ebXML 3.0 레지스트리를 위한 Federation 설계*

이병현°, 김상균, 이동헌, 이규철
충남대학교 컴퓨터공학과
{bhlee°,skkim, dhlee, kclee}@ce.cnu.ac.kr

Design of Federation for ebXML 3.0 Registry

Byung-Hyun Lee°, Sang-Kyun Kim, Dong-Heon Lee, Kyu-Chul Lee
Dept. of Computer Engineering, Chungnam National University

요 약

ebXML은 UN/CEFACT와 OASIS가 주축으로 제정한 전자상거래 표준으로 XML 기반의 인프라를 제공하여 글로벌 e-비즈니스가 가능하도록 하는 것을 목적으로 한다. 최근 여러 ebXML 레지스트리간의 Cooperation이 중요해지면서 2003년 6월에 제정된 ebXML RIM(Registry Information Model) 버전 2.5와 ebXML RS(Registry Service) 버전 2.5 스펙**에서는 Federation, Object Relocation, Object Replication을 포함하는 레지스트리 간의 Cooperation 기능과 더불어 Content-based 질의, Event Notification, Iterative 질의와 같은 기능들이 새로 추가되었다. 따라서, 본 논문에서는 ebXML 버전 3.0 레지스트리를 구현하기 위해서 ebXML 버전 3.0 레지스트리의 시스템 구조를 설계하였으며, 버전 3.0에서 가장 중요한 기능인 Federation을 효율적으로 관리하는 방안을 분석, 설계하고 사용자가 보낸 Federation에 대한 질의를 처리하기 위한 알고리즘을 설계하였다.

1. 서 론

최근 들어, 각 표준 단체나 기업들은 인터넷 기반의 B2B 전자 상거래를 위한 다양한 프레임워크를 연구, 개발하고 있으며, 제안된 프레임워크들은 각 기업들의 비즈니스에 관련된 기업정보 및 서비스 정보 등을 저장하여 여러 사용자 및 어플리케이션들이 공유할 수 있는 레지스트리를 기반으로 다양한 서비스를 제공하고 있다.

이러한 프레임워크 중에 ebXML[1]은 UN/CEFACT과 OASIS가 주축이 되어 제정한 전자상거래 표준으로 'Creating A Single Global Market'이라는 기치 아래 XML을 이용하여 인터넷 기반의 e-비즈니스가 가능하도록 만들어졌다. ebXML 표준은 2001년 5월에 버전 1.0 이 발표되었으며, 그 후에도 웹 서비스와 같은 다양한 표준을 ebXML에서 수용하는 등 표준의 가치를 높이기 위한 연구가 진행 중이다. 특히 ebXML RIM 버전 2.5[2]와 RS 버전 2.5[3]는 OASIS ebXML 기술 위원회에 의해 2003년 6월에 Draft가 제정된 상태이다.

ebXML 버전 3.0에는 여러 레지스트리들의 Cooperation을 위해 Federation, Object Relocation, Object Replication 기능이 포함되었으며, 또한 사용자가 관심있는 정보에 이벤트가 발생하였을 때 이를 통보해주는 Event Notification, 레지스트리 객체 내용에 대한 검색을 제공하는 Content-based 질의, Iterative 질의 등이 새로 추가되었다.

본 논문에서는 레지스트리들의 Cooperation 중에서도 가장 중요한 기능인 Federation의 생성과 그 생성된 그룹에 효과적으로 질의를 전송하는 방법에 대해 알아보고자 한다.

2. 관련 연구

ebXML 버전 3.0에 새로 정의된 기능은 다음과 같다.

■ Federation을 통한 레지스트리들의 협력은 Peer to Peer 기반으로 여러 레지스트리들을 결합함으로써 하나의 레지스트리에 질의를 전송하여도 여러 레지스트리에서 검색 결과를 가져오는 효율적인 검색이 가능하다.

■ Object Relocation Protocol은 임의의 한 레지스트리에서 다른 레지스트리로, 또는 한 레지스트리 내의 특정 사용자에서 다른 사용자로 레지스트리 객체의 소유권을 변경하는 것이다.

■ Object Replication은 다른 레지스트리에 저장된 레지스트리 객체를 로컬 레지스트리에 복제하는 기능이다. 그럼으로써 객체에 대한 접근 시간을 줄이고 원래 소유의 레지스트리 서버가 시스템 오류 등으로 인해 동작하지 않을 경우에도 원활히 수행된다는 효과를 얻을 수 있다.

