

# 원격 프로세스 프록시를 이용한 실용적 워크플로우 상호운용 모델

권현걸<sup>0</sup> 이성득 한동수  
한국정보통신대학교  
{hgkwon, sdlee, dshan}@icu.ac.kr

서범수  
전자통신연구원  
bsseo@etri.re.kr

## Practical Workflow Interoperation Model Using Remote Process Proxy

Hungul Kwon<sup>0</sup> Sungdoke Lee Dongsoo Han Beomsu Seo  
Information and Communications University ETRI

### 요 약

최근, 인터넷이라는 개방형 네트워크가 보편화 되면서 인터넷을 매개로 한 전자적인 거래의 형태가 기업 대 소비자(B2C) 거래에서부터 기업 대 기업(B2B) 거래 그리고 기업 대 정부(B2G) 거래까지 다양하게 발전하고 있다. 본 논문에서는 WfMC에서 제시하는 표준을 준수하면서 동시에 워크플로우 시스템의 사용자인 프로세스 설계자 그리고 워크플로우 프로세스 참여자 모두에게 쉬운 인터페이스를 제공할 수 있는 워크플로우 시스템의 아키텍처 및 모델을 제시한다. 본 모델은 P2P기반의 현실적인 대안을 위해서 설계되었으며, 기존 연구에서 가볍게 취급되어 왔던 워크플로우 시스템 사용자를 위한 인터페이스를 고려하여 설계되었다.

### 1. 서 론

기업간(B2B)에 전자적인 방법을 거래에 사용한 예는 인터넷이 널리 보급되기 이전부터 있어 왔다. 대표적인 예로는 EDI로서, EDI는 주로 거래에 수반되는 문서의 교환에 그 목적을 두고 있었다. 최근에 인터넷이라는 개방형 네트워크가 보편화 되면서 인터넷을 매개로 한 전자적인 거래의 형태가 나타나기 시작하였고, 그 형태가 기업 대 소비자(B2C) 거래에서부터 기업 대 기업간(B2B) 거래 그리고 기업 대 정부(B2G) 거래까지 다양하게 발전하고 있다. 이러한 시대적인 배경을 바탕으로 WfMC(Workflow Management Coalition)는 워크플로우 시스템간 상호운용성을 위한 표준을 수립함에 있어서 통신 프로토콜 및 세부 구현 방법에 대하여 독립적인 방안을 마련하였고, 그 구체적인 예가 Wf-XML(eXtensible Markup Language) 이다[1][2][3].

본 논문에서는 WfMC에서 제시하는 표준을 준수하면서 동시에 워크플로우 시스템의 사용자인 프로세스 설계자 그리고 워크플로우 프로세스 참여자 모두에게 용이한 인터페이스를 제공할 수 있는 워크플로우 시스템의 아키텍처 및 모델을 제시하고자 하는데 그 목적이 있다. 본 연구에서 중점적으로 다루고자 하는 상호운용성의 문제는 상호운용하는 파트너의 프로세스를 어떻게 기술할 것인가와 관련이 있다. 기존의 WPD(Workflow Process Definition Language) 표준에는 파트너의 프로세스를 표시할 수 있는 규칙이 없으며 가상기업을 가정한 이전의 여러 연구들에서는 조직의 정보에 대한 독립성 유지에 어려움이 있다. 기존의 많은 관련 연구들이 주로 가상기업 또는 마켓플레이스를 염두에 두고 진행된 것이었다면, 제안한 모델은 P2P 기반의 현실적인 대안을 위해서 설계되었으며, 기존 연구에서 가볍게 취급되어 왔던 워크플로우 시스템 사용자들을 위한 인터페이스를 고려하였다. 이러한 목표를 위해서 외부에 있는 거래 파트너의 프로세스를 내부 워크플로우 시스템에서 표현할 수 있는 대리자(proxy) 역할의 원격 프로세스 프록시(WfRemoteProcessProxy)를 중심으로

외부 프로세스에 대한 원격 요청자(WfRemoteRequester)를 새롭게 소개하고 Wf-XML메시지의 생성 및 전송을 담당하는 엔진 외부 어댑터(IfAdapter)를 두었다. 또한 실제 비즈니스 환경에서 나타날 수 있는 많은 파트너와의 동적인 상호운용(Dynamic interoperation)을 지원할 수 있는 구조로 설계되었다.

따라서, 제안 모델은 점점 더 확대되어 가고 있는 인터넷의 사용과 워크플로우 시스템의 기업간 전자 상거래용 플랫폼으로서의 가능성을 구체적으로 제시하는데 있어서 그 의의가 있으며, 표준의 확장에 따라 탄력적으로 대처할 수 있는 아키텍처로서 실제 워크플로우 시스템에 적용될 수 있기를 기대한다.

