2001 년 한국정보처리학회 춘계 학술발표논문집 제 8 권 제 1 호

IHWA 시스템의 반 구조적 Web 문서에서의 정보수집 서비스

정종석, 오동익
순천향대학교 정보기술공학부
e-mail : jungjs@cse.sch.ac.kr, dohdoh@sch.ac.kr

Gathering Services of IHWA from Semi-Structured Web Information Sources

Jong-Seok Jeong, Dong-Ik Oh
Division of Information Technology Engineering, College of Engineering
SoonChunHyang University

요 약

IHWA 는 분리개발 된 객체 컴포넌트를 통합함으로써 새로운 응용 시스템을 작성할 수 있는 CBSE(Component Based Software Engineering)기법을 바탕으로 한, 웹 기반 정보 저장/검색의 전형적 모델을 제공하는 소프트웨어 시스템이다. 현재 본 연구팀은 IHWA 시스템의 적용 사례로서 e-business 를 위한 쇼핑몰을 구축하고자 검색, 지불, 응용분야의 모듈들을 S/W 컴포넌트로 개발 중에 있다. 본 논문에서는 그 중, IHWA 의 검색분야에서 상품정보를 보다 효율적으로 수집하고 사용자에게 양질의 정보를 제공하기 위해 필수적으로 요구되는 정보 수집 서비스의 구조에 대하여 설명하고, 특히 비 정형화 된 XML 웹 소스에서의 정보수집 방법에 있어 현재 진행중인 연구에 대하여 설명하고자 한다.

1. 서론

IHWA 시스템[1][1]은 다양한 웹 지향 정보검색 시스템 구축을 위한 정형화된 하나의 모델을 제공하고자 개발되어지고 있는 통합형 웹 기반 정보 저장/검색 시스템이다. IHWA 는 CBSE (Component Based Software Engineering) 기법[1]에 바탕을 두고 설계되었으며, 기존의 다양한 소프트웨어 컴포넌트들을 통합하고 새로운 컴포넌트를 개발함으로써 구현되어지고 있다. 이러한 시스템의 컴포넌트는 재사용이 용이하여 웹 상의 쇼핑몰을 구축하고자 하는 개발자들은 IHWA 에서 개발/사용된 컴포넌트들을 이용하여 손쉽게 각자의 시스템을 구성할 수 있다.

IHWA 시스템의 컴포넌트 구조는 JCC(Java Commerce Client)기반의 클라이언트 컴포넌트와 EJB(Entreprise JavaBeans)기반의 서버 컴포넌트들을 사용한다. 그리고 이 두 가지 컴포넌트들 사이의 통신은 CORBA/IIOP 채널을 통하여 이루어진다.

그림 1 은 이러한 IHWA 시스템의 전체적인 구조를 보여주고 있다.

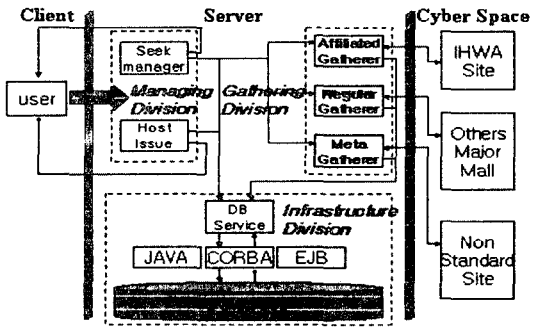

그림 1 IHWA 시스템의 구성도

[1] 본 연구는 정보통신부의 대학 전자상거래 S/W 연구 센터 지원사업에 의해 수행된 것임

IHWA 시스템에서는 6 개의 서버측 컴포넌트들이 Managing, Gathering, Infrastructure 의 3 가지 Division 으로 분리되어 제공된다. Managing Division 은 시스템에서의 동작을 컨트롤하는 역할을 담당하는데, 외부로부터의 검색요구, 데이터베이스 갱신, 정보수집과 같은 다양한 시스템 동작을 제공하고 관리한다. Gathering Division 은 3 가지의 정보수집기로 구성되어 있고 이들은 분산된 환경에서의 정보를 수집하는 역할을 담당한다. 마지막으로 Infrastructure Division 은 모든 시스템자원과 관련된 데이터 및 EJB 와 CORBA 객체에 대한 정보들을 조작하고 저장하는 역할을 담당한다.

