원격 분산처리에 의한 효율적인 정보수집 시스템

공용해* 최인석**
순천향대학교 정보기술공학부* 홍성기능대학 전자계산기과**

Efficient Information Extracting System using Remote Distributed Processing

Yong Hae Kong* In Seok Choi**
Div. of Information Technology Engineering, Soonchunhyang University*
Dept. of Computer, HongSung Polytechnic College**

요 약

사용자에게 제공할 정보 수집의 효율을 증대하기 위하여 Java 기반 정보 수집 이동 에이전트 시스템을 구현하였다. 정보수집 이동 에이전트는 원격 사이트로 이동하여 XML 문서를 파싱하고, 정보를 추출하여 호스트의 데이터베이스에 저장하도록 한다. 이동 에이전트는 원격 사이트에서 XML파서를 활용하여 필요한 정보만을 수집하여 전송하므로 네트워크의 부하를 줄일 수 있음과 동시에 호스트의 처리 부하를 크게 줄일 수 있을 뿐만 아니라 향후 원격 사이트의 고유한 문서 특성에 적합한 정보 추출이 가능하도록 확장할 수 있다.

1. 서론

사용자에게 다양한 양질의 정보를 제공하기 위하여 정보수집 이동 에이전트를 개발하기로 하였다. 정보수집 이동에이전트는 원격 사이트에서 적당히 구성된 된 DTD와 XML 문서를 파싱하여 유용한 정보를 추출한다. 추출된 정보는 SQL 정보로 변환하여 원래의 호스트로 보내서 데이터베이스에 저장하도록 하고 파견된 에이전트는 호스트로 되돌아오거나 소멸된다. 정보수집이동에이전트는 원격 사이트에서 필요한 정보만 추출하여 전송되므로 최소한의 호스트의 실행시간만 필요하며, 네트워크의 부하를 줄일수 있다. 따라서, 정보수집 이동 에이전트는 스파이더 방식보다 효율적이다. 우리는 정보수집 이동 에이전드를 구현하기위하여 IBM 에서 만든 자바기반 Aglets을 사용한다. Aglets은 atp 프로토콜을 이용하여 정보 전송 및 에이전트를 파견하고 관리한다. 그림 1은 정보수집 이동 에이전트 시스템의 구성도이다.

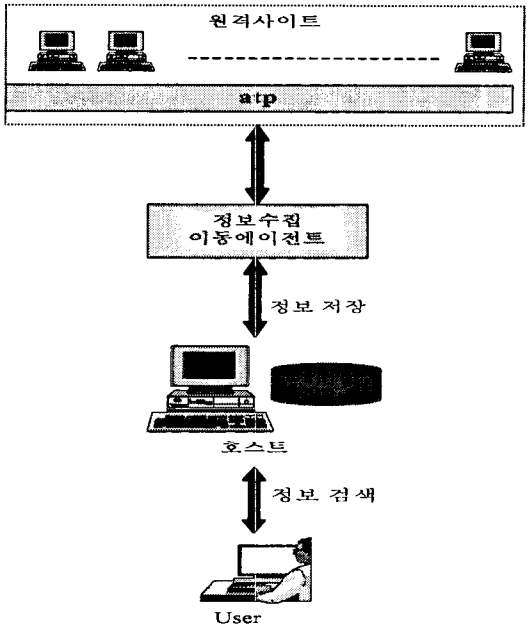


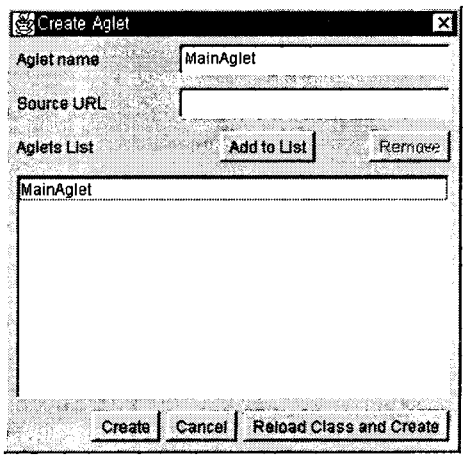
그림 1 정보수집 이동 에이전트 시스템 구성도

본 연구는 정보통신부의 ITRC 사업에 의하여 수행된 것임

2. 이동 에이전트 시스템

이동 에이전트란 사용자를 대신하여 이 기종 분산 환경의 네트웍에서 자율적으로 이동 및 반응하면서 필요한 정보를 수집하는 소프트웨어를 의미한다[1]. 이 기종 분산 환경의 네트웍에서 이동이 원활한 Java 기반 이동 에이전트 시스템은 Java 가상머신(Java Virtual Machine)상에서 구현된 이동 에이전트 시스템이다[2].