### 2. 원격 프로세스 프록시 모델

WfMC에서는 워크플로우 운용 시스템을 위하여 5가지의 인터페이스를 정의하였으며, 각 인터페이스에 대한 표준을 확립하였다[1]. 기본적으로 인터페이스 및 데이터 교환 포맷의 표준화 목적은 워크플로우 산물간의 상호운용을 원만히 이루기 위한 것이다. 그렇지만 각 워크플로우 벤더는 독립된 데이터 모델 및 독자적인 기능을 갖는 시스템으로 개발되었기 때문에 서로 다른 워크플로우 시스템간의 상호운용은 쉽게 달성되지 않았다. 또한, 종래의 연구는 워크플로우 상호운용에 있어서 하나의 워크플로우 시스템 구축의 시점에서 보지 않고, 전자시장이나 가상기업의 시점에서 보고 실행되었다.

결국, WfMC에 의해 개발된 워크플로우 상호운용 표준에 적용하면서, 단일 워크플로우 시스템의 견지에서 시스템간 상호운용의 조건을 식별하는 것이 필요하게 되었다.

본 논문에서는 서로 다른 워크플로우 시스템간의 상호운용을 위하여 새롭게 워크플로우 엔진 및 외부 소프트웨어 구성

요소들을 도입하기로 한다.

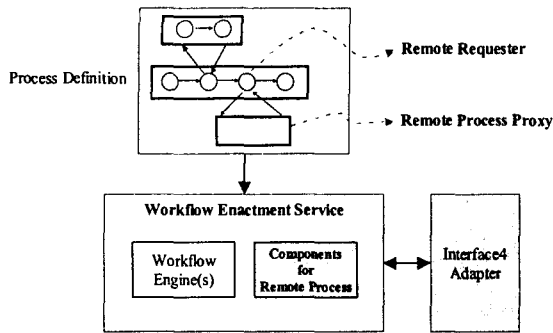


그림 1 개념적 모델

그림 1은 제안한 원격 프로세스 프록시의 개념적 모델이며, 각 구성요소는 다음과 같다.

- ◆ 원격 프로세스 프록시(wfRemoteProcessProxy)
- ◆ 원격 요청자(WfRemoteRequester)
- ◆ 외부 어댑터(IIf4Adapter)

여기에서, 원격 프로세스 프록시 및 원격 요구자는 워크플로우 엔진 내부에 존재하며, 각각 build-time 특징 및 runtime behavior로 구분하여 표현된 구성요소이다. 그리고, 외부 어댑터는 엔진의 외부에 있다. 각 요소는 OMG카텍처 [4]를 참고로 구성되었다. 그밖의 위 모델에 포함된 것이 있는데 워크플로우 상호운용 계약이다. 이는 소프트웨어 요소가 아니라, 단지 교류 상대간의 운용 계약으로서 시스템의 주소, 프로세스 정의, 취지 등의 프로세스를 인스턴스하기 위해 필요한 것들이다. 각 구성요소의 역할은, 다음 절에서 설명하기로 한다.

### 2.1 원격 프로세스 프록시(WfRemoteProcessProxy)

원격 프로세스 프록시는 워크플로우 시스템에서 원격 프로세스의 일을 대리 수행하는 역할을 하기 때문에 원격프로세스의 복제처럼 보인다. 다시말해서 프로세스 프록시는 일을 수행하는데 있어서 실제의 프로세스가 아님을 의미한다. 원격 프로세스 프록시가 실제활동을 갖지않는 이유는 동적 상호운용 접근수단과 많은 관련이 있다. 동적 상호운용 메카니즘을 지원하기 위하여는 실행시간(runtime)에 상대 프로세스를 결정해야 할 필요가 있다. 따라서, 인스턴스화 될 실제의 프로세스들을 결정하지 않으며, 프로세스 활동들을 포함하지 않는다. 한편, 새로운 원격 프로세스 프록시가 정의될 경우 항상 한개의 이름을 가져야 한다. 예를들면, 로컬 전송 비즈니스 프로세스를 표현하는 "LocalDelivery" 라고 이름 지어진 2개의 원격 프로세스 프록시가 존재하면 안된다. 원격 프로세스 프록시에 대한 한개의 이름은 실행시간에서 상대식별 코드를 구비하면서 정확한 프로세스 정의를 결정하기 위한 것이다. 일단 원격 프로세스 프록시의 이름이 정의되면, 그 정의된 것에 의해 상호운용을 필요로 하는 어떤 비즈니스 프로세스에 재사용이 가능하다.