IHWA 시스템의 Gathering Division 에서 제공하는 정보수집기는 Affiliated Gatherer 와 Regular Gatherer 및 Meta Gatherer 의 세 가지인데, 이들을 통해 IHWA 시스템은 분산된 환경에서 정보 수집 활동을 할 수 있게 된다. Affiliated Gatherer 는 서로 동일한 IHWA 시스템 구조를 채택하고 있는 사이트간의 상호 정보교환을 담당하고, Regular Gatherer 는 서로 협약 되어지지 않은 인터넷 쇼핑몰로부터 DTD 가 공개된 XML 상품 카탈로그를 수집하고 수집된 카탈로그에서 상품정보를 추출하여 데이터베이스에 저장하는 기능을 수행한다. 본 연구진에서는 이들 Gatherer 를 CORBA 미들웨어 및 인터넷 프로토콜을 사용하여 이미 구현하고 이를 보고 한 바 있다[1]. Meta Gatherer 는 웹 상에서 표준화 되어있지 않은 정보, 즉, 일반적인 웹 문서(HTML)나 DTD 가 정의되어 있지 않은 XML 문서에서 제공되는 정보를 추출하기 위하여 필요한 Gatherer 로서, 아직까지 웹에서 제공되는 대부분이 이러한 범주의 표현방식으로 제공되고 있는 것을 감안할 때 [3], 정확하고 포괄적인 정보를 제공해야 하는 검색 서비스에 반드시 필요한 모듈이라 할 수 있다. 본 논문에서는 이러한 Meta Gatherer 의 기능 중, DTD 가 정의되어 있지 않은 XML 문서에서의 정보수집 및 추출기능의 구현과 그의 활용방법에 대해 설명하고자 한다.

2. 관련연구

비정형화 된 문서에서 상품정보를 추출하기 위해 필요한 가장 핵심적인 기술부분은 문서내의 정보를 계층구조로 표현하고 이를 추출하는 것이다. 비 정형화 된 데이터 소스에서 정보 추출을 위해서는 Data Mining 분야에서 활발한 연구가 진행되고 있고, 특히 우리의 연구와 직접적인 연관을 가진 HTML 문서에서의 정보추출에 관한 작업도 Wrapper[4]를 활용한 방식 등을 통하여 활발히 진행되고 있으나 아직까지 만족 할 만한 정보 추출의 단계에는 이르지 못하고 있는 실정이다.

특히 우리가 구현한 DTD 가 제공되지 않는 XML 문서 (반정형화 된 문서)에서 DTD 를 추출하는 연구는 Bell 연구소의 XTRACT[2]와 Singapore 의 Nanyang Technological University[3], 그리고 IBM Alphaworks 의 DdbE[2]등에서도 이루어졌으며, 3 가지 연구 방법은 모두 반 정형화된 데이터 소스에서 DTD 를 추출하기 위하여 트리나 그래프를 이용하여 계층적인 구조를 생성하고 이 구조에 보다 효율적이고 함축적인 DTD 를 생성하기 위한 알고리즘을 적용하는 방법을 채택하고 있다. 이들은 모두 반 정형화된 데이터 소스들에서 유효한 정보를 추출할 수 있는 계층적인 데이터 구조를 생성하는 연구로서 우리가 필요로 하는 정보추출기와 직접적인 연관성을 가지고 있으나 전체적인 DTD 문법을 활용할 수 없다거나, 다른 프로그램 모듈에서 인터페이스가 될 수 없다는 제약점을 가지고 있다. 또한 이들은 XML 문서들의 공통된 구조를 추출하기 위하여 먼저 XML 문서 각각의 구조를 트리나 그래프를 이용하여 표현하고 이들을 통합하는 2 단계 알고리즘을 채택하고 있다. 따라서 본 연구에서는 독립된 모듈로 구현되지만 인터페이스를 통해 다른 모듈에서 활용이 가능하고 기존의 2 단계 DTD 추출 알고리즘에서 각 XML 문서의 구조를 트리나 그래프로 변환하는 단계를 삭제한 향상된 알고리즘을 적용한 DTD 추출기를 구현 할 필요성을 느끼게 되었다.

3. 본론

Meta Gatherer 는 DTD 가 제공되지 않는 XML 문서에서의 정보 추출을 수행한다. 이 모듈은 "문서수집기", "XML 정보추출 및 그룹화 모듈", "DTD Extractor", "자동 파서 생성기" 나누어지는데, 그림 2 는 Meta Gatherer 의 전체적인 구성도와 동작을 설명하고 있다.