■ 사용자는 레지스트리 내의 변경 사항이나 CPP의 갱신 등과 같이 레지스트리에서 발생하는 모든 이벤트를 Event Notification 서비스를 통해 받아 볼 수 있다.

■ 컨텐트 기반의 질의는 레지스트리 객체 내용에 대한 검색을 제공한다.

■ 검색된 결과의 수과 많을 경우 효과적인 처리가 가능하도록 Iterative 질의도 새롭게 추가되었다.

3. Federation 구성

Federation은 여러 개의 레지스트리를 하나의 그룹으로 묶는 것을 말하는데 이 경우 각 레지스트리를 물리적으로 결합하는 것이 아니라 분산된 각각의 레지스트리를 loosely-coupled하게 구성해서 사용자들이 이 Federation을 통해 여러 ebXML 레지스트리들을 하나의 커다란 논리적인 레지스트리로 다룰 수 있게 한다. 그림 1은 두 레지스트리가 하나의 Federation으로 구성되는 간단한 예를 보이고 있다.

Federation은 Peer-to-Peer 모델을 따르기 때문에 하나의 Federation에서 이 Federation에 속한 레지스트리들은 모두 동등한 역할을 가진다. 따라서 어떤 Submitting Organization도 Federation을 생성, 참가, 탈퇴, 삭제를 할 수 있으며 하나의 레지스트리는 여러 개의 Federation에 속할 수도 있다. 이러한 Federation의 가장 큰 장점으로는 federated 질의를 지원한다는 점이다. 사용자가 어떤 Federation에 속한 하나의 레지스

* 본 연구는 한국과학재단 지정 충남대학교 소프트웨어연구센터(SOREC)와 정보통신부의 대학 IT연구센터(ITRC) 지원을 받아 수행되었다.

** ebXML RIM 버전 2.5와 RS 버전 2.5 스펙은 ebXML 3.0 레지스트리의 기능을 지원하기 위해 제정되었다.

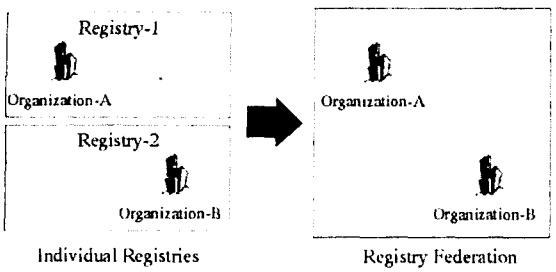

그림 1 레지스트리 Federation

트리에 federated 질의를 하게 되면 이 질의는 이 레지스트리가 속한 Federation의 모든 멤버에 똑같은 질의가 분산 처리되어서 질의를 받은 레지스트리로 결과가 전달된다. 따라서 사용자는 똑같은 질의를 여러 레지스트리에 여러 번 보낼 필요 없이 federated 질의를 통해 보다 효율적인 검색을 할 수 있게 된다.

본 논문에서는 이러한 ebXML 레지스트리의 Federation을 효과적으로 관리하고, Federated 질의를 전송하는 메카니즘을 설계하기 위하여 다음과 같은 방법을 고려하였다.

3.1 Hypercube 토폴로지

Hypercube[4]는 네트워크 상의 여러 노드들을 연결하는 방법 중 최선의 브로드캐스팅과 검색 방법을 지원한다. 이 방법으로 Federation을 구현하면 federated 질의 메시지를 수신한 레지스트리는 자신과 연결된 이웃 레지스트리에만 메시지를 전송하므로 네트워크의 부하를 최소로 할 수 있다는 장점을 갖는다. 그러나, Federation을 구성하는 ebXML 레지스트리 간에 Hypercube 토폴로지를 유지하기 위해서는 스펙에 정의되지 않은 부가적인 데이터베이스 스키마와 서버 모듈을 추가해야 한다. 즉, 임의의 한 레지스트리에만 Hypercube 모듈을 추가해서는 스펙에 준하는 다른 레지스트리와 통신이 불가능하며, 또한 스펙에 준하지 않으므로 효과적인 설계 방안이 될 수 없다.

3.2 JXTA

JXTA[5]는 현재 개발 중인 많은 P2P 시스템의 원활한 동작을 목적으로 개발되었으며, 서로 다른 이기종 P2P 서비스의 Peer간에 상호 운용이 가능한 환경을 제공한다. 또한 XML 문서를 기반으로 개발되었기 때문에 개발 언어, 통신 프로토콜, 시스템 플랫폼에 독립적이며, 네트워크에 접속 가능한 모든 디바이스 상에 구현 가능한 환경을 제공한다.