#### 2.1.1 Build-time 특징

원격 프로세스 프록시는 워크플로우 시스템 모델[1]의 확장된 요소이다. 따라서, WPDL로 그것을 기술할 필요가 있는데 그 기술을 위한 표현문법은 다음과 같다.

```
<Workflow Remote Process Proxy Definition> ::=
  WORKFLOW <process id>
    <Workflow Process Definition Header>
    REMOTE_PROCESS_NAME <remote process name>
    [<Transition Information List>]
  END WORKFLOW
<process id> ::= <identifier>
<remote process name> ::= <string>
```

#### 2.1.2 Runtime behavior

원격 프로세스 프록시는 한 프로세스를 수행함에 있어서, 원격요청자에 의해 실행시에 활성화 된다. 원격 프로세스 id 및 상대가 설정되고, 프로세스 인스턴스를 생성하기 위한 Wf-XML 메시지가 보내지면, 원래 프로세스 인스턴스는 상대의 워크플로우 시스템 내에서 활성화 되어진다. 상대의 워크플로우 시스템으로부터 응답 메시지를 받은다음, 원격 프로세스 프록시는 상대의 프로세스 인스턴스 키, 상태 및 프로세스기술과 같은 워크플로우 프로세스의 실제 정보를 얻어, 원격의 프로세스 인스턴스 특성으로서 정보를 유지한다.

원격 프로세스 프록시는 그 상태의 모니터링 및 요소들의 변화에 대한 인터페이스를 제공한다. 결국, 관리 또는 모니터링 둘은 쉽게 원격 프로세스 프록시를 통하여 상대 프로세스 의뢰에 접근할 수 있다. 상대의 프로세스 의뢰가 종료할 경우 WfXML메시지에 의해 종료상황을 받아서 원격 요청자에게 프로세스 종료를 알린다.

### 2.2 원격 요청자(WfRemote Requester)

원격 요청자는 OMG워크플로우 모델의 WfActivity 같은 기능을 기본적으로 가지면서, 상호운용을 위한 몇가지 추가 기능을 가지고 있다. 그 추가 기능이란, 하나의 프로세스 스텝 작업 및 원격의 프로세스 요청자(또는 관찰자)을 동시에 표현하는 역할이다. Build-time에서 요청자 활동으로 정의된다. 그리고, run time에서 서버 프로세스 의뢰를 위한 참고를 갖는다. 즉, 원격 프로세스 의뢰에 대하여 실제의 참고를 갖는게 아니라, 2.1절에서 언급한 원격 프로세스 프록시의 의뢰에 대한 참고를 갖는다.

원격 프로세스에 대한 데이터 구조 및 상대와의 관계는 그림 2에서 보여주고 있다.

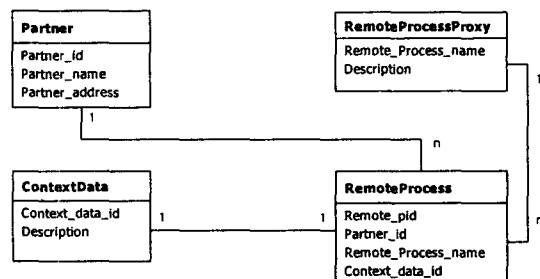


그림 2 원격 프로세스 및 상대의 데이터 개요

2.2.1 Build-time 특징

원격 요청자는 프로세스 내 활동으로서, 다음과 같이 WPDL 내 ACTIVITY와 같은 방법으로 표현할 수 있다. 그러나, 서브 프로세스의 요청자이기 때문에, 속성 <Activity Kind Information>의 표현을 제한하였다.

```

ACTIVITY <activity id>
  [NAME <name>]
  [DESCRIPTION <description>]
  [LIMIT <duration>]
  <Activity Kind Information>
  [<Access Restriction part>]
  [<Transition Restriction part>]
  [<extended attribute list>]
END_ACTIVITY
    
```

2.2.2 Runtime behavior

프로세스 유동 제어가 한 작업으로써 입력 되었을때 원격 요청자에서 수행되는 것은 Partner\_ID 및 원격 서브 프로세스 이름을 이용한 우선 프로세스 결정의 선택이다. 여기에서 Partner\_ID는 특별한 목적에 관련있는 데이터이고, 원격 서브 프로세스 이름은 구성의 내부코드이다.

명확한 프로세스 결정의 선택은 그림 2 의 데이터 구조와 같이 간단하다. 데이터 구조상에서의 정보는 각 구성이 상호 운용 계약을 교환할 때 저장 및 지속된다.

원격 프로세스 결정이 선택된 후, 원격 요청자는 WfActivity와 같은 동작을 실행한다.

