



웹 서비스 구현을 위한 SonicXQ

백운기*

● 목 차 ●

- 1. 서론
- 2. SonicXQ의 구조와 기능
- 3. 결론

1. 서론

현재 업계의 동향을 바라볼 때, 오늘날 IT 트렌드 자체는 '통합(Integration)'이라는 개념을 중심으로 해서 흘러가고 있는 것처럼 보인다. 기업들은 그 어느 때보다도 기업 내부의 여러 조직들 사이에서 사용하고 있는 다양한 플랫폼들과 애플리케이션들을 서로 연결할 필요성을 느끼고 있다. 뿐만 아니라 기업과 기업 사이에서도 인터넷을 통해 서로 다른 플랫폼들과 애플리케이션들을 연결해서 비즈니스 데이터를 서로 주고 받는 가운데 협력적인 관계를 유지하는 일이 비즈니스 수행을 위한 필수 요건으로서 자리 잡아가고 있다. 이렇게 애플리케이션들을 서로 연결해서 필요한 데이터를 서로 주고 받아 비즈니스 프로세스를 원활하게 처리할 수 있도록 하는 일이 바로 여기서 말하는 통합의 의미이다.

또 한편으로, 지금 IT 업계에서 가장 중요한 이슈는 '웹 서비스'이다. 웹 서비스는 비즈니스 로직을 캡슐화한 독립적인 소프트웨어 모듈이다. 이런 모듈들이 인터넷 상에 분산 배치되며 일반적으로

여러 상이한 조직들이 소유하고 관리한다. 이들은 서로 함께 느슨하게 연결될 수 있어 지속적으로 쉽게 수정 될 수 있는 애플리케이션을 만들 수 있게 해준다. 이런 웹 서비스들은 대개 이기종 시스템들이기 때문에 이들은 파라미터 값들을 전달하고 받기 위한 공용 언어로서 XML을 이용하며 통신 프로토콜로서 HTTP를 이용한다. 이 밖에도 웹 서비스 개념을 지원하기 위해 많은 개방적 표준들이 채택되었다. 이런 것들에는 SOAP, WSDL, UDDI 등이 있다.

범용적인 산업 표준에 기반을 두고 있으며 플랫폼이나 프로그램 언어에 상관 없이 데이터를 교환할 수 있는 웹 서비스 자체의 특성 상 기업의 애플리케이션들을 통합하기 위한 기술적 기반으로서 웹 서비스가 자리매김을 하고 있는 현실이다. 여기서 우리가 주목해야 할 부분은 웹 서비스가 여러 시스템들 사이의 확실한 통신 및 통합을 위해 신뢰성 있고 안전한 메시징 인프라스트럭처를 반드시 필요로 한다는 점이다.

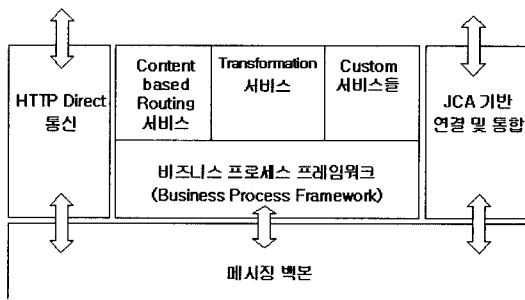
본고는 신뢰성 있고 안전한 메시징 백본을 기초로 해서 서비스 지향적인 아키텍처 안에서 웹 서비스를 구축할 수 있도록 해주는 웹 서비스 플랫폼인 SonicXQ 제품에 대해 기술적, 구조적 측면에서 살

* 엔시스(구 한국인프라이즈) 인터넷 미들웨어 기술팀장

펴보고자 한다. 참고로 SonicXQ는 국내에도 잘 알려진 미국의 Progress 사의 자회사인 Sonic Software 사가 개발 및 제공하는 제품이다.

2. SonicXQ의 구조와 기능

SonicXQ는 웹 서비스와 같은 분산 서비스 지향적 아키텍처의 구현 그리고 이기종 시스템들의 통합을 위한 백본이다. SonicXQ의 기능적인 구조를 그림으로 표현하면 다음과 같다. 그림을 토대로 각 구성 요소들에 대해서 알아보도록 하자.



