

Wf-XML 2.0 기반 상호운용성 지원을 위한 프로세스 엔진 설계

안형진⁰ 박민재 김광훈
경기대학교 전자계산학과
{ctrl_workflow⁰, mean222, kwang}@kyonggi.ac.kr

Design of Process Engine for Interoperability based Wf-XML 2.0

Hyungjin Ahn⁰ Meanjae Park Kwanghoon Kim
Dept. Computer Science, Kyonggi University

요 약

현재의 급변하고 있는 비즈니스 환경은 기업 내부의 인트라적인 업무 처리 뿐만 아니라, 기업과 기업간의 상호교환을 통한 업무 수행을 요구하고 있으며, 이러한 요구 사항에 적절하게 대응할 수 있는 전략 개념이 프로세스 상호운용성이다. 본 논문에서는 현재의 비즈니스 시장에서 이슈화되고 있는 프로세스 상호운용성을 실제적으로 구현하고자 하는 과정의 일환으로 워크플로우 표준 단체인 WfMC(Workflow Management Coalition)에서 제안하는 상호운용성 표준 메시지 프로토콜인 Wf-XML 2.0을 지원하는 프로세스 엔진의 시스템 아키텍처를 구성하고자 하며, 기반이 되는 프로세스 엔진은 자체 연구 제작된 e-chautauqua라는 워크플로우 시스템을 이용하여 프로세스 상호운용성이 가능할 수 있도록 확장 설계하고자 한다.

1. 서 론

현재의 비즈니스 환경은 단순한 기업 내부의 프로세스를 통한 업무 자동화를 지원하는 인트라 개념의 프로세스 관리 시스템 뿐만 아니라, 서로 다른 기업 및 조직간에 보유한 프로세스 엔진들 간에 서비스를 상호 교환하는 인터-워크플로우 & BPM 시스템의 개념으로 확대되어 비즈니스의 영역은 날로 거대화되어가고 있다. 각 개개의 기업 및 기관들이 사용하는 프로세스 엔진의 종류가 서로 다른 이기종 시스템이라 할 때, 이러한 이기종 프로세스 엔진들 간에 상호운용이 가능하게 하기 위해서 가장 적절하게 사용될 수 있는 방법이 데이터의 표준으로서 각광을 받고 있는 XML을 이용한 바인딩이다. XML은 현재 이슈가 되고 있는 웹 서비스의 근간이 되는 데이터 표준으로서 서로가 보유하고 있는 시스템의 환경이 다르더라도 XML을 이용하게 되면 상호 간의 통신이 가능해지게 된다. 비즈니스 프로세스 상호운용을 통한 서비스의 특성 상 업무 요청에 대해서 긴 시간의 응답 시간을 필요로 하는 업무들이 많기 때문에, 동기적인 업무 처리 방식 뿐만 아니라 비동기적인 처리 방식의 서비스 제공 또한 필요하다. 프로세스 상호운용 지원 환경에서의 비동기 서비스를 가능하게 하기 위하여 워크플로우 표준 단체인 WfMC(Workflow Management Coalition)에서는 OASIS 표준 단체에서 제안하는 비동기 웹 서비스 프로토콜인 ASAP(Asynchronous Service Access Protocol)을 기반으로 한 Wf-XML을 상호운용성 구현의 표준으로서 제시하고 있으며, 현재 2.0 버전이 배포된 상태이다.

본 논문에서는 Wf-XML 2.0을 이용하여 프로세스 엔진 간의 상호운용성을 구현하기 위한 기반이 되는 아키텍처를 설계하고자 한다. 2장은 관련 연구로서 프로세스 엔진 간의 상호운용성이 무엇인지에 대해 간략히 살펴보고, 비즈니스 프로세스 상호운용성의 개념들을 토대로 제작된 표준 메시지 프로토콜인 Wf-XML 2.0에 대해 알아본다. 3장에서는 자체 제작된 e-chautauqua라는 워크플로우 관리 시스템에 Wf-XML 2.0을 이용한 상호운용성에 대한 지원 가능 시스템 아키텍처를 구성한다. 4장에서 앞선 내용들을 토대로 향후 시스템의 구현 및 발전 방향들에 대해서 기술하면서 논문을 마치고자 한다.