2.3 외부 어댑터(If4Adapter)

외부 어댑터는 소프트웨어 컴포넌트로서 워크플로우 엔진의 외부에 존재하며, Wf-XML 메시지의 생성 및 전송을 담당한다. 메시지가 도달하였을때 그 유효성을 체크하여 도달된 메시지를 해석한 다음에 어떤 운용을 실행할 것인지 결정한다. 컴포넌트를 유연하게 하기 위해 그림 3 과 같은 3개의 층으로 설계하였는데 각 layer의 간략한 설명은 다음과 같다.

- ◆ Transport Layer  
Sender 및 Receiver의 두 구성요소로 구분된다. Sender는 이미 체결된 네트워크 프로토콜을 이용하여 Operation Layer에서 생성된 Wf-XML 메시지를 상대 워크플로우 시스템에 보낸다. Receiver는 도달하는 Wf-XML 메시지를 기다리며, 메시지가 도달하였을때 연산 메니저에게 그 메시지를 보낸다.
- ◆ Operation Layer  
Operation layer의 주요 기능들은 creating/parsing Wf-XML 메시지이다. WAPI layer의 컴포넌트를 이용하여, 도착된 메시지에 대응하는 워크플로우 인터페이스를 제공한다. 그리고 도착된 Wf-XML 메시지의 정당성을 체크한다.
- ◆ WAPI Layer  
WAPI layer는 프로세스 결정 및 의뢰에 액세스하기 위한 통로로서 operation layer에게 워크플로우 엔진의 인터페이스를 제공한다. 로컬 프로세스 핸들러 및 원격 프로세스 핸들러의 두 컴포넌트가 있다.

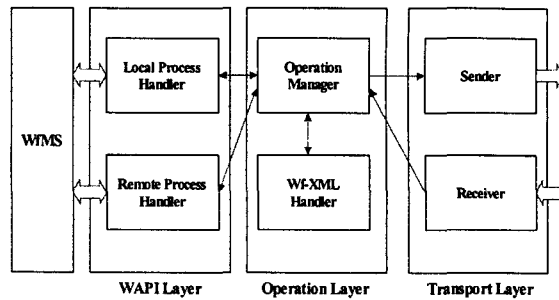


그림 3 외부 어댑터의 아키텍처

2.4 워크플로우 상호운용 계약 (Workflow Interoperation Contract)

시스템의 서로간 상호운용을 하기 이전에 WfMC[2]에서 규정된 상호운용 계약과 같은 종류의 계약이 우선적으로 필요하다. 첫째, 어떤 종류의 데이터가 필요한가에 따라서 원격 프로세스 정의를 표현하는 방법이 있는데, 이것은 프로세스 정의 인식자로 표현한다. 예를들면, WfXML내, WfMessageHeader의 키 요소값을 위해 사용된다. 둘째로, 프로세스 실행을 작성할때 필요한 계약 데이터이다. 이것은 서로 다른 프로세스간에 공동 작업을 하기 위해 필요하다. 계약 정보는 그림 2 의 형식으로 저장되어, runtime 프로세스 정의를 선택하기 위해서 사용된다.

3. 결론

본 논문에서는 WfMC에서 제시하는 표준을 준수하면서 동시에 워크플로우 시스템의 사용자인 프로세스 설계자 그리고 워크플로우 프로세스 참여자 모두에게 쉬운 인터페이스를 제공할 수 있는 워크플로우 시스템의 아키텍처 및 모델을 제시하였다. 제안한 모델은 P2P 기반의 현실적인 대안을 위해서 설계되었으며, 기존 연구에서 가깝게 취급되어 왔던 워크플로우 시스템 사용자들 위한 인터페이스를 고려하였다.

이후, 점점 더 확대되어 가고 있는 인터넷의 사용과 워크플로우 시스템의 기업간 전자 상거래용 플랫폼으로서의 가능성을 구체적으로 제시하는데 있어서 본 모델의 의의가 있으며, 표준의 확장에 따라 탄력적으로 대처할 수 있는 아키텍처로서 실제 워크플로우 시스템에 적용될 수 있기를 기대한다.

참고문헌

- [1] WfMC, The Workflow Reference Model (Doc. Number TC00-1003), 1995.
- [2] WfMC, Workflow Standard □ Interoperability Abstract Specification (Doc. Number WfMC-TC-1012), 1999.
- [3] WfMC, Workflow Standard □ Interoperability Wf-XML Binding (Doc. Number WfMC-TC-1023), 2001.
- [4] OMG, Workflow Management Facility Specification, V1.2, 2000.
- [5] Myungjae Kwak, Dongsoo Han, Jaeyong Shim, A Framework for Dynamic Workflow Interoperation Using Multi-Subprocess Task, Proceedings of Twelfth International Workshop on RIDE-2EC, 2002.