(그림 1) SonicXQ의 구조

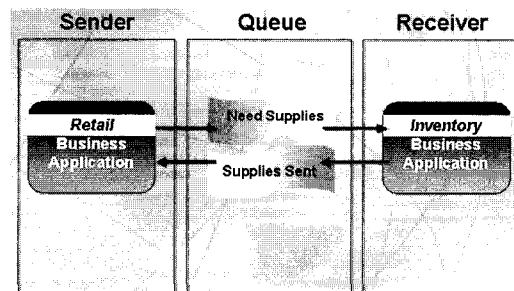
2.1 메시징 백본

메시징 백본이란 쉽게 말해서 시스템들 사이의 메시지 송신과 수신을 전달하는 미들웨어이다. 웹 서비스 환경에서는 이질적인 각각의 프로세스들 사이의 신뢰성 있고 안전한 통신을 가능하게 해 줄 수 있는 기술이 필요하다. 즉 오늘날의 분산 환경 안에서는 데이터 포맷, 플랫폼, 프로토콜, 사용 프로그래밍 언어 등이 다양하기 때문에, 애플리케이션 소스 코드에 수정을 가하지 않고서도 애플리케이션 통합을 가능하도록 하기 위해 표준 기술들을 많이 이용하는 메시징 미들웨어를 채택할 필요가 있다.

Java Message Service(JMS)는 이기종 애플리케이션들 사이의 통신 메커니즘을 정의한 표준으로서 J2EE 표준의 일부이다. SonicXQ의 메시징 백본은

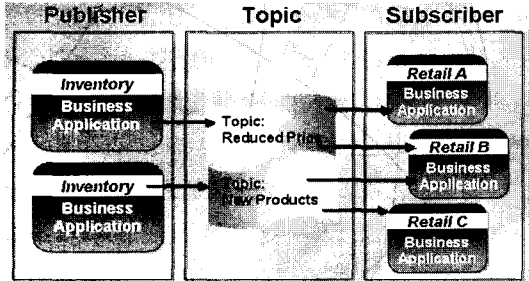
JMS 표준을 구현하고 있으며 인터넷 상에 분산되는 서비스들 사이의 통신을 위한 인프라스트럭처를 제공한다. 보다 구체적으로 말해서 SonicXQ의 메시징 백본은 JMS 표준이 요구하는 대로 Pub/Sub 메시징 모델과 Point-to-Point 메시징 모델을 모두 지원하며 절대로 메시지를 손실하는 일 없이 확실한 메시지 전달(통신)을 보장하는 뛰어난 장치들을 가지고 있다.

Point-to-point 메시징 모델이란 메시지 큐(Queue)를 매개체로 해서 메시지를 송신하고 수신하는 구조이며 일대일 관계의 통신에 적합한 모델이다. 수신자가 메시지 큐로부터 메시지를 수신하면 메시지 큐의 메시지가 삭제되므로 오직 한 수신자만 메시지를 수신할 수 있다.



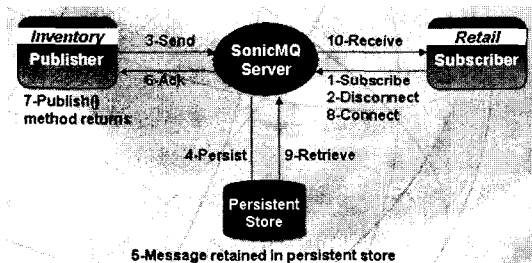
(그림 2) Point-to-point 메시징 모델

Pub/Sub 메시징 모델이란 토픽(Topic)을 매개체로 해서 메시지를 송신하고 수신하는 구조이며 일대다 관계의 통신에 적합한 모델이다. 우선 메시지를 수신하고자 하는 수신자는 자신이 관심을 가지고 있는 토픽에 대해 구독 신청(Subscribe)을 해야 한다. 그래서 이 모델에서는 메시지 수신자를 구독자(Subscriber)라고 부른다. 이와 반대로 메시지 송신자는 토픽에 대해 메시지를 발행한다. 그래서 메시지 송신자를 발행자(Publisher)라고 부른다. 이 모델에서는 토픽에 대한 구독자의 수에 상관 없이 모든 구독자들이 토픽으로부터 발행자의 메시지를 수신할 수 있다.



(그림 3) Pub/Sub 메시징 모델

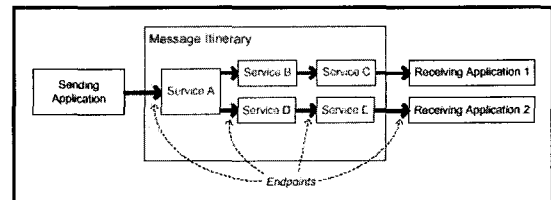
이번에는 어떤 방식으로 메시지 전달이 보장 되는지 다음(그림 4)와 함께 확인해 보자. 우선 메시지 구독자가 메시징 백본에 'durable subscription'으로 연결한다. 그리고 어떤 이유에서든 그 연결이 끊어졌다고 가정하자. 다음으로 메시지 발행자 혹은 생성자는 토픽에 대해 메시지를 persistent 모드로 발행한다. 개발자는 메시지를 토픽에 대해 발행할 때 혹은 메시지를 큐에게 송신할 때 persistent 모드를 지정할 수 있다. 이것은 메시징 백본으로 하여금 만일의 경우에 대비해서 메시지의 복사본을 별도로 저장해 두도록 지시하는 것이다. 이 경우 SonicXQ 메시징 백본에서는 데이터베이스에 메시지의 복사본을 별도로 저장한다. 다음으로 메시징 백본이 메시지를 제대로 수령했음을 알리는 신호가 메시지 발행자에게 보내진다. 그런 다음 어떤 때이든 메시지의 구독자가 다시 연결을 할 때 메시지가 다시 전달된다.