JXTA에서 메시지의 전송은 non-centralized 방식이며 JXTA hub에 의해 통신이 이루어진다. JXTA hub Peer는 주로 지역적 특성에 의해 선정되는데, hub로 설정된 노드는 JXTA 기반의 Federation에서 다른 Peer에게 메시지를 전송하는 Super Peer의 역할을 수행하게 된다. 그러나, 이와 같은 접근은 ebXML 레지스트리간의 공통된 관심사에서 비롯된 Federation이 아니라 지리적 요구에 의해 생성될 수 있으므로 공통된 관심사를 갖는 레지스트리들끼리 Federation을 구성한다는 ebXML 스펙에서의 취지에 어긋나며, 모든 ebXML 레지스트리 위에 JXTA 엔진을 설치해야 한다는 단점을 갖는다.

3.3 Parallel-distributed 브로드캐스팅

이 방법은 ebXML 스펙에 명시된 방법으로 federated 질의를 수신한 ebXML 레지스트리가 Federation 그룹의 다른 모든 레지스트리들에게 그 질의를 재 전송한다. 같은 Federation 그룹에 속해 있다 하더라도 지리적으로 멀리 떨어진 레지스트리에 전송해야 하는 경우 브로드캐스팅을 하기 때문에 네트워크 부하가 커질 문제점이 있으나, 추가적인 모듈이 필요 없는 가장 유연한 방법이라 할 수 있다.

3.4 Federation 메카니즘의 비교

Hypercube 네트워크 토폴로지를 사용하는 방법은 네트워크의 부하를 최소로 하는 장점이 있으나, ebXML 레지스트리의 구조를 변경해야 한다. 또한, JXTA를 이용하는 방법은 JXTA hub를 중심으로 Federation을 생성하여 JXTA 엔진 기반으로 메시지를 송, 수신하기 때문에 Federation 그룹 내에서의 메시지 전송을 효과적으로 처리할 수 있으나, 모든 레지스트리에 JXTA 엔진을 설치해야 하는 단점이 있다. 각 국가별로 대표 레지스트리를 두어 JXTA의 hub로 설정하고, 각 국내의 다른 레지스트리에 메시지를 전송하도록 하며, 각 국가간 대표 레지스트리들은 Hypercube 토폴로지에 따라 구성되도록 ebXML 레지스트리의 구조를 변경하면 가장 이상적인 방법이 된다. 그러나, 모듈의 추가와 JXTA 엔진을 모든 레지스트리에 반드시 설치해야 하는 단점으로 인해 본 논문에서는 네트워크 부하를 감수해야 하지만, 추가적인 모듈이 필요 없으며, ebXML 버전 3.0에서 규정하고 있는 parallel-distributed 브로드캐스팅 방법을 설계하고자 한다.

4. Federation 설계

4.1 ebXML 3.0 레지스트리의 시스템 구조

그림 2는 본 논문의 전체 시스템 구조이다.

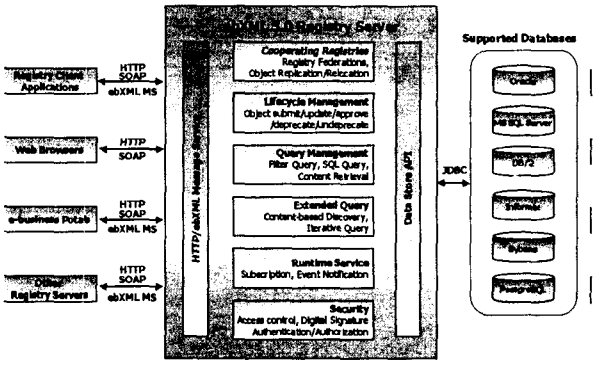

그림 2 시스템 구조

ebXML 레지스트리 버전 3.0 시스템은 크게 클라이언트, 레지스트리 서버, backend 데이터베이스 이렇게 3-tier로 나누어진다. 클라이언트는 HTTP와 SOAP 또는 ebXML MS(Message Service)를 통해 레지스트리 서버와 접속할 수 있으며, backend에서는 JDBC를 통해서 어떠한 데이터베이스와도 연결이 가능하다. 또한 레지스트리 서버는 기존의 레지스트리 버전 2.0에서 지원되었던 Lifecycle management와 Query Management 기능과 버전 3.0에서 추가된 Cooperating Registry, Extended Query, Runtime Service와 같은 기능이 포함되어 있다.