2. 관련 연구

2.1. 프로세스 엔진 간 상호운용성

프로세스 엔진 간 상호운용성은 둘 이상의 프로세스 엔진들 간의 서비스 교환을 통해 이루어지며, 각 프로세스 엔진들은 동종의 시스템일 수도 있고, 이기종의 시스템일 수도 있다. 또한 중점 시스템의 형태로서, 여러 프로세스 엔진이 그룹화되어 인엑트먼트 서비스를 제공할 수도 있다. 동종의 인엑트먼트 서비스를 지원하는 프로세스 엔진 환경일 경우에는 상호간의 통신에 아무 하자가 생기지 않는다. 직접적으로 메소드 호출을 통해 실제적인 인스턴스 데이터들이 네트워크를 통해 교환되어 서비스 요청에 대한 응답을 상호 간에 가능하게 할 수 있다. 다음의 그림 1은 동종의 인엑트먼트 서비스를 제공하는 중점 프로세스 엔진의 상호운용을 나타내고 있다.

Wf-XML 2.0은 XML과 SOAP을 기반으로 구성됨으로서

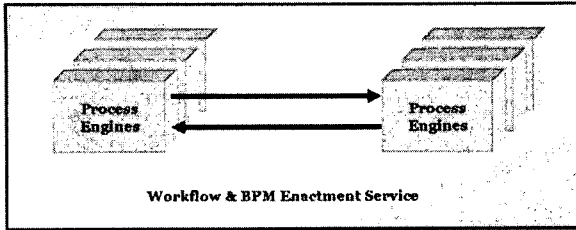


그림 1. 동종의 프로세스 엔진들 간의 상호운용

그러나, 비즈니스 프로세스 환경에서 각 개개의 기업들이 모두 동일한 프로세스 엔진을 사용할 리는 만무하다. 여러 벤더들이 제공하는 프로세스 엔진이 존재하고 사용하는 회사들은 다양한 프로세스 엔진들을 사용하게 되며, 한 기업 내에서도 사용되는 프로세스 엔진들이 각각 다른 기종의 엔진일 수도 있다. 아래의 그림 2는 이기종 간의 인엑트먼트 서비스를 수행하는 중점 프로세스 엔진의 상호운용을 나타낸다.

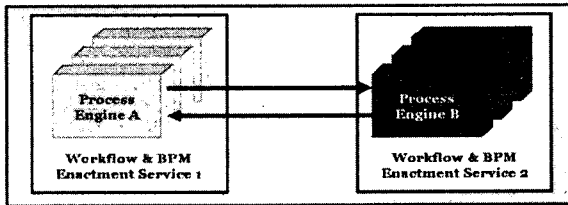


그림 2. 이기종 프로세스 엔진들 간의 상호운용

이기종의 프로세스 엔진들 간의 상호운용일 경우에는 동종의 시스템들에서 이루어지는 상호운용의 통신과는 다른 메커니즘으로 수행된다. 서로 다른 시스템들이 의사소통이 가능하기 위해서는 각각의 시스템들 사이에 교량 역할을 할 수 있는 매개체가 필요하며 이러한 상황을 만족시킬 수 있는 환경이 XML 기반의 웹 서비스 환경이다.

2.2. Wf-XML 2.0

Wf-XML은 워크플로우 표준 단체인 WfMC에서 OASIS의 ASAP(Asynchronous Service Access Protocol) 비동기 웹 서비스 프로토콜을 기반으로 비즈니스 환경에서 일어나게 되는 상호운용성을 위한 구현에 적합하도록 제안된 프로세스 엔진 간 상호운용성에 대한 표준이다. Wf-XML은 ASAP가 가지는 웹 서비스를 통한 비동기 서비스 구현의 모습을 그대로 가지고 있기 때문에, 비즈니스 기업 조직들 간에 서로 다른 프로세스 엔진을 가지고 있다 하더라도 ASAP를 통한 웹 서비스 메세징 기술을 통하여 상호 통신을 통한 서비스 요청 및 긴 시간의 트랜잭션을 요하는 비즈니스 업무 처리에 적합하게 사용할 수 있는 프로토콜이다. 이기종 프로세스 엔진들 간의 상호운용성을 위해 WfMC에서 제안하는 상호운용성 표준인 Wf-XML은 현재 2.0버전이 제작되어 있는 상태이다.