(그림 4) Persistent 메시징

2.2 비즈니스 프로세스 프레임워크

비즈니스 프로세스 프레임워크(Business Process Framework)는 '서비스'들을 이용해서 메시지를 처리한다. 좀 더 알아듣기 쉽게 얘기하자면, 메시징 백본을 통해 전달 된 일반 JMS 메시지 혹은 SOAP 메시지는 '메시지 일정표(Message Itinerary)'라는 것을 이용해 비즈니스 프로세스 프레임워크 안에 정의 된 여러 '서비스'들을 거치게 된다. 메시지 일정표 안에는 해당 메시지가 어떤 서비스들을 어떤 순서로 거쳐야 하는지에 대한 정보가 기록되어 있다. 그리고 서비스들은 독립적인 모듈의 소프트웨어이며(인터넷을 포함한) 네트워크 상에 분산되어 있고 입력 값으로 메시지를 받은 후 이를 처리해서 다른 메시지를 만들어 내거나 EJB와 같은 외부 서비스를 호출한다. (그림 5)는 어떻게 개별적인 서비스들이 하나의 비즈니스 프로세스를 이루기 위해 맞춰지는지 보여준다.



(그림 5) 비즈니스 프로세스

메시지는 '종단점(Endpoint)'이라는 단일 entry point를 통해서 비즈니스 프로세스에 들어온다. 종단점은 JMS 토픽, 큐 혹은 JCA 기반의 커넥터와 같은 기타 다른 시스템들로부터 메시지들을 수신할 수 있다. 비즈니스 프로세스 내부에서 이 메시지는 일련의 서비스들을 통과하면서 이동한다. 즉 메시지가 또 다른 종단점에서 비즈니스 프로세스를 빠져 나와 목적지로 이동하기 전에 보다 많은 종단점들을 통해 이런 서비스들에 들어가고 빠져 나온다.

SonicXQ는 비즈니스 프로세스를 정의하는 일을

보조하기 위해 Content-based Routing 서비스와 Transformation 서비스 이렇게 두 가지의 미리 정의된 서비스들을 제공한다. 이 밖에도 개발자가 나름대로 Custom 서비스를 정의할 수도 있다.

Content based Routing 서비스는 메시지의 내용(body, content)에 기초해서 메시지 라우팅을 결정한다. 예를 들어서, 어떤 주문 정보를 담은 XML 기반의 메시지가 비즈니스 프로세스 안에 들어오면, Routing 서비스가 이 메시지의 내용을 조사하고 내부적으로 정해진 규칙에 따라 이 주문을 제대로 처리할 수 있는 다른 적절한 목적지로 전달해 준다.

Transformation 서비스는 사용자가 메시지 내용을 어떤 한 형식에서 다른 형식으로 변환할 수 있게 해주는 XML style sheet(XSLT)를 이용한다. 예를 들어서, 비즈니스 프로세스에 들어온 주문 정보가 SAP ERP 시스템에 맞는 포맷의 메시지이고 라우팅 서비스에 의해 Oracle ERP 시스템의 소프트웨어 쪽으로 전달 되어야만 하는 상황이라고 가정해보자. 이 경우 Oracle ERP 시스템에 맞는 포맷의 메시지로 자동 변환을 해주는 서비스가 Transformation 서비스이다. 이 서비스는, 송신자가 미리 수신자에게 적절한 형태의 메시지를 준비한 후 보내도록 작업해야만 하는 짐을 없애준다. 이는 송신 애플리케이션 쪽의 수정 없이도 여러 서비스들의 쉬운 통합을 가능하게 해준다는 장점을 제공한다.

2.3 HTTP Direct 통신

SonicXQ는 HTTP Direct 통신 기능을 이용해서 HTTP 클라이언트들이 HTTP 혹은 HTTPS 통신 프로토콜을 통해 하부 메시징 백본과 직접 inbound 및 outbound로 통신할 수 있게 해준다. 이는 클라이언트 컴퓨터 쪽에 메시지 처리를 위한 별도의 소프트웨어 모듈이 설치될 필요가 없음을 의미한다. 따라서 외부 애플리케이션이나 서비스들이 적은 비용과 플랫폼 독립적인 방식으로 쉽게 비즈니스 프로세스에 통합될 수 있다.

