안 스키마의 경우 데이터의 양이 증가하여도 검색속도의 증가폭이 일반 스키마에 비해 적은 것을 확인할 수 있다.

데이터를 5,000건에서 1,355,000건 까지 비교하였을 경우, 최소 1.49에서 최대 3.14배 검색 속도가 향상되었다. 이는 일반 스키마를 사용하여 태그를 검색할 경우, 검색어와 연결된 연결 태그 정보를 가져오기 위해서 TAG_RELATION 테이블을 두 번 검색하지만, 제안 스키마의 경우 TAG_RELATION 테이블을 한 번만 검색하기 때문인 것으로 판단된다. 또한 인덱스 생성시 이전 스키마의 경우 컬럼 2개를 인덱스 컬럼으로 설정해야 하기 때문에 제안 스키마의 인덱스 비용보다 많이 든다.

표 10. N개의 글에 대해서 생성된 평균 태그 개수

데이터	평균 태그 수	다른 사람이 쓴 적이 없는 태그 사용 비율
5,000	4.64	11.23%
10,000	4.72	10.39%
50,000	4.63	10.27%
100,000	4.72	11.47%
350,000	4.81	11.82%
735,000	4.61	11.72%
1,355,000	4.51	10.87%

그림 10은 일반 스키마에서 연관 태그 검색 질의를 보여주고 있고, 그림 11은 제안 스키마에서 연관 태그 검색 질의를 보여주고 있다.

표 11. N 개의 글에 대해서 생성된 태그 검색 시간

데이터	일반 스키마	제안 스키마
5,000	51.6ms	34.5ms
10,000	58.6ms	37.1ms
50,000	71.3ms	41.0ms
100,000	98.2ms	47.3ms
350,000	153.2ms	50.1ms
735,000	206.3ms	76.9ms
1,355,000	286.7ms	78.9ms

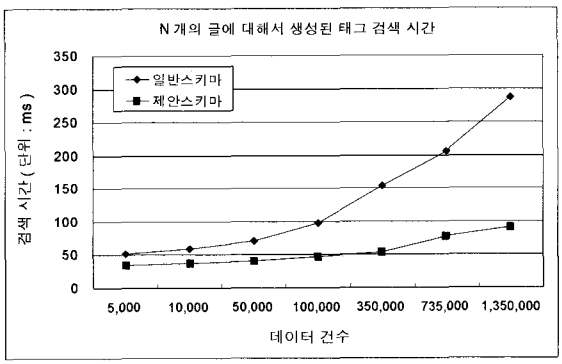


그림 9. N 개의 글에 대해서 생성된 태그 검색 시간

```
SELECT * FROM TAG_RELATION
WHERE ENTRYID IN (
   SELECT ENTRYID FROM
TAG_RELATION
   WHERE TAGID = 검색어
)
```

그림 10. 일반 스키마에서 연관 태그 검색 질의

```
SELECT * FROM TAG_RELATION
WHERE ANCESTOR = 검색어
```

그림 11. 제안 스키마에서 연관 태그 검색 질의

4.3 연관 태그 누적 데이터

표 12는 사용자가 입력한 N 개의 글에 대해서 연관 태그를 구성하기 위하여 생성된 데이터의 개수를 알려준다. 그림 12는 표 12의 내용을 그래프로 표현한 결과를 보여주고 있다.

그림 12에서 보는 것과 같이 연관 태그를 생성하는 과정에 있어서는 누적된 데이터가 적을 때에는 일반 스키마가 데이터를 적게 생성지만, 누적된 데이터가 많아질수록 제안 스키마가 유리하다는 것을 알 수 있다. 이는 일반 스키마의 경우 사용자가 하나의 글에 N 개의 글을 입력하였을 경우, TAG_RELATION에 N 개의 데이터를 입력하지만, 제안 스키마의 경우 $\{N^2 - 중복건수\}$ 만큼 입력하기 때문이다. 데이터를 5,000건에서 1,355,000건 까지 비교하였을 경우, 연관 태그를 구성하는데 사용되는 데이터 건수가 최대 66% 감소함을 확인할 수 있었다.