Java기반 이동 에이전트 시스템의 개발을 위하여 IBM사의 Tokyo Research Lab에서 Aglets Workbench를 개발하였다. Aglets Workbench는 실행 환경과 함께 이동 에이전트 프레임워크를 Java 클래스 라이브러리 형태로 제공하여 에이전트 개발을 쉽게 하였다. Aglets은 Tahiti라 불리는 GUI 기반의 Java 응용프로그램이다[3]. 이동 에이전트에서는 웹서버를 두지 않고 Tahiti를 이용한다. 또한 이동 에이전트 객체를 원격 사이트로 보낼 때에도 Tahiti를 이용한다[4]. 그림 2는 Tahiti에서 에이질렛을 생성시키는 화면이다.



3. 이동 에이전트 시스템의 구축 환경

본 연구에서는 다양한 양질의 정보수집을 위하여 약간의 제약 조건을 갖는 원격 사이트에서 필요한 정보를 추출하여 사용자에게 제공하는 정보수집 이동 에이전트시스템을 구축하였다.

정보수집 이동에이전트는 원격 사이트에서 표준화된 상품 정보만을 수집하여 데이터베이스에 저장하는 기능을 수행한다. 상품정보 표준화는 각각의 상품 종류마다 DTD를 구성하게 하고, XML 데이터로 표현되도록 하여 문제를 해결하였다[5]. XML문서를 분석하여 각 테그의 의미를 파악하고 필요한 정보를 추출하는 일은 파서가 담당한다. XML 파서의 기능은 첫째 XML 문서의 선언부와 프롤로그 정보를 이용해 문서를 해석하는 기능을 제공한다. 둘째, 문서의 파싱을 통해 얻어진 엘리먼트나 엔터티의 구성정보, 현재 파싱되고 있는 문서의 위치 및 파싱 상태 정보, 파싱이 끝난 후 재가공 된 문서 정보(문서의 트리나 DTD 트리 등)를 제공하여 XML 애플리케이션에서 활용할 수 있는 XML 데이터 구조 제공한다[6]. 이러한 파서의 기능을 갖는 API는 DOM과 SAX, ElementHandler가 있다.[7] DOM은 XML 문서를 하나의 트리로 표현하며 각 노드들은 엘리먼트와 텍스트(Text)로 이루어진다. Dom은 XML문서를 구조적으로 변경할 때와 메모리 안에 있는 문서를 다른 응용프로그램과 공유할 때 유용하다. SAX는 자료구조를 생성하지 않고 XML문서의 구성요소의 시작과 끝과 같은 이벤트를 생성한다. 프로그래머가 일어날 수 있는 이벤트를 설정해 놓으면, SAX는 그 이벤트가 일어났을 때 제어권을 가지고 직접 상황을 처리한다. SAX는 XML 문서를 앞에서 뒤로 읽어가면서 어떤 요소, 속성 등을 파악하여 제공해 준다. SAX는 XML 문서를 정보의 흐름으로 바꾸기 위하여 문서의 처음에서 끝까지 순차적으로 처리한다. 따라서 SAX를 이용하는 응용프로그램은 구성요소의 속성을 얻어내는 즉시 처리되어야 한다. SAX는 메모리에 합당하지 않는 사이즈가 큰 문서를 다룰때나 혹은 문서의 구조와 무관한 작업을 시행할 때 유용하다. ElementHandler는 이벤트 구동형이지만 DOM처럼 트리를 생성한다. 따라서 이벤트 구동형 작업이면서 구성요소의 내부구조를 다룰 필요가 있는 작업에 적합하다.