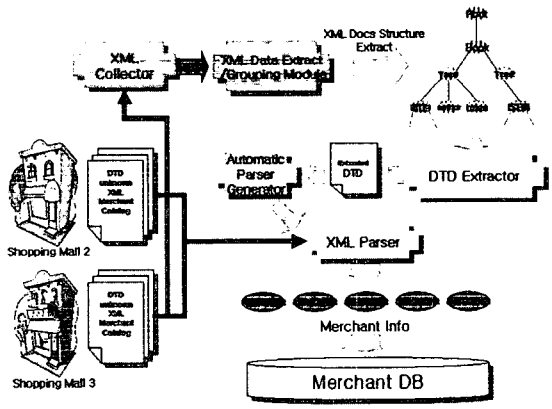

그림 2 Meta Gatherer 의 시스템 구성도

"문서 수집기"는 웹 사이트들을 정기적 또는 비정기적으로 돌아다니며 XML 수집하는 역할을 담당하며, 이는 Regular Gatherer 에서 구현된 문서수집기와 동일하게 작동한다[1]. 이렇게 수집된 XML 정보는 "XML 정보추출 및 그룹화 모듈"을 통해 트리로 표현되고, 이 트리는 계속적인 XML Input 에 의해 확장되고 그룹화 되어진다. 모든 XML Input 에 의해 트리가 완성되면 "DTD Extractor" 가 트리를 읽어 문서들에 적합한 DTD 를 생성해 내고, 이 DTD 에 기초하여 "자동 파서 생성기"가 XML 문서의 파싱을 위해 필요한 핸들러를 자동으로 생성한다. 이렇게 하여 준비된 XML 파서를 통해 XML Input 에 대한 정보추출과 하부 DB 로의 저장이 가능해 진다.

3.1 XML 정보추출 및 그룹화 모듈

여러 XML 데이터를 위한 단일 DTD 를 생성하기 위해서는 입력되는 XML 정보를 계층적 자료구조(Tree)로 표현하는 것이 필요하다. XML 데이터는 중첩된 구조로 구성되고 (well-formed XML 데이터) 이는 n-ary 트리로써 잘 표현 될 수 있으며, 이러한 트리에서 문서의 구조를 파악하기가 용이하기 때문이다. 따라서 중첩된 구조를 가진 XML 데이터를 적합하게 구조화시켜 트리로 구성한다면, 이 트리에서 데이터의 일반구조 (DTD)를 생성하는 작업이 용이해 질 것이다. 그림 3 는 XML 정보추출 및 그룹화 모듈에서 XML 문서를 읽어 들여 이를 트리로 표현하는 방법을 도식화하고 있다.

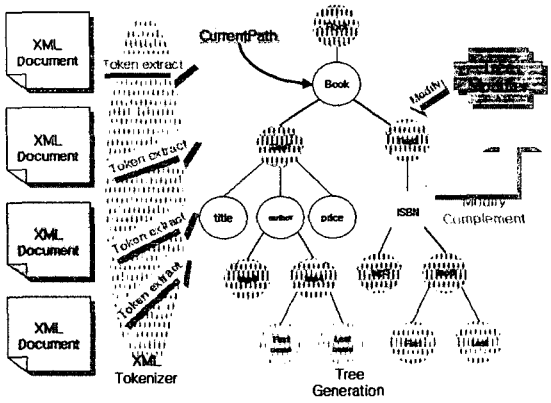

그림 3 XML 문서의 구조적 정보를 트리로 변환

XML Tokenizer 는 XML 문서를 읽어들여 StartTag 와 EndTag 에 해당하는 토큰을 추출하고 Stack 을 이용하여 추출되어지는 Element 간의 부모 자식관계를 표현한다. 또한 구축하려는 트리에는 XML Tokenizer 에 의해 추출되어진 Element 가 삽입될 위치를 가리키는 CurrentPath 라는 포인터가 존재하여 트리내의 삽입 및 Element 의 존재여부를 검사할 때 사용되어진다.

StartTag 에 해당하는 토큰이 추출되었을 때 본 모듈에서는 다음과 같은 사항을 검사한다.

- CurrrentPath 가 가리키는 노드의 자식노드 유무
- CurrentPath 가 가리키는 노드가 자식노드를 가지고 있을 때 추출되어지는 Element 의 순서와 트리로 구축된 노드들의 순서비교
- XMLTokenizer 가 전달한 부모정보와 현재 CurrentPath 가 가리키는 노드정보의 비교
- 현재의 문서처리 중 각 노드들의 방문여부

위와 같은 사항을 검사한 결과에 따라서 새로운 노드를 생성하여 트리에 추가하거나 이미 존재하는 노드의 속성을 변경시키게 된다. 또한 현재 처리중인 XML 문서내의 Element 의 순서와 구축된 트리에서 노드들의 순서가 상이할 때에는 두개의 임시노드를 새로이 생성하여 그 하부에 이들을 위치시킴으로써 이 두 그룹사이에 OR 관계가 있음을 표현한다. 동시에 새로운 노드의 추가나 노드의 속성 변경 시에 CurrentPath 의 현재 위치를 생성한 노드나 존재하는 노드로 변경함으로써 이후에 있을 XML Tokenizer 가 추출한 노드의 검색 및 추가되어질 위치를 가리키게 한다.

