

Wf-XML HTTP 바인딩을 이용한 WfMS 간의 연동

곽명재, 심재용, 한동수
한국정보통신대학원대학교 공학부
e-mail : {mjkwak, jaeyong7, dshan}@icu.ac.kr

Interoperation among WfMSs using Wf-XML HTTP binding

Myung-Jae Kwak, Jae-Yong Shim, Dong-Soo Han
School of Engineering, Information and Communications University

요 약

최근 B2B 시스템 통합 및 전략적 제휴가 시장을 변화시키고 있고 관련 기업들의 효율적인 연계요구가 증가됨에 따라 기업간 시스템들의 연동이 중요한 문제로 부각되고 있다. 워크플로우 표준화 단체인 WfMC에서도 이러한 문제 해결의 일환으로 인터넷을 통한 워크플로우 관리 시스템간 연동을 위한 표준인 Wf-XML Interoperability standard 를 발표하였다. 본 논문은 Wf-XML 표준 및 관련 연구들을 조사하고 WfMC 표준에는 정의되어 있지 않으나 시스템간의 신뢰성있는 연동을 위해 고려할 사항 중에서 동기적 및 비동기적 상호연동을 위한 요구사항들을 분석한다. TCP/IP 를 기반으로 한 HTTP 프로토콜을 전송 프로토콜로 채택하는 경우 여러 오퍼레이션들을 일괄처리하기 위해서는 비동기적인 상호연동(asynchronous interoperation)을 위한 별도의 메커니즘이 필요하다. 본 연구에서는 Wf-XML 문서를 HTTP 프로토콜을 사용하여 전송하는 경우 일괄처리요청(batch request) 및 그 밖의 동기적 또는 비동기적 요청들을 처리하기 위한 상호연동 모델을 분석하고 분산 워크플로우 시스템의 일종인 ICU/COWS에서의 적용방안에 관해서 논의한다.

1. 서론

많은 기업들이 기업 내부에 존재하는 정보의 흐름을 효과적으로 제어하기 위하여 워크플로우 관리 시스템(WfMS)을 도입하고 있다. 특히 최근에는 B2B 시스템 통합 및 전략적 제휴가 상거래 형태를 변화시키고 있고, 이에 따라 관련 기업들의 효율성이 증가됨으로써 기업간 시스템들의 연동이 중요한 문제로 부각되고 있다[6]. 이러한 문제 해결의 일환으로 워크플로우 표준화 단체인 WfMC(Workflow Management Coalition)에서도 지난 5월 인터넷에서의 워크플로우 관리시스템간의 연동을 위한 Wf-XML(Workflow eXtensible Markup Language)표준을 발표하였다[1].

종전에는 e-Mail 등을 이용한 비동기적인 연동만이 가능했으나, HTTP 프로토콜과의 바인딩으로 동기적 연동이 가능한 Wf-XML 표준은 WfMC에서 IETF(Inter Engineering Task Force)의 SWAP(Simple Workflow Access Protocol) 프로토콜의 장점을 흡수하여 제안되었다 [1][6].

Wf-XML 표준은 HTTP, SMTP 및 IIOP 등의 전송

프로토콜에 관계없이 사용할 수 있으나, 대부분의 워크플로우 시스템이 원거리에 있는 다른 시스템과 연동할 경우 인터넷을 사용할 것이므로 Wf-XML HTTP 바인딩을 이용할 것으로 판단된다. 즉 e-mail의 경우는 비동기적인 연동만이 가능하며, IIOP(Internet Inter-ORB Protocol)는 ORB 간 상호 작용을 위해 구현된 GIOP(Global Inter-ORB Protocol) 메시지가 TCP/IP 네트워크 상에서 교환되는 방법을 명시한 것으로서 각 시스템이 상이한 ORB를 사용하는 경우 연동이 용이하지 않기 때문이다.

Wf-XML HTTP 바인딩을 이용하여 연동하는 방법은 여러 가지 장점을 가지게 된다. 먼저 프로세스 상태정보가 XML로 표현되기 때문에 사용자가 웹상에서 원격 프로세스의 상태를 쉽고 빠르게 파악할 수 있을 뿐만 아니라, 기계 판독이 가능한 형식으로 표현되므로 추가적인 이점을 가진다. 또한 시스템간의 동기적(synchronous) 및 비동기적(asynchronous) 연결이 가능함에 따라 프로세스 자동 수행의 수준 향상을 기대할 수 있다.