아울러 SonicXQ는 두 가지 프로토콜 처리기를 제공함으로써 이 기능을 확장한다:

HTTP for SOAP Protocol Handler: 들어오는 SOAP 메시지들(attachment 있거나 혹은 없거나)에 대해 유효성 검사를 하고 이들을 JMS 메시지로 자동 변환한다. 그러면 이들이 JMS 목적지(토픽 혹은 큐)에 직접 전달될 수 있다.

HTTP for JMS Protocol Handler: 모든 JMS 액션들과 속성들을 HTTP 헤더에 끼워 넣어서 들어오는 메시지들이 JMS 토픽 혹은 큐 상에 위치할 수 있게 된다.

2.4 JCA 연결 및 통합

기업은 외부 시스템들과의 통합에 대한 필요성을 가지고 있다. 이는 특별히 웹 서비스 기반의 인프라스트럭처로 이동할 때 그렇다. 일반적으로 이런 상황에서 하부 애플리케이션들의 복잡성 그리고 통합 작업을 수행하기 위해서 필요한 다양한 기술들이 가장 큰 문제점이다. SonicXQ 는 J2EE Connector Architecture(JCA)를 이용해서 이런 애플리케이션 통합을 용이하게 해준다. JCA는 이기종 시스템들의 통합을 표준화 하기 위해 고안된 새롭게 떠오르는 기술이다. 이것은 애플리케이션 서버들 및 Enterprise Resource Planning(ERP) 애플리케이션들과 같은 다양한 메시징 도메인들 및 애플리케이션들에 대한 커넥터를 개발하기 위한 표준을 제공한다. 아울러 SonicXQ는 JCA Connector 툴킷을 제공해서 개발자가, 애플리케이션 서버들, 레거시 시스템들 및 기타 메시징 인프라스트럭처들과 같은 J2EE 호환 애플리케이션들에 대한 커스텀 어댑터를 개발할 수 있도록 해준다.

3. 결론

지금까지 웹 서비스 및 애플리케이션 통합 플랫폼으로서의 SonicXQ의 구조 및 기능적 측면을 살

퍼 보았다. 이제 웹 서비스를 위해 필요한 기술 요소들의 관점에서 SonicXQ가 어떻게 웹 서비스를 지원하고 있는지 간략하게 정리해 보겠다.

웹 서비스는 시스템들 사이의 통신을 보장하기 위해서 신뢰성 있고 안전한 메시징 인프라스트럭처를 필요로 한다. SonicXQ는 이런 메시징 기능들을 제공하며 웹 서비스에 있어 공통적으로 이용되는 표준(HTTP, XML, SOAP, WSDL, UDDI)들을 함께 지원한다.

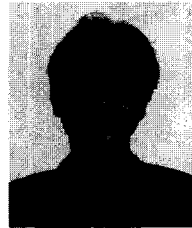
SonicXQ를 이용해서 서비스들의 집합인 비즈니스 프로세스를 정의하고 나면 이에 대한 WSDL (Web Service Definition Language)을 GUI 환경 안에서 쉽게 생성할 수 있다. 이 때 만들어진 WSDL은 외부 UDDI 디렉토리에 등록되어 비즈니스 프로세스가 웹 서비스로서 작동할 수 있게 된다.

웹 서비스와 웹 서비스 사이 혹은 웹 서비스와 웹 서비스의 호출자 사이에서는 데이터 교환을 위해 SOAP 표준이 이용된다. SonicXQ는 HTTP Direct 통신 기능을 통해 자신에게 들어오는 SOAP 메시지를 메시징 백본에서의 처리를 용이하게 하기 위해 유효성 검사 후 JMS 메시지로 직접 변환할 수 있다. 그리고 SonicXQ는 SOAP Document 모델(비동기식)과 SOAP RPC(동기식) 모델을 모두 지원한다.

지금까지 소개한 SonicXQ가 추구하는 '서비스 지향적인 아키텍처'는 쉽고 비용이 적게 드는 방법으로 한 조직의 시스템을 외부 교역 파트너들의 시스템들과 통합할 수 있게 해준다. SonicXQ는 인터넷 등 분산된 네트워크 상에 위치하고 있는 서비스들을 통합하기 위한 최선의 플랫폼이다. 이 제품이 지니고 있는 표준 기반의 통합성은 기존 애플리케이션들에 대한 투자를 포기하지 않고서도 효율적으로 발전시켜 나갈 수 있도록 해준다. 기업 외부 애플리케이션에 대한 연결 능력은 교역 파트너들과의 상호 협력을 지속화 시키고 기업의 글로벌 규모를 확장시킬 수 있게 해준다. SonicXQ의 신뢰성 있는 메시징 백본은 비즈니스에 있어 중요한 데이

터를 절대로 손실되지 않게 보장해준다.

저자 약력



백운기

엔시즈(구 한국인프라이즈) 인터넷 미들웨어 기술팀장
엔시즈 직영 교육 센터 강사