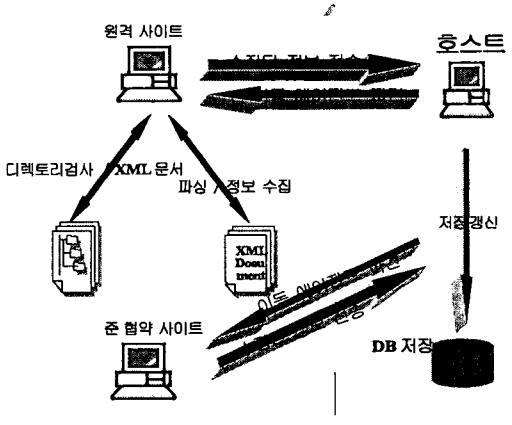


그림 3 이동에이전트 시스템의 기능

본 연구에서는 SAX를 활용하였다. 그림 3은 이동 에이전트의 기능을 도식화하였다.

4. 정보 수집 이동 에이전트 시스템

본 연구에서 구현에 필요한 오퍼레이션은 생성, 복제, 파견, 회수, 활성화, 비활성화, 제거가 있다. 생성은 컨텍스트라고 불리는 서버의 주소와 이름으로 구성된 작업실에서 발생한다. 컨텍스트에서 자바의 이동 객체인 Aglets이 생성되면서 식별자가 주어지고 초기화된다. 초기화가 되면 Aglets은 곧 실행을 시작한다. Aglets의 복제는 같은 컨텍스트안에서 원래의 Aglets과 똑같은 Aglets을 복사하는 것이다. 그러나 이 복사된 Aglets은 자신의 식별자를 가지며 실행을 다시 시켜야 한다. Aglets의 파견은 현재의 컨텍스트에서 해당 Aglets을 제거하고, 이동될 컨텍스트안에 해당 Aglets을 추가하여 실행시킨다는 의미이다. 회수란 특정 Aglets이 현재의 컨텍스트에서 제거되고 해당 Aglets을 요청한 컨텍스트안에 추가된다는 의미로 파견에 비해 수동적이다. 비 활성화는 Aglets의 실행을 임시로 중지한 후 상태를 디스크에 저장하는 것을 말한다. 활성화는 이러한 Aglets을 같은 컨텍스트안에서 다시 실행함을 말한다. Aglets의 제거는 현재 실행을 중지시키고 그 Aglets이 있는 컨텍스트로부터 Aglets을 제거하는 것이다. 그림 4는 Aglets의 생명주기를 나타낸 그림이다.

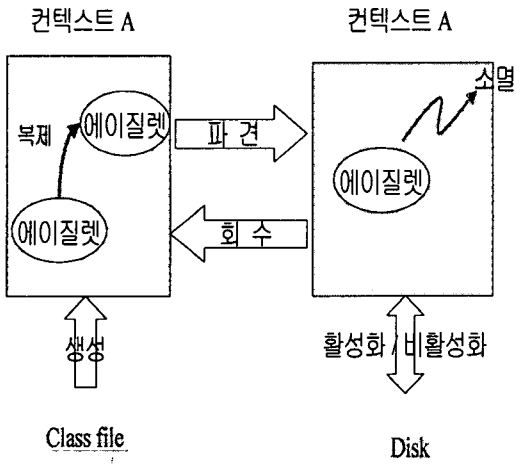


그림 4 Aglets의 생명주기

본 연구에서 구현한 정보 수집 에이전트 시스템은 디렉토리 리스팅, XML 파싱, 정보 수집 등의 작업을 원격 사이트에서 수행하도록 하였다.

호스트는 HOST 에이전트와 Remote 에이전트를 생성하고 현재의 컨텍스트 정보를 확보하여 파견된 에이전트를 회수하거나 추출된 정보를 원래의 호스트로 전송할 때를 대비한다. Remote 에이전트는 원격 사이트에 파견되어 디렉토리에서 확장자가 XML인 문서 파일을 찾는 작업을 수행한다.