이러한 시스템간의 직접 연결을 통한 프로세스 자동화를 가능케 하는 Wf-XML의 장점을 활용할 수 있는 워크플로우 시스템을 구현하기 위해서는 WfMC 표준에서 언급되지 않은 여러 가지 요구사항이 충족되어야 한다. 본 논문에서는 프로세스 정의 및 그에 관련된 입출력 데이터의 형식 및 상호 인증 등 보안문제에 대한 합의가 있고, HTTP를 사용한다는 가정 하에 논의를 전개한다.

본 논문에서는 한국정보통신대학원대학교 소프트웨어 시스템 연구실에서 분산객체기술에 기반하여 개발된 워크플로우 시스템인 ICU/COWS(Information and Communications Univ. Connector-Oriented Workflow System)를 동기적 및 비동기적 상호작용이 가능하도록 구현하기 위하여 "Message Handler Manager"을 시스템에 추가하였다. Message Handler Manager는 수행객체인 "Message Handler"를 생성하고 이 수행객체로 하여금 원격 시스템에 있는 Message Handler와 Wf-XML 메시지를 교환하도록 디자인하였다.

본 논문의 2 장에서는 워크플로우 시스템간의 상호연동이 필요한 비즈니스 프로세스의 예, Wf-XML 표준 사양, 동기적 및 비동기적 상호연동 모델 등을 소개한다. 3 장에서는 Message Handler의 생성과정, 기능 및 엔진 구성요소와의 관계를 표현한 시스템 구성도 및 클래스 다이어그램을 통해 ICU/COWS에 대한 적용 모델을 설명한다.

2. Workflow Interoperability

2.1 E-Business Process

그림 1은 통상적으로 이루어지는 발주회사와 수주회사의 비즈니스 프로세스 패턴이다. Wf-XML 표준을 이용하여 두 회사의 워크플로우 시스템을 직접 연결할 경우 많은 장점이 있다. 먼저 발주회사는 수주회사

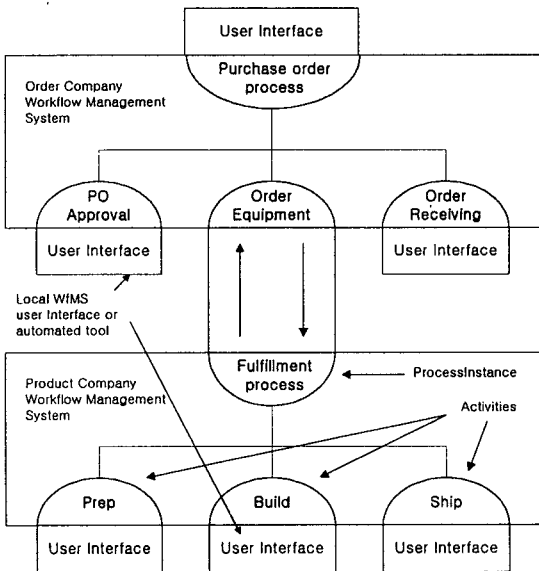


그림 1. 기업간 WfMS 상호연동 예

의 주문처리 프로세스가 자동으로 시작될 수 있도록 자신의 비즈니스 프로세스를 정의할 수 있으며, 제조회사의 주문처리 프로세스가 모두 수행된 후에도 수행 결과를 표준화된 형식으로 통보하므로 별도의 작업 없이 자동으로 주문회사의 다음 업무가 수행된다. 또한 두 프로세스가 수행되는 도중에 주문회사의 직원들이 제조회사의 주문처리 프로세스의 상태정보를 웹상에서 쉽게 조회할 수 있다[6].

그림 1을 비즈니스 프로세스 정의의 관점에서 보면 그림 2와 같다. 즉 워크플로우 서비스 A의 단위업무 중 두번째 업무는 자동업무로서 워크플로우 서비스 B의 프로세스를 자동으로 시작하도록 하고 그 프로세스가 종료한 후 결과를 받으면 다음 업무가 시작된다.

