

# DRRS<sup>1</sup>: ebXML 정보등록저장소를 위한 Federation and Replication 구현 사례

이영곤  
포스데이타 e-BI 개발팀  
ykleee@posdata.co.kr

장재경  
한국전자거래진흥원 표준개발팀  
jasmine@kiec.or.kr

## DRRS: An Implementation Case of Federation and Replication for ebXML Registry and Repository

Youngkon Lee  
e-BI Development Team  
POSDATA

Jasmine Jang  
Standardization Team  
KIEC

### 요 약

기업간 전자상거래 국제 표준인 ebXML의 표준 사양에 따른 정보등록저장소를 활용하여 각 산업별 B2B 표준 콘텐츠를 저장, 등록, 관리하는 사례가 증가하고 있다. 각 산업별 기관별 등록저장소가 만들어지고 또한 국가별로도 중앙등록저장소가 구현됨에 따라 이들을 서로 연계하여 산업간의 B2B 활용도를 높이고 또한 연동성을 향상시킬 수 있는 방안을 찾고 있는 중이다. 이미 OASIS에서는 Federation 및 Replication 기능을 추가한 ebRS 2.5 버전 사양에 대한 드래프트를 제시하였고 표준으로 승인받기 위한 절차를 진행 중에 있다. 본 논문에서는 한국전자거래진흥원의 지원하에 개발된 정보등록저장소의 Federation 및 Replication 처리시스템의 개발방법에 대해 살펴보고 현실에 적용할 때에 발생할 수 있는 제반 프로토콜상에서의 문제점 및 해결방안에 대해 논의해 보고자 한다.

### 1. 서론

기업간 협업을 위한 구현하기 위한 방법으로 국제 표준으로 제시된 ebXML의 표준 콘텐츠 저장을 위해 ebRS(ebXML Registry Service) 2.0[1]과 ebRIM(ebXML Registry Information Model) 2.0[2] 사양을 중심으로 정보등록저장소가 구축되어 왔다. 현재는 기구축된 정보등록저장소들간의 협력과 상호연동성을 기반으로 산업간, 기업간, 그리고 국가기관간에 서로의 콘텐츠를 공유하고 끊임없이 검색할 수 있는 시스템에 대한 요구가 커지고 있다. 이러한 요구사항을 바탕으로 OASIS 측에서는 Federation 기능과

Replication 기능이 추가된 ebRS 2.5 드래프트를 발표하였고 ebRS 3.0 표준으로 승인될 것이 유력한 상황에 있다. 본 논문에서는 그간 진행한 한국전자거래진흥원의 정보등록저장소인 REMKO와 한국철강협회의 철강산업 정보등록저장소인 KOSARR 간의 연동을 위해 ebRS 2.5를 기반으로 구축한 정보등록저장소의 Federation 기능의 구현사례에 대해 설명하고 구현과정상 노출된 문제점에 대해 논의해 보고자 한다. 특히, 정보등록저장소간 협력의 중심기능인 Federation과 Replication 간의 프로토콜 정의의 세부기능이 미진하기 때문에 발생할 수 있는 여러가지 이슈사항들에 대해 중점적으로 살펴보고자 한다. 이러한 문제점

<sup>1</sup> 한국전자거래진흥원(KIEC)의 지원하에 개발된 분산처리 및 Federation 기능이 가능한 정보등록저장소 시스템의 이름.  
DRRS: Distributed Registry and Repository System

들은 현 출시된 ebRS 2.5 가 드래프트이고 정보등록저장소의 사용형태가 한번 등록에 다수의 검색이라는 형태라는 점을 감안한다 하더라도 정보등록저장소간 데이터의 정합성과 자원요소 낭비라는 부가적이고 심각한 문제점을 초래할 수 있으며 향후 국가적 단위로 ebXML 중심의 B2B 표준 인프라 구축시 큰 쟁점이 될 수도 있을 것이다. 2 장에서는 ebRS 2.5 의 특징과 중점 기능, 3 장에서는 기반이 되는 아키텍처, 4 장에서는 ebRS 스펙의 문제점 및 개선사항에 대해 논의하고자 한다.