XMLparser는 SAX를 활용하여 찾아진 XML 문서의 요소와 속성을 파악한다. 그리고 Remote 에이전트는 추출된 정보를 SQL로 변환하여 호스트에 전송하고 host 에이전트에 의하여 시스템의 데이터베이스에 저장하도록 한다. 모든 정보 수집 활동이 완료되면 수집된 정보를 전송하고, 호스트로 귀환 또는 제거하도록 하였다. 그림 5는 정보수집 이동에이전트 시스템의 상세도이다.

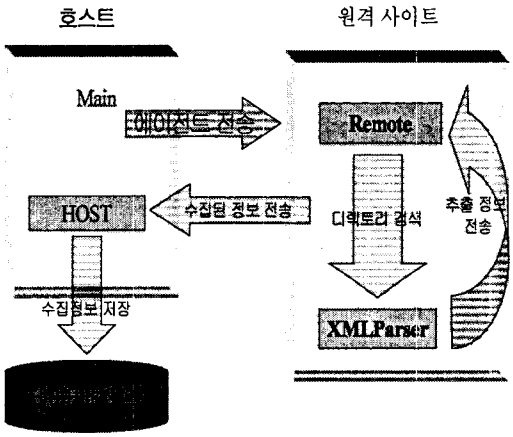


그림 5 정보수집 이동에이전트 시스템의 상세도

그림 6은 구현된 정보 수집 이동 에이전트의 가상 코드(pseudo code)이다.

호스트 :

1. host 에이전트 생성
2. Remote 에이전트 생성

host 에이전트 :

3. 사건발생 감시
4. 에이전트 행동 개시
5. 현재 호스트 URL 확보
6. Remote 에이전트 파견
7. 전송된 정보를 DB에 저장 (SQL)
8. 에이전트 제거

원격 사이트 :

Remote 에이전트

1. 사건 발생 감시
2. 행동 개시
3. 디렉토리 리스팅
4. XML문서 파싱
5. 정보 수집
6. SQL문 생성
7. SQL문 호스트 7로 전송
   (정보 전송)

7. 에이전트 제거

그림 6 이동 에이전트 가상코드

이렇게 구축된 이동 에이전트 시스템은 원사이트에 파견되어 정보를 수집하고 SQL문을 생성하여 원래의 호스트로 전송한다. 이러한 정보 수집은 정보 전송량을 최소화하여 네트워크의 부하를 줄일 수 있으며 호스트의 부하를 크게 줄일 수 있었다.

5. 결론

사용자에게 보다 다양한 정보를 제공하기 위하여 원격 사이트에서 정보를 수집해주는 이동 에이전트 시스템을 구현하였다.

본 연구에서 구현한 정보수집 이동에이전트 시스템은 에이전트를 생성하여 원격 사이트에 파견한 후, 원격 사이트의 디렉토리에서 XML문서를 검색한다. 검색된 XML문서는 XMLparser를 이용하여 필요한 상품정보만을 추출하여 정보를 수집한다. 수집된 상품정보는 SQL정보로 변환하여 원래의 호스트로 전송 후 DB에 저장하도록 한다. 파견된 이동에이전트는 원격 사이트에서의 정보수집 활동이 완료되면 호스트로 귀환시키거나 제거하였다. 이동에이전트에 의한 정보수집은 웹 스파이더 방식의 중앙 정보 수집의 단점인 과다한 통신 부하와 처리 부하를 완화할 수 있었다. 또한 이러한 정보수집 이동에이전트는 원격 사이트의 특수한 형태의 정보 수집에 적합하도록 확장 가능하므로 향후 정보 수집의 질적 향상을 기대할 수 있다.

[참고문헌]

[1] 석황희, 김인철 "이동 에이전트의 개념과 응용", 한국멀티미디어학회, 제3권, 제2호, pp29-39, 1999.

[2] http://www.alphaworks.ibm.com/formula.

[3] http://www.trl.ibm.co.jp/aglets.

[4] Danny B. Lange, "Programming an Deploying Java Mobile Agents with Aglets", Addsion Wesley, 1998.

[5] http://xml.t2000.co.kr/products/intro_parser.html.

[6] http://maso.zdnet.co.kr/maso/2000/11/02/014005,973154270,179.html

[7] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto, "XML and JAVA", Addsion Wwsley, 1999.