확장성 및 유연성이 뛰어나며, 특정 플랫폼을 구분하지 않기 때문에 임의의 플랫폼에 위치한 프로세스 엔진들 간의 상호운용성에 대한 처리가 가능하고, 비동기 웹 서비스를 지원하기 때문에 장시간의 트랜잭션이 필요한 비즈니스 서비스 수행에 적합한 프로토콜이라고 할 수 있다. 또한 ASAP가 SOAP을 토대로 만들어진 프로토콜이기 때문에, Wf-XML 2.0 또한 기본적인 메시지 구성은 SOAP의 모습과 많이 다르지 않으며, 이러한 점은 개발자에게 비즈니스 프로세스 환경에서의 상호운용성에 대한 구현을 용이하게 해주는 장점을 가지고 있다.

3. 상호운용성 지원을 위한 e-chautauqua 워크플로우 시스템 설계

3.1. e-chautauqua 워크플로우 시스템

본 논문에서 사용하고 있는 워크플로우 관리 시스템은 e-chautauqua라는 이름을 가진 초대형 워크플로우 시스템이다. 초대형 워크플로우라는 것은 워크플로우 시스템을 사용하는 조직이 점차 거대화되어 조직에서 사용되는 워크플로우 프로세스의 수가 많아지고, 그에 따른 작업의 수가 대량으로 일어남에 따라 발생하는 대량의 워크플로우 인스턴스를 처리하는 것에 목적을 둔 워크플로우를 말하며, 이러한 초대형 워크플로우 개념을 기반으로 e-chautauqua 워크플로우 시스템은 제작되었다. 아래의 그림 3은 e-chautauqua 워크플로우 시스템의 아키텍처 구성도를 나타내고 있다.

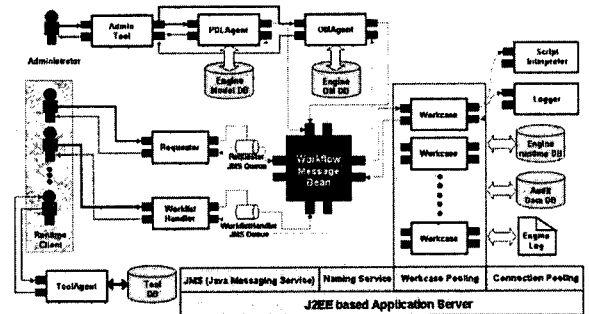


그림 3. e-chautauqua 워크플로우 시스템 아키텍처

e-chautauqua 워크플로우 엔진은 J2EE의 EJB 컴포넌트 구조를 기반으로 개발되었다. 엔진을 구성하는 핵심 컴포넌트들은 Requester, Worklist Handler, Workcase 이다. Requester는 모니터링 또는 런타임 클라이언트로부터 작업 요청을 받아들여 워크케이스를 생성하고 시작하여 워크케이스 및 액티비티 인스턴스등과 관련된 서비스에 대한 결과를 제공해주는 역할을 하는 컴포넌트이다. Worklist Handler는 비즈니스 업무를 수행할 사용자에게 할당되는 단위 업무인 워크아이템과 관련된 서비스를 처

리해주는 컴포넌트이며, Workcase는 프로세스 인스턴스인 워크케이스와 대응되는 컴포넌트로서 실질적인 프로세스 업무 처리의 주체가 되는 컴포넌트이다. e-chautauqua 워크플로우 엔진은 외부로부터의 서비스 요청에 대해서 핵심 구성 컴포넌트들의 상호작용을 통해 그에 대한 서비스 처리를 하게 된다.