이 논문에서는 웹 상에 공개된 데이터를 사용하여 실험하였기 때문에 실제 환경과 비교하여 약간의 실험 오차가 있을 수 있으나, 그 정도는 무시할 수 있을 정도라고 판단하여 오차에 대해서는 기술하지 않았다.

표 12. N 개의 글에 대해 생성된 연관 태그의 개수

데이터	일반 스키마	제안 스키마
5,000	23,207	43,312
10,000	47,321	68,907
50,000	231,475	253,120
100,000	471,750	374,320
350,000	1,863,220	673,228
735,000	3,390,555	1,048,443
1,355,000	6,116,470	1,715,990

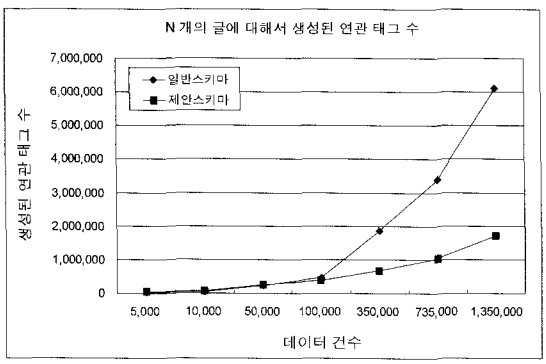


그림 12. N 개의 글에 대해 생성된 연관 태그의 개수

5. 결 론

본 논문에서는 웹 2.0이 등장하면서 대두되고 있는 웹 신디케이션 기술을 이용하여 데이터 수집 엔진을 개발하고, 정확한 정보와 빠른 처리 시간을 요하는 피드 수집 및 태그 관리의 효율성을 향상시킬 수 있는 방법을 제시하였다. 기존 메타 데이터 수집 사이트와 비교 했을 때, 본 논문에서 제안한 시스템의 가장 큰 특징이자 장점은 태그 검색 속도를 향상시키고, 연관 태그를 구성하는데 필요한 데이터의 양을 감소시킨 것이다. 태그 검색 시 이전 스키마의 경우 셀프 조인을 이용하여 테이블을 2번 호출하여 태그를 검색하지만, 제안 스키마의 경우 테이블을 1번 호출하기 때문에 검색 비용이 줄어든다. 또한 인덱스 생성시 이전 스키마의 경우 컬럼 2개를 인덱스 컬럼으로 설정해야 하기 때문에 제안 스키마의 인덱스 비용보다 많이 든다.

이 논문에서 제안한 기법을 사용하게 되면 연관 태그를 생성하는데 사용자가 입력한 태그에 대해 자동으로 연관 관계를 맺어주게 되므로 연관 관계를 생성하는데 있어 생기는 불편함을 덜 수 있게 되었다.

향후 연구 방향으로는 태그의 의미를 생각하여 태그와 태그 사이에 의미론적인 관계가 생성될 수 있도록 확장시키고, 대표 태그를 선정하는 기준을 더 정확하게 마련해야 할 필요가 있다. 또한, 대표 태그 선정을 위해 사용자가 불법적으로 조회빈도를 높일 수 있는 현상을 방지할 수 있는 방안을 마련하고, 스팸에 대한 보안을 강화해야 할 필요가 있다.

참 고 문 헌

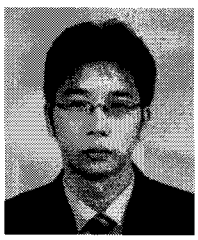
[1] http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

[2] 최호찬, "인터넷의 새로운 문화 블로그," 경향잡지, pp. 107-109, 2004년 3월호.

[3] 오광, "무한으로 확장하는 웹 2.0 세계," 월간 말, pp. 224-225, 2006년 9월호.