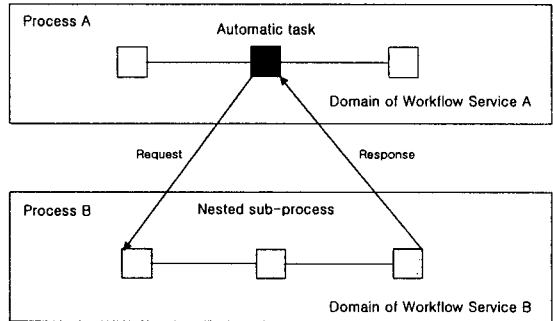


그림 2. Nested sub-process

이러한 시스템간의 직접 연결을 통한 프로세스 자동화는 Wf-XML HTTP 바인딩을 사용함으로써 가능하며, 기존 워크플로우 시스템에서 Wf-XML HTTP 바인딩을 이용하여 상호작용하기 위해서는 WfMC 표준에서 언급되지 않은 여러 가지 요구사항이 충족되어야 한다. 즉 전송프로토콜의 제한, 보안, 에러처리, 데이터 타입 및 데이터의 길이 제한 등의 요구사항들에 대한 시스템 구현자들 사이에 합의가 있어야만 안정적이고 신뢰성있는 시스템간의 연동을 구현할 수 있다.

또한 많은 기업들이 전자상거래를 위해 사용하고 있는 EDI(Electronic Data Interchange) 시스템에서의 수주 및 발주 업무와 같은 일괄처리 업무를 위해서는 비동기적 상호작용이 필요하다. 그러나 TCP/IP를 기반으로 한 HTTP 프로토콜은 기본적으로 동기적 상호작용만이 가능하므로 이를 응용프로그램 레벨에서 비동기적으로 처리할 수 있어야 한다. 따라서 워크플로우 시스템에는 이러한 동기적 및 비동기적 상호작용을 처리할 수 있는 별도의 모듈이 필요하다.

2.2. Wf-XML 표준

WfMC에서는 다른 워크플로우 시스템에 있는 프로세스의 인스턴스 생성 및 생성된 인스턴스의 제어 및 상태정보 검색을 위해 다음과 같은 표준 operation들을 제공하고 있다.

- CreateProcessInstance
- GetProcessInstanceData
- ChangeProcessInstanceState
- ProcessInstanceStateChanged

또한 모든 Wf-XML 메시지는 XML 1.0 표준을 따르는 XML 문서 객체이며 근본 요소가 되는 WfMessage 태그는 WfTransport, WfMessageHeader 및 WfMessageBody 의 세 요소로 구성된다[1].

- WfTransport: XML 문서 교환에 사용할 프로토콜이나 사용자 정의 정보를 포함.
- WfMessageHeader: operation 에 사용되는 모든 태그를 포함하며, Request, ResponseRequired, Response, Key, ResponseLang 등의 요소를 포함.
- WfMessageBody: operation 및 관련 데이터를 포함.

예를 들어 CreateProcessInstance XML 메시지는 다음과 같이 구성될 수 있다.

```
<?xml version=" 1.0" ?>
<WfMessage Version=" 1.0" >
  <WfMessageHeader>
    <Request ResponseRequired=" yes" />
  </WfMessageHeader>
  <WfMessageBody>
    <CreateProcessInstance.Request
      StartImmediately=" true" >
      <ObserverKey>http://- /wfprocessor
      </ObserverKey>
    <ContextData>
      ...
    </ContextData>
    </CreateProcessInstance.Request>
  </WfMessageBody>
</WfMessage>
```

2.3. WfMS 간의 동기적 및 비동기적 연동

본 논문에서는 동기적 연동을 Request Message 가 전달되고 이를 수행한 결과를 Response Message 로 받는 것으로, 비동기적 연동을 Request Message 가 전달되면 일단 Ack. Message 를 받은 다음 수행결과를 Response Message 로 받고 이에 대한 Ack. Message 를 보내는 응용프로그램 레벨의 메시지 교환 방식으로 정의한다.

HTTP 프로토콜은 연결 지향적인(Connection-Oriented) 프로토콜로서 기본적으로 동기적 연동만이 가능하다. 그러나 요청사항을 처리하는데 다소 긴 시간이 걸리는 경우나 여러 오퍼레이션을 일괄 처리해야 하는 경우 비동기적인 연동메커니즘이 필요하다. 따라서 각각의 워크플로우 시스템에 별도의 모듈로서 Message Handler 를 두고 이를 처리하도록 디자인하였다.