2. ebRS 2.5 특징

ebXML 등록저장소 2.5 에서는 각각의 등록저장소의 구조와 역할을 정의하는 것보다 여러 ebXML 등록저장소들간의 협력모델에 대한 내용이 중심을 이루고 있으며 이를 위해 다음과 같은 기능들을 지원한다 [3](그림 1 참조).

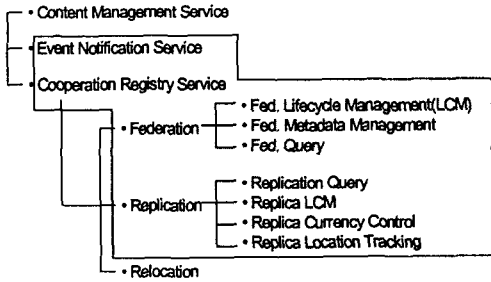


그림 1: ebRS 2.5 중점 추가기능

- Cooperating Registries: ebXML 등록저장소들이 여러 곳에 존재할 때 각 등록저장소간의 협업 모델을 구축하는 방안에 대한 내용을 제시. 이를 위해 Registry Federation, Object Relocation, Object Replication 에 대한 내용을 포함.
- Event Notification: 등록저장소에 이벤트가 발생하였을 때 사용자 또는 다른 등록저장소에게 이 사실을 알려주는 알려주는 메카니즘을 정의.
- Content-based Discovery: 단순히 ebXML 등록저장소에 저장된 메타데이터에 대한 검색하는 것이 아니라 실제 repository item 에 대해서 질의할 수 있도록 하는 기능.
- REST Interface to Registry: REST (Representational State Transfer)는 분산 하이퍼미디어 시스템을 위한 아키텍처로서 등록저장소에서 제공하는 서비스를 URL 을 통해 접근 가능하도록 함.
- Iterative Queries: Iterative 질의는 사용자의 질의에 대한 검색 결과가 매우 큰 경우 이 검색 결과를 iterate 하기 위한 방법이다. 즉, 하나의 질의를 여러 개로 나누어 처리함으로써 검색 결과는 같지만 각 질의어의 검색 결과 크기를 줄이기 위해 사용한다.

ebRS 2.5 의 기능을 요약하면 관련된 정보등록저장소의 연계를 표현하는 Federation 과 이 범위내에서의

정보 Replication, 그리고 Federation 이나 Replication 정보의 변경에 따른 데이터 Relocation 과 데이터 상황의 변경에 따른 이벤트 처리를 위한 event Notification 등으로 요약할 수 있다.

3. DRRS Architecture

분산처리 및 Federation 기능을 처리하기 위해서는 중앙등록저장소가 각 멤버 등록저장소들간의 연계를 위한 메타데이터를 가지고 있어야 하며, 각 등록저장소에 포함된 콘텐츠(CPP, CPA, BP, BPSS, 카탈로그)[4]의 정보연계표준 및 내용인증보안 모델을 제시해야만 한다. 또한 각 업종등록저장소에 저장된 콘텐츠를 연계하기 위해서 ebRS 에 나오는 Replication, Relocation, Event Notification 등을 처리할 수 있는 등록저장소간의 정보검색 기능 및 여러 등록저장소에 질의하고 결과를 통합할 수 있는 기능이 지원되어야 한다.

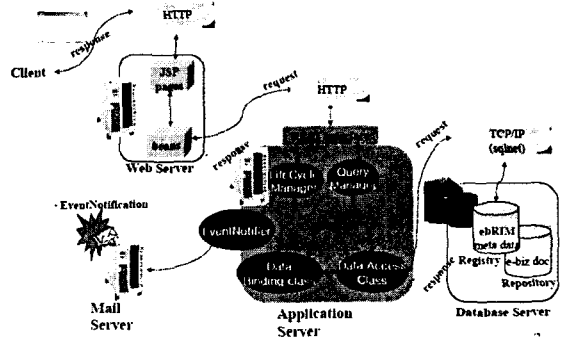


그림 2: DRRS Architecture

분산처리 기능의 구현과 데이터 관리의 편의성을 위해 우리는 Web Application Server(WAS)를 기반으로 한 3 Layer Architecture 로 시스템을 구성하였고, 특히 애플리케이션 서버의 경우 임의의 웹서버나 정보등록저장소에서 받을 수 있는 Federated Query 를 처리하기 위해 REST 인터페이스를 사용하였다. 또한 기존의 Life Cycle Manager 와 Query Manager 외에 Event Notifier 를 별도로 제작하여 데이터 오브젝트의 변경시 Federation 멤버간 데이터 상태공유를 쉽게 처리할 수 있도록 하였다[5].