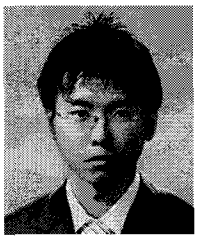
[4] 강필구, 김남중, 이예슬, 채진석, "웹 2.0을 위한 효율적인 태그 관리 시스템의 설계 및 구축," 한국정보과학회 2006 가을 학술발표 논문집, 제33권, 제2호(D), pp. 170-173, 2006.

[5] http://www.dal.co.kr/col/pcline/pcline200601_web20_1.html.
[6] http://www.web2hub.com/wiki/index.php/Web_syndication.
[7] 전종홍, 이승윤, "웹 2.0 기술 현황 및 전망," 전자통신동향분석, 제21권, 제5호, pp. 141-153, 2006.
[8] "RDF Rich Site Summary (RSS)," http://xml.coverpages.org/rss.html, 2007.
[9] Atom Publishing Format and Protocol, http://xml.coverpages.org/atom.html, 2007.
[10] http://help.yahoo.com/l/kr/yahoo/hub/hub-98623.html.
[11] R. Sinha, "A Cognitive Analysis of Tagging," http://www.rashmisinha.com/, 2005.
[12] M. Halvey and M. Keane, "An Assessment of Tag Presentation Techniques," *Proc. of The 16th International World Wide Web Conference*, pp. 1313-1314, 2007.
[13] 이치주, "온라인 연속간행자료 수집 및 보존에 관한 연구," 한국문헌정보학회지, 제41권, 제2호, pp. 359-386, 2007.
[14] 강영주, "학습객체 재사용을 위한 메타데이터 자동 수집 방안," 충남대학교 교육대학원 석사학위논문, 2006.
[15] 이영석, 조정원, 김준일, 최병욱, "주제 중심 수집기를 이용한 RSS 채널 추천 시스템," 전자공학회논문지, 제43권, 제6호, pp. 52-59, 2006.
[16] http://www.pui.ch/phred/archives/2005/04/tags-database-schemas.html.
[17] http://www.eolin.com.



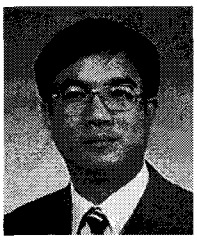
강 필 구

2005년 인천대학교 컴퓨터공학과 졸업(공학사)
2005년~2007년 인천대학교 대학원 컴퓨터공학과 졸업(공학석사)
2007년~현재 아이티플러스(주) 재직
관심분야 : 데이터베이스, XML, RSS



김 재 환

2007년 인천대학교 컴퓨터공학과 (공학사)
2007년~현재 인천대학교 대학원 컴퓨터공학과 석사과정
관심분야 : RFID, WEB2.0



채 진 석

1990년 서울대학교 컴퓨터공학과 졸업(공학사)
1992년 서울대학교 대학원 컴퓨터공학과 졸업(공학석사)
1998년 서울대학교 대학원 컴퓨터공학과 졸업(공학박사)
1992년~1997년 서울대학교 공학연구소 조교
1997년~1998년 한국학술진흥재단 부설 첨단학술정보센터 선임연구원
2006년~2007년 미국 California State University San Bernardino 방문교수
1998년~현재 인천대학교 컴퓨터공학과 부교수
관심분야 : 인터넷 소프트웨어, 전자문서 처리, 디지털 도서관

이 상 준

1996년 서울대학교 컴퓨터공학과 졸업(공학사)
1998년 서울대학교 대학원 컴퓨터공학과 졸업(공학석사)
2004년 서울대학교 대학원 전기컴퓨터공학부 졸업(공학박사)
2004년~2005년 자동제어특화연구센터 연구원
2005년~현재 숭실대학교 컴퓨터학부 조교수
관심분야 : 멀티미디어, 데이터베이스, 데이터 마이닝, P2P 시스템