EndTag 에 해당하는 토큰이 추출되었을 때 단순히 CurrentPath 의 위치를 그 부모노드로 이동함으로써 현재의 노드에 대한 자식노드의 삽입 등의 작업이 완료되었음을 나타낸다.

이러한 방법으로 기존의 2-Path 알고리즘[2,3]에서 각 XML 문서의 구조를 트리나 그래프로 표현하는 중간 단계를 생략하고 바로 XML 문서들에서 트큰을 추출하여 그 토큰에 해당하는 노드들을 트리에 추가하거나 갱신함으로써 XML 정보추출 및 그룹화 모듈은 수집된 모든 XML 문서들의 중첩된 구조를 포함하는 트리를 1-Path 로 구성하게 된다.

본 논문에서는 Oracle 의 XMLTokenizer 를 사용하여 XML 문서에서 토큰을 추출하는 모듈을 구현하였고 Java Swing 컴포넌트인 DefaultTreeModel 과 DefaultMutableTreeNode 를 사용하여 트리와 트리의 노드를 표현하였다. 또한 구축된 트리는 바로 Jtree 컴포넌트로 전달하여 화면에 출력할 수 있으며 이러한 GUI 를 이용하여 구축된 트리에 대한 디버깅과 전체 XML 문서 구조의 이해를 쉽게 하였다.

3.2 DTD Extractor

DTD Extractor 는 XML 정보추출 및 그룹화 모듈에서 생성된 트리를 실제 XML DTD 파일로 변환 시켜주는 모듈이다. 구축되어진 트리는 XML 문서의 계층적 구조와 각각의 부모와 자식간의 관계를 그대로 포함하고 있기 때문에 DTD Extractor 는 트리를 Level 별로 순회하면서 노드들의 순서와 부모 Element 와의 관계 즉 각 노드에 표현된 Mandatory, Optional, OR, Repetition 과 같은 속성을 '?', '|', '*', '+' 등의 XML 연산자를 사용하여 DTD 파일을 생성하게 된다. 단 트리의 노드들 중에서 실제 XML 문서에는 존재하지 않지만 OR 관계를 표현하기 위해 생성한 임시 노드에 대해서는 두 개의 쌍을 이루는 임시 노드대신에 그 노드의 서브 트리의 내용을 대체함으로써 DTD 의 OR 관계를 표현하게 된다.

3.3 자동 파서 생성기

XML 문서수집기가 수집한 문서에 포함된 데이터들은 추출되어 DB 에 저장되어야 한다. 이러한 동작을 수행하기 위해서는 DTD 에 기초한 XML Parser Handler 를 새로이 작성했어야 했다. 하지만 우리의 경우 DTD 가 새로운 문서를 수집할 때마다 갱신되므로 이 때마다 새로운 Parser Handler 를 작성하는 것은 매우 번거로운 일이다. 자동 파서 생성기는 DTD Extractor 가 생성한 DTD 를 사용하여 파싱핸들러를 자동으로 생성하고 이것을 바탕으로 각각의 XML 문서를 파싱하여 포함되어진 각각의 상품정보를 DB 에 저장하는 일을 자동으로 담당하게 된다. 그림 4 는 자동 파서 생성기의 구조 및 동작에 대하여 설명하고 있다.

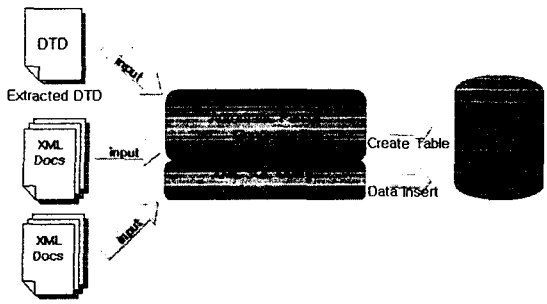

그림 4 자동 파서 생성기의 구조 및 동작

자동 파서 생성기의 전체적인 동작을 설명하면 먼저 Oracle 의 XDK(XML Development Kit)에서 제공하고 있는 "XML SQL Utility for Java"도구를 사용하여 추출되어진 XML DTD 의 내용에 부합하는 테이블을 생성한다. DB 에 테이블이 생성되었으면 수집되어진 XML 문서를 해당 테이블에 저장하여야 하는데 이러한 동작 또한 필요한 파싱핸들러를 "XML SQL Utility for Java"도구 내부에서 자동으로 생성해 줌으로써 수행되어 진다. 즉 자동 파서 생성기는 수집되어진 문서를 바탕으로 추출한 DTD 를 가지고 Oracle DB 엔진의 유틸리티를 활용하여 자동으로 DB 에 상품정보를 저장하는 일을 수행하게 된다.