Message Handler 를 통하여 그림 2 의 내포된(nested) 다른 워크플로우 시스템의 프로세스와의 상

호작용들을 일괄처리하기 위해서 Mission, Operation, Phase 의 세가지 개념을 도입하였다. Mission 은 원격지에서 실행될 프로세스의 시작에서부터 종료까지를 의미하며, Wf-XML request 메시지로 시작하여 Wf-XML response 메시지로 끝나게 된다. Operation 은 request 메시지, 이에 대한 acknowledgement 메시지, response 메시지 및 이에 대한 acknowledgement 메시지의 4 Phase 로 구성되며, 그 기능에 따라 세가지로 분류된다[2].

먼저 Register operation 은 Mission 의 처음과 끝을 알리는 operation 이며, Refer operation 은 프로

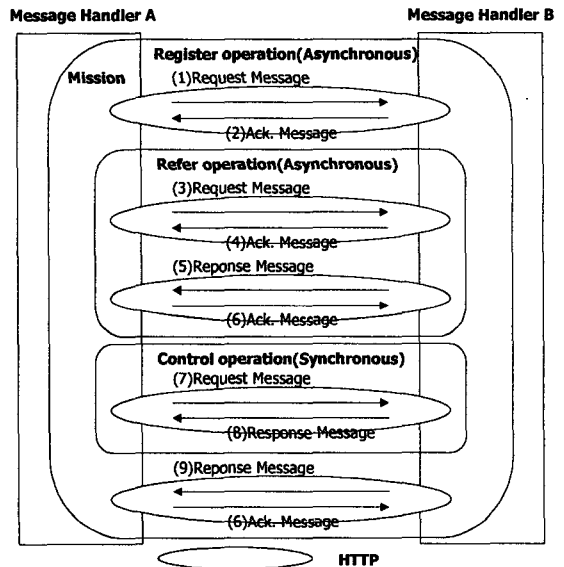


그림 3. 동기적 및 비동기적 연동 예

세스의 진행상태 및 결과를 검색하는 operation 이다. 또한 Control operation 은 프로세스의 정지, 취소 및 재시작 등 프로세스를 제어하는 operation 이다[2].

그림 3 은 이러한 개념 및 operation 과 앞서 설명한 Message Handler 를 그림으로 설명한 것이다.

3. ICU/COWS: Message Handler 기능 및 설계

ICU/COWS 는 분산 객체기술에 기반하고 Java 를 사용하여 개발된 분산 워크플로우 관리시스템으로서 그림 4 와 같이 구성되어 있다. GTMIG(Global Task Management Instance Generator)에 정의된 비즈니스 프로세스의 인스턴스를 생성하도록 요청하면, TMIF(TMI Factory)에 의해 각 단위 업무(Task)에 대한 수행객체(TMI: Task Management Instance)가 생성되어 해당업무의 수행을 관리한다.

본 연구에서는 원격지의 워크플로우 시스템과 상호 연동하는 경우 원격 프로세스를 제어하기 위해서 Message Handler 를 이용하도록 하였으며, 동시에 여러

비즈니스 프로세스 또는 그의 인스턴스들이 존재할 수 있으므로 Message Handler Manager 를 두고 원격 워크플로우 시스템과의 연동이 필요한 업무가 발생하면

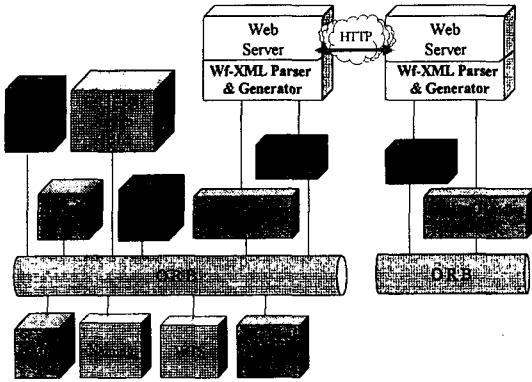


그림 4. ICU/COWS Component 구성도

Message Handler 을 수행객체로 생성하여 원격 프로세스와 상호작용하도록 설계하였다. 또한 Message Handler Manager 는 내부적으로는 엔진 구성요소와 Java CORBA 로 연결되어 있으며 외부 워크플로우 시스템과는 HTTP 프로토콜을 사용하여 상호 연동되도록 설계된다.

나게 되면 두번째 업무는 자동적으로 MhManager 에 원격 워크플로우 시스템과의 연동을 위한 수행객체 생성을 의뢰한다. MhManager 는 MhFactory 로 하여금 Mhandler 를 생성하도록 하고 원격 Mhandler 와 Wf-XML 메시지를 교환하도록 한다.