4.2 Federation 참가 요청 알고리즘

```
알고리즘 : 레지스트리 간의 Federation을 참가요청

Input : Federation 참가 요청 메시지
Output : Federation 참가 요청에 대한 응답 메시지

단계 :
레지스트리i(Ri)의 운영자(Oi)가 R1에서 F1을 검색;
while true do
   if Extramural Association 메시지를 통해
      Oi가 F1에 참가 요청 메시지 수신 then
      if Oi == Ri의 운영자 then
         confirmation 요구 메시지를 Ri에 송신;
         if Ri로부터 confirmation 메시지 수신;
            then F1에 Ri 추가;
            return Success 응답 메시지;
         end if
         else return Failure 응답 메시지;
      end if
      else return Failure 응답 메시지;
   end if
end while
```

그림 3 Federation 참가 요청 알고리즘

위의 알고리즘에 나타나듯이 Federation의 생성 및 참가 요청은 각 레지스트리를 관리하는 운영자에 의해 수행된다. 운영자는 자신의 관심사와 유사한 Federation을 가진 레지스트리에 참가 요청을 하며 Confirmation 과정을 거친 후 그룹의 멤버로 승인된다.

4.3 Federation 질의 알고리즘

```
알고리즘 : Federation 그룹 내에서 질의를 처리

Input : Federated 질의 메시지 Q1
Output : Federated 질의에 대한 ResultSet
단계 :
while true do
   if 레지스트리1(R1)에서 질의 메시지(Q1)을 수신
   then
      if Q1 == Local 질의 then
         Local 레지스트리에서 검색 수행;
         return 레지스트리 검색 결과;
      end if
      else if Q1 == Federated 질의 then
         Q1을 Federation 그룹에 속한 모든 레지스트
         리에 전송;
         각 레지스트리의 질의 결과를 수신;
         return 취합된 ResultSet;
      end if
   end if
end while
```

그림 4 Federation 질의 처리 알고리즘

Federation의 한 멤버가 질의 메시지를 수신하면 우선 Local 질의인지 또는 Federated 질의인지를 검사를 한다. Federated 질의일 경우에만 Federation의 그룹에 속한 레지스트리에 질의 메시지를 전달한다.

5. 결론 및 향후 연구 방향

본 논문에서는 ebXML 버전 3.0에서 새롭게 정의된 기능인 Federation에 참가 요청하며, 질의를 전송하는 메카니즘을 설계하였다. 이를 위해 ebXML 레지스트리 서버의 시스템 구조를 분석하고, 최적의 Federation을 설계하기 위한 여러 가지 방법을 분석하였다.

각 국가별로 대표 레지스트리를 두어 JXTA의 hub로 설정하고, 각 국내의 다른 레지스트리에 메시지를 전송하도록 하며, 각 국가간 대표 레지스트리들은 Hypercube 토폴로지에 따라 구성되도록 ebXML 레지스트리의 구조를 변경하면 가장 이상적인 방법이 된다. 그러나, 데이터베이스 스키마, 서버 모듈의 추가와 JXTA 엔진을 모든 레지스트리에 반드시 설치해야 하는 단점이 있다. 그에 비해 Parallel-distributed 브로드캐스팅 방법은 지리적 위치와 비례적으로 성능이 저하되지만, 추가적인 모듈이 요구되지 않는 가장 유연한 메카니즘이라 할 수 있다. 앞으로는 설계된 Parallel-distributed 브로드캐스팅 P2P Federation을 구현하여 ebXML 레지스트리 간에 효과적인 Federation을 제공함으로써 B2B 전자상거래를 위한 새로운 솔루션으로 이용될 수 있도록 할 것이다.

참고 문헌

[1] UN/CEFACT, OASIS, "electronic business eXtensible Markup Language", http://www.ebxml.org
[2] OASIS/ebXML Registry Technical Committee, "OASIS/ebXML Registry Services Specification v2.5", http://www.oasis-open.org/committees/regrep/documents/2.5/specs/ebrs-2.5.pdf
[3] OASIS/ebXML Registry Technical Committee, "OASIS/ebXML Registry Information Model Specification v2.5", http://www.oasis-open.org/committees/regrep/documents/2.5/specs/ebrim-2.5.pdf
[4] M. Scholosser, M. Sintek, S. Decker, W. Nejdl, "A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Service", Second International Conference on Peer-to-Peer Computing(P2P'02), pp.104-111, 2002. 09
[5] S. Waterhouse, D. M.Doolin, G. Kan, Y. Faybishenko, "Distributed search in Peer-to-Peer networks", IEEE Internet Computing pp.2-6, 2002.02