4. 결론

IHWA 시스템의 gathering division 이 보다 광범위한 정보를 수집하기 위해서는 Meta Gatherer 의 역할이 중요하다. 우리는 Meta Gatherer 의 DTD 생성 모듈과 자동 파서 생성기의 구조를 설계하고 Java 언어를 사용하여 구현하였다.

Meta Gatherer 에서 가장 중요한 모듈인 XML 정보수집 및 그룹화 모듈은 기존의 2-Path[2,3]알고리즘에서 수집된 XML 문서 각각을 트리나 그래프로 표현하는 단계를 생략하고 곧바로 각 XML 문서에 포함되어진 Element 를 추출하여 트리에 추가함으로써 향상된 I-Path 알고리즘을 사용하여 구축하였다. 또한 구축되어진 트리를 화면으로 출력함으로써 구축되어진 트리의 유효성 검사를 위한 디버깅 및 전체 XML 문서들의 구조를 쉽게 파악할 수 있도록 구현하였다. Meta Gatherer 는 IHWA 의 다른 컴포넌트들과의 상호작용을 위하여 프로그래밍 인터페이스를 제공하며 이를 EJB 컴포넌트로 구현하여 서버측 컴포넌트로써 다른 인터넷 응용 프로그램에서 XML 문서의 정보수집 및 추출 컴포넌트로써 사용되어질 수 있다.

5. 향후 연구과제

"XML 정보추출 및 그룹화 모듈"에서 산출된 트리는 상당히 복잡한 구조를 갖는다. 또한 이러한 복잡한 트리구조는 DTD 로 변환 시 복잡하고 DTD 의 가독성이 떨어지는 원인이 된다. 이에 대한 해결책으로 bell 연구소의 XTRACT 나 IBM 의 DDbE 에서는 독자적인 알고리즘을 사용하여 효율적이고, 간단하게 DTD 간략화 시키는 단계를 수행하고 있다. 이와 같이 생성된 트리를 순회하면서 트리의 구조를 단순화시킬 수 있는 연구가 필요하다. 현재 구현된 Meta Gatherer 에서는 단순한 몇 가지 경우에 대하여 구현되어 있으나 이를 DTD 의 가독성 및 단순성을 좀더 높일 수 있도록 개선시키는 작업이 있어야 한다.

또한 IHWA 시스템의 Gathering Division 의 세가지 정보수집기가 서버측 컴포넌트로써 혹은 Java Beans 와 같은 클라이언트측 컴포넌트로써 다른 인터넷 정보제공 시스템을 개발하는 개발자가 손쉽게 사용할 수 있도록 구현되어진 세가지 정보수집기를 EJB 컴포넌트 및 JCC 로 구현하는 작업을 진행 중에 있다.

참고문헌

[1] Dong-Ik Oh and Jong-Suk Jung Effective Web-Based Information Gathering Services of IHWA. Proceedings of ICEIC'2000 International Conference, Shenyang, China, 2000.8 pp. 202-205.

[2] M. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, and K. Shim. XTRACT: A System for Extracting Document Type Descriptors from XML Documents. Bell Labs
Tech. Memorandum, 1999.

[3] Re-engineering Structures from Web Documents by C.-H. Moh, E.-P. Lim, W.-K. Ng. Proceedings of the 5th ACM International Conference on Digital Libraries (DL2000), San Antonio, Texas, USA, June 2-7, 2000.

[4] N. Ashish and C. A. Knoblock. Semi-automatic wrap-per generation for internet information sources. In Proceedings of Coopis Conference, 1997.

[5] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo. Extracting semi-structured data from the web. Proceedings of Workshop on Management of
Semi-structured Data, pages 18-25, 1997.

[6] N. Kushmerick, D. Weil, and R. Doorenbos. Wrapper induction for information extraction. In Proceedings of Int. Joint Conference onArtical Intelligence (IJCAI), 1997.

[7] Cay S. Horstmann, Gary Cornell, Core Java Volumn Ⅰ-Ⅱ, Sun Microsystems Press, 1998

[8] Tom Valesky, Enterprise JavaBeans - Developing Component-Based Distributed Applications, Addison-wesley, 1999