한편 B 시스템에서는 MhManager 가 최초의 메시지를 받아 MHandler 의 생성을 의뢰한다. MHandler 엔진에 해당 프로세스의 인스턴스 생성을 의뢰한 다음 생성된 프로세스 인스턴스에 대한 제어권한을 갖고 A 시스템의 MHandler 와의 상호작용을 담당한다.

또한 Message Handler 는 동기적 또는 비동기적 메시지 여부를 WfMessageBody 에 <Synchronous/> 또는 <Asynchronous/> 등의 태그를 삽입함으로써 판별하며, 각각의 Mission, Message 및 Request 에 ID 를 붙임으로써 동일한 요청에 대한 중복수행을 방지할 수 있다.

4. 결론 및 향후 연구방향

본 연구에서는 B2B 전자상거래 등으로 관심이 모아지고 있는 기업간 워크플로우 시스템 연동에 대하여 WfMC 에서 발표한 Wf-XML 표준을 이용하여 실제로 구현하는 경우 고려할 사항을 살펴보았다. 또한 Wf-XML HTTP 바인딩을 이용하여 시스템을 구현하는 경우 비동기적 상호작용을 위해 필요한 Message Handler 및 Message Handler Manager 의 기능 및 엔진 구성요소와의 관계를 제안하고 ICU/COWS 에서의 구현모델을 디자인하였다.

Message Handler 의 역할은 비동기적 상호작용을 위한 제어 뿐만 아니라 그 밖에 많은 역할을 하도록 설계되어야 한다. 예를 들어 해당 프로세스 인스턴스의 상태정보를 인터넷을 통하여 빠르고 원활하게 공급하기 위한 Process Instance Status Information Repository 의 기능도 가져야 하는데 엔진의 부담을 최소한으로 줄이는 메커니즘에 대한 연구 및 실제 구현이 필요하다.

또한 DCOM 등 Java CORBA 가 아닌 ORB 를 사용하는 시스템과 상호 연동하는 경우 발생할 수 있는 문제점을 조사하고 해결책을 제시하는 것도 필요하다.

참고문헌

- [1] Workflow Management Coalition. "Workflow Standard - Interoperability Wf-XML Binding", TC-1023, May 2000.
- [2] Nobuyuki Kanaya et al. "Workflow Standard - Interoperability: XML-HTTP binding", Submission 2.0, Feb. 2000.
- [3] Workflow Management Coalition. "Workflow Standard - Interoperability: Abstract Specification", TC-1012, Oct. 1996.
- [4] Workflow Management Coalition. "Workflow Standard - Interoperability: Internet e-mail MIME Binding". TC-1018, Jan. 2000.
- [5] Workflow Management Coalition. "The Workflow Reference Model", TC-003, Jan. 1995.
- [6] James G. Hayes et al. "Workflow Interoperability Standards for the Internet", Journal of IEEE Internet Computing, p. 37-45, May-June 2000.

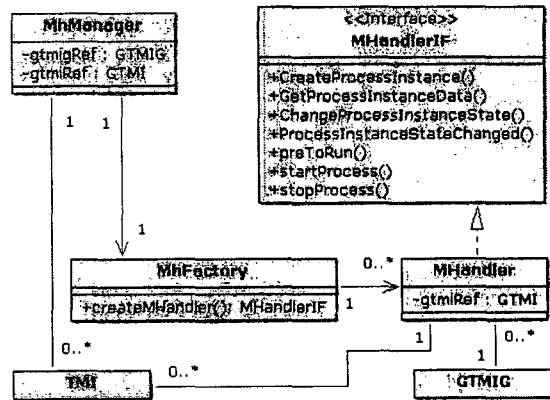


그림 5. MHandler 의 클래스 구성도

그림 5 는 MhManager(Message Handler Manager) 및 그의 수행 객체인 MHandler(Message Handler)와의 관계 및 엔진 구성요소인 TMI 및 GTMIG 와의 관계를 표현한 UML 클래스 다이어그램이다.

그림 2 의 비즈니스 프로세스를 가지고 구체적인 수행절차를 알아보면 먼저 미리 정의된 비즈니스 프로세스를 가지고 프로세스의 인스턴스를 생성하면 세계의 수행객체(TMI)가 생성된다. 그 두번째 수행객체가 자동업무(automatic task)로서 첫번째 업무가 끝