• Federation home



• Federation member



그림 3: Federation metadata 구조

또한 하나의 Federation 을 관리하는 Federation Home 사이트와 Federation 멤버 사이트의 Federation 데이터를 정의하기 위해 그림 3 과 같이 Meta 데이터 형태로 정의하였다. 이를 별도의 관계를 이용해 구축하였다. 아래 그림 4 에서는 Federation metadata 를 REMKO 와 KOSARR 간 적용한 사례를 볼 수 있다.

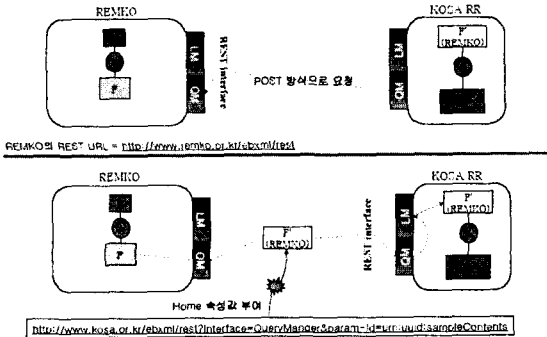


그림 4: Federation meta 데이터 적용사례

위에서 설명한 기능들을 중심으로 기개발된 REMKO 및 KOSA RR 에 그림 5 에서 보이는 기능들을 DRRS 에 추가하였다. 추가 기능 중 REMKO 와 KOSA RR 연계에 필수적인 Federation 관리 부분과 Replication Control 및 Event Notification 부분을 중점 개발하였고, Relocation 의 경우 Object Ownership 의 변경에 따른 Authorization, Authentication, Access Control 등의 정책이 정보등록저장소마다 워낙 상이한 관계로 개발에서는 배제하였다.

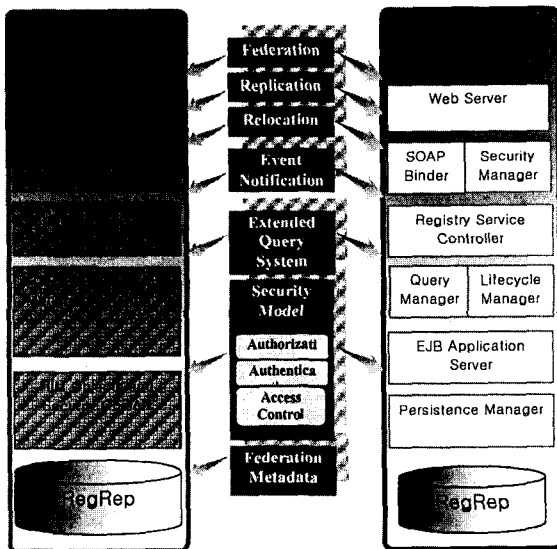


그림 5: DRRS 개발대상 영역

#### 4. ebRS 2.5 사양의 문제점 및 개선방향

Federation 은 여러 등록저장소를 하나의 그룹으로 묶는 것을 말하는데 이 경우 각 등록저장소를 물리적으로 결합하는 것이 아니라 분산된 각각의 등록저장소를 loosely-coupled 하게 구성해서 사용자들이 이 Federation 을 통해 여러 ebXML 등록저장소들을 하나의 커다란 논리적인 등록저장소로 다룰 수 있게 한다.