3.2. Wf-XML 2.0 기반 e-chautauqua 워크플로우 시스템 아키텍처의 확장 설계

본 논문에서는 기존에 제작된 e-chautauqua 워크플로우 시스템에 이기종의 프로세스 엔진 간의 상호운용성을 지원하는 시스템의 확장된 아키텍처를 설계하고자 한다. 엔진들 간의 상호운용성 관련 오퍼레이션 수행은 엔진을 보유하고 있는 벤더들이 제공하는 서비스 목록을 공개하는 WSDL 페이지 또는 JSP / Servlet 웹 페이지를 통해 시작된다. 이러한 서비스 목록 제공 페이지를 통해 다른 엔진을 통해 얻고자 하는 서비스를 요청하게 되며, 이러한 요청 메시지는 Wf-XML 2.0 메시지 표준을 통해 구성되어진다. Wf-XML 2.0은 SOAP을 토대로 구성된 메시지 프로토콜이기 때문에 프로세스 엔진의 앞단에는 SOAP 엔진이 위치하게 되며, SOAP 엔진을 통해 Wf-XML 메시지가 도착되면 JAX-RPC와 같은 SOAP 파서를 통해 Wf-XML 메시지를 파싱하게 된다. 이 때, 메시지에 대한 파싱에 대한 룰을 제정하고 그에 대해 프로그래밍하는 것은 표준의 권한 밖이며, 전적으로 엔진 개발자가 책임을 지게 된다. 파싱된 메시지의 내용을 해석하여 요청된 작업이 동기적 또는 비동기적으로 처리되어야 하는지를 우선적으로 파악한다. 동기적 처리냐 비동기적 처리냐에 대한 판별은 Wf-XML의 헤더 메시지의 접두어를 통해 파악이 가능하다. 비동기적으로 처리되어야 하는 비즈니스 업무는 Wf-XML 헤더에 "aws(asynchronous web service)"라는 접두어를 가진다. 메시지 헤더를 통해 비동기 여부를 판단하여 현재 요청된 오퍼레이션이 비동기적인 업무 수행을 요하는 오퍼레이션이라면, 해당 오퍼레이션의 메소드 및 파라미터들은 메시지-드러본 빈에 전달되어 메시지 빈과 연동된 JMS(Java Message Service)의 큐에 적재된다. 적재된 메시지는 해당 오퍼레이션을 처리하는데 연동되는 엔진 내 컴포넌트와 연계되어 실질적인 작업 처리를 하게 되며, 처리 완료된 데이터들은 다시 역으로 Wf-XML 형태의 메시지로 구성되어 요청 측의 엔진으로 전송하게 된다. 다음의 그림 4는 e-chautauqua 워크플로우 엔진에 상호운용성을 지원 가능하도록 확장한 아키텍처 구성도를 나타내며, 이 그림에서의 엔진의 모습은 앞서 보았던 그림 3의 구성과 동일하다.

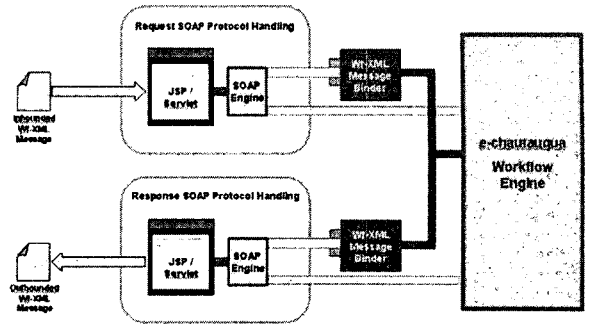


그림 4. 상호운용성을 고려한 e-chautauqua 워크플로우 관리 시스템의 확장 아키텍처

4. 결론 및 향후 발전 방향

지금까지 워크플로우 표준 단체인 WfMC에서 제안하는 프로세스 상호운용성 표준인 Wf-XML 2.0 표준 메시지 프로토콜과 자체 제작한 워크플로우 관리 시스템과의 연동을 통해 프로세스 엔진 간 상호운용성을 지원하는 시스템 아키텍처를 구성해 보았다. 향후에는 본 논문에서 제안한 상호운용성 아키텍처를 이용하여 실제적인 구현을 하고, 구현된 시스템과 이기종의 프로세스 엔진과의 실질적인 상호운용성 실험을 하고자 한다.

현재의 급변하고 거대화되어가는 비즈니스 프로세스 환경은 기업 내부에서의 업무 처리 뿐만 아니라, 더 나아가 기업과 기업간의 상호교환을 통한 업무 수행을 요구하게 되었고 이러한 요구 사항에 적절하게 대응할 수 있는 전략으로서의 개념이 바로 프로세스 상호운용성이다. 이러한 비즈니스 프로세스 상호운용성이 지원되는 프로세스 엔진을 통해 기업의 자동화 및 정보화의 상호 교류 발전에 큰 기여를 할 수 있게 될 것이다.

Acknowledgement

본 연구는 정보통신연구진흥원 정보통신 기초기술 연구지원사업(04-기초-0005)의 지원으로 수행되었음.

5. 참고문헌

- [1] "Simple Workflow Access Protocol (SWAP)", Keith D Swenson
- [2] "Wf-XML 2.0 - XML Based Protocol for Run-Time Integration of Process Engines", Keith D Swenson, Sameer Predhan
- [3] "ASAP / Wf-XML 2.0 Cookbook", Keith D Swenson, Fujitsu Software Corporation
- [4] "Workflow Standard - Interoperability Internet e-mail MIME Binding", The Workflow Management Coalition