먼저 Federation 에 관련된 문제점부터 살펴보면, FHR (Federation Home Registry) Setup 에 대한 프로세스가 명확하지 않은 것이 가장 큰 문제라 할 수 있다. 초기 FHR 을 설정하는 과정에서 누구라도 FHR 이 될 가능성이 존재하기 때문에 한 산업이나 국가내에서 혼란이 야기될 가능성이 매우 크다. FHR 의 Ownership 을 누가 가질 것인가? 또한 어떤 Federation 에 가입할 경우 FHR(Fed. Home Registry)을 어떻게 알 수 있는가? 따라서 RR 간의 Collaboration 을 위해서 FHR List 를 관리할 수 있는 국가인증체계가 필요하다고 하겠다.

내 용	개 요	해결방안
어떤 Federation에 가입할 경우 FHR(Fed. Home Registry)을 어떻게 알 수 있는가?		FHR List를 국가인증기관에서 관리
FHR의 Ownership을 누가 가질 것인가? FHR의 난립 가능성		FHR Authority 에 대한 인증 필요

그림 6: FHR 선정과정의 문제점

하나의 federation 내 중앙등록저장소에 저장되는 정보는 업종등록저장소를 관리하기 위한 메타 정보뿐만 아니라 각 업종에서 공통으로 사용될 수 있는 데이터를 가지고 있어야 한다. 따라서 이 정보들은 특별한 업종에만 사용되는 것이 아니라 모든 업종에서 공통으로 사용될 수 있는 문서 및 분류체계를 포함해야만 한다.

그러나 Federation 및 Replication 에 관련된 문제점으로는 Replicated Data Object 의 일관성 유지, 트래킹 등에 대해 자세히 언급되지 않음으로 어려움을 들 수 있다. 이러한 경우에는 Federated Query 가 Replicated Object 에 주어질 경우 컴퓨팅 리소스의 낭비가 예상되며 어떤 Federation 탈퇴시, 관련된 Replica 정보들을 어떻게 관리 할 것인가 하는 것도 결정해야 할 문제라 할 수 있다. 이러한 경우에는 Replica 정보와 Registry Home Object 를 별도로 관리하고 탈퇴시 모든 Replicated 데이터를 퍼지아웃시킴으로써 해결책을 찾아볼 수 있을 것이다. 즉 Federated Query 가 Replica 에게 주어질 경우의

트랜잭션 비효율성을 방지하기 위해서는 Replicated Object에 대한 Federated Query가 방지되어야 한다.

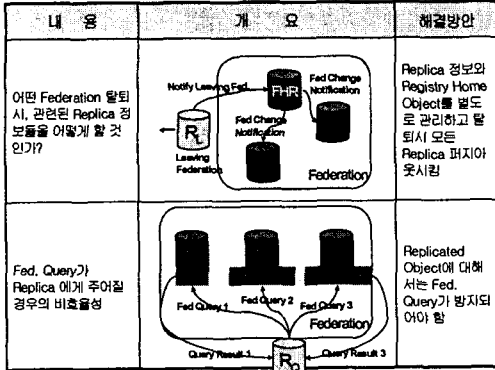


그림 7: Replication 데이터 변경에 따른 문제점

각 등록저장소는 각 업종에서 필요한 BP/CC, CPP/CPA 와 카탈로그 등에 대한 데이터를 저장한다. 각 멤버등록저장소는 자신의 정보뿐만 아니라 다른 등록저장소에 저장된 정보에 접근할 수 있다. 하지만 각 등록저장소는 분산되어 있기 때문에 네트워크나 기타 환경에 따라 접근이 불가능할 수 있다. 또한 다른 등록저장소의 정보를 자주 접근할 경우 접근 시간에 대한 오버헤드가 생기게 된다. 이러한 fault tolerance 와 접근 시간 단축을 위해서 각 등록저장소는 Local cache 정보를 저장할 수 있다.

Federated Query 와 관련된 또 다른 문제점은 Local Query 와 Federated Query 를 어떻게 구분해서 이슈 할 것인가라는 점이다. 바로 앞에서 얘기한 바와 같이 Local Query 와 Fed Query 를 구분하지 않을 경우 하나의 federation 에 가입한 RR 의 숫자가 많아질수록 리소스의 낭비는 물론 퍼포먼스의 엄청난 저하가 예상된다. 이러한 경우 Federation 내의 모든 Replication 정보가 트래킹 되어야 하고 Federated Query 를 발의하는 시점에 그 정보에 대한 활용이 가능해야만 할 것이다. 이러한 점들을 감안하여 다음과 같은 점들이 정보등록저장소 연계시스템을 구현할 때 고려되어야만 한다.

- FHR 뿐만 아니라 각 멤버등록저장소들은 서로의 관계에 따라 논리적인 Federation 을 구성할 수 있어야 한다. FHR 은 등록저장소들이 Federation 을 유지할 수 있도록 Federation 메타데이터 정보를 관리해야 하며 이 정보는 Federation, Registry, Association 정보를 포함한다.
- FHR 에서는 일반적으로 각 등록저장소에서 사용될 수 있는 공통 데이터들을 가지고 멤버등록저장소의 요구가 있을 때 이를 제공할 수 있어야 한다. 이러한 데이터는 BP, CC, CPP, CPA, BD, 카탈로그 등이 될 수 있으며 이 모든 데이터는 한 업종에 종속적인 것이 아니라 모든 업종에서 공통으로 사용될 수 있는 형태를 가져야 한다.

- 등록저장소에 저장된 데이터들은 각기 나름대로의 특성을 가지고 있기 때문에 이를 이용하여 효율적인 관리와 검색을 제공하기 위해서는 분류체계에 따라 체계적으로 정리할 필요가 있다. 따라서 중앙등록저장소에서는 표준 분류 코드를 정의하고 이에 따라 비즈니스 정보들을 분류할 수 있는 방법을 제시해야 한다.
- FHR 은 일반적으로 멤버등록저장소에게 제공할 정보를 가지게 되는데 이에 해당하는 것이 위에서 설명한 Federation 메타 데이터, 공통 데이터, 분류 코드 등이다. 하지만 이와는 반대로 멤버등록저장소의 정보를 replication 을 통해 중앙등록저장소에 유지하고 있어야 할 필요가 있다. 이 정보는 가장 작게는 모든 업종등록저장소에 저장된 모든 기관에 대한 Organization 정보가 되며 최대로는 각 멤버등록저장소의 모든 정보가 될 수 있다. 하지만 이 모든 정보를 유지하기에는 한계가 있기 때문에 일반적으로 비즈니스에서 많이 필요로 하는 정보가 되는 Business Process, Business Document, Collaboration Protocol Profile 정보를 유지함으로써 효율적인 비즈니스가 이루어질 수 있도록 한다.

5. 맺음말

정보등록저장소간 Federation 을 통한 정보연계가 원활하게 되기 위해서는 위에서 살펴 본 바와 같이 FHR 선정과정의 명확성과 Federation 변경에 따른 Registry Replication 데이터의 정합성을 좀 더 보장할 수 있는 방안들이 제시되어야 한다. 또한 event notification 기법의 보장을 통해 이벤트 발생에 따른 Replication 데이터 수정 및 데이터 정합성 보정 작업이 자동적으로 진행될 수 있는 메커니즘에 대한 연구가 좀 더 진행되어야 할 것으로 생각된다. 현재는 Federation metadata 의 보정을 자동으로 해 줄 수 있는 방법이 제시되어 있지 않으며 이 부분에 대한 구체적인 프로토콜도 제시되어야 할 것이다.

참고문헌

- [1] ebXML Registry Service, Version 2.0, Technical Committee Specification, <http://www.ebxml.org/specs/ebrs2.pdf>
- [2] ebXML Registry Information Model, Version 2.0, Technical Committee Specification, <http://www.ebxml.org/specs/ebrim2.pdf>
- [3] 이규철 외 6 인, “ ebXML 등록저장소간 협력표준정립 최종보고서, ” KIEC, 2002.12
- [4] <http://www.ebxml.org/specs/ebTA.pdf> “ ebXML Technical Architecture Specification v 1.0.4 ”
- [5] 이영근 외 3 인, “ ebXML 실적용 프로젝트 최종보고서, ” KIEC, 2003.12