

# 워크플로우 시스템 사이의 상호운영성을 위한 데이터 교환 프레임워크

## (A Data Exchange Framework for Interoperability between Workflow Systems)

나 흥 석<sup>†</sup> 김 중 일<sup>\*\*</sup> 이 이 섭<sup>\*\*\*</sup> 백 두 권<sup>\*\*\*\*</sup>  
(Hong-Seok Na) (Joong-Il Kim) (Lee-Sub Lee) (Doo-Kwon Baik)

**요 약** 웹 기반의 워크플로우 시스템은 기업간 전자상거래에 필수적인 요소로 부각되고 있다. 그러나 상이한 벤더들에 의해 제작된 워크플로우 엔진들은 상호운영성이 보장되지 않기 때문에 전자상거래의 제 기능을 발휘하지 못한다. 이와 같은 문제를 해결하기 위한 노력으로 WfMC(Workflow Management Coalition)에서는 워크플로우 시스템간의 상호운영성을 위한 기본 명세를 제공하고 있으나, 인터넷상에서의 구체적인 활용방안은 제시되지 못하고 있다.

본 논문에서는 워크플로우 시스템 사이의 데이터 교환을 위한 프레임워크(WDEF : Workflow Data Exchange Framework)를 제시한다. WDEF는 데이터 레지스트리를 중심으로 하여 워크플로우간 상호운영성을 제공하기 위한 메타데이터를 기술하고, 네트워크 상의 자원을 기술하기 위한 기반 기술인 RDF(Resource Description Framework)를 이용하여 워크플로우간의 메시지를 구성한다.

데이터 레지스트리를 이용함으로써, WDEF는 WfMC에서 정의하는 워크플로우 상호운영성을 위한 요구조건들을 모두 수용하면서 보다 확장성 있고 유연한 구조를 가지며, RDF에 기반한 자동적인 XML 인코딩을 통해 전자 상거래에 이용되는 다른 XML문서와도 완벽한 호환성을 가진다.

**Abstract** Currently, web-based workflow system becomes the essential element for electrical commerce between companies, but workflow engines, produced by different vendors, dose not guarantee the interoperability.

In this paper, we propose a framework(WDEF : Workflow Data Exchange Framework) that supports data exchange between heterogeneous workflow systems. WDEF uses a ISO/IEC 11179-based data registry to describe metadata for workflow interoperability, and uses RDF(Resource Description Framework) to encode and compose the messages for workflow interoperability.

Using data registry, WDEF supports the requirements for workflow interoperability defined in WfMC(Workflow Management Coalition), and has flexible structure with extensibility. Also, RDF-based automatic XML encoding makes it possible for messages to be shared with other XML document in CALS/EC.

### 1. 서 론

<sup>†</sup> 비 회 원 : 고려대학교 기초과학연구소 연구원  
nhs@swsys2.korea.ac.kr

<sup>\*\*</sup> 비 회 원 : 고려대학교 컴퓨터과학기술연구소 연구원  
jikim@swsys2.korea.ac.kr

<sup>\*\*\*</sup> 비 회 원 : 삼성SDS 연구원  
eesub@yahoo.com

<sup>\*\*\*\*</sup> 종신회원 : 고려대학교 컴퓨터학과 교수  
baik@swsys2.korea.ac.kr

논문접수 : 1999년 12월 24일

심사완료 : 2000년 12월 26일

비즈니스 환경이 인터넷상으로 옮겨짐에 따라, 전자상거래를 지원하기 위한 다양한 시도들이 이루어지고 있다. 그 중의 하나가 바로 웹 기반 워크플로우이며 워크플로우 관리 시스템들이 기존의 ERP(Enterprise Resource Planning), PDM(Product Data Management) 등의 시스템들과 연동하여 보다 효율적인 비즈니스 환경을 구축하려는 노력들이 앞다투어 이루어지고 있다[1].

워크플로우란 “기업 내외적으로 정의된 업무들과 그에 관련된 사람, 정보 및 기타 자원의 흐름을 통합적으

로 관리, 지원해 주는 업무처리 자동화 시스템”이다[2]. 현재 워크플로우는 전사적, 또는 그 이상의 조직적 관점에서 부분이 아닌 프로세스 전체를 통합하고 자동화하며 관리, 지원한다는 관점으로 확장되고 있다. 이러한 워크플로우는 그 성격에 따라 관리적(Administrative), 비정형(Ad-hoc), 생산(Production), 협업적(Collaborative) 워크플로우로 구분되는데, 이들 분류와 워크플로우 제품의 기능성에는 밀접한 관련이 있다. 예를 들어, 계좌 이체, 온라인 클레임 처리 및 복잡한 애플리케이션과의 연계 등과 같이 기업에 중대한 업무 처리에 필요한 생산 워크플로우와 같은 제품도 있고, 문서 중심의 간단한 작업이나 결재 처리, 자동 문서 전달 및 분배와 같이 비정형/관리적 워크플로우에 속하는 제품도 있다.

그러나 다양한 워크플로우 시스템의 등장은 복수의 워크플로우 시스템의 연계 및 워크플로우 시스템간의 워크플로우 어플리케이션 공유의 필요성을 제기하고 있다. 즉 하나의 워크플로우 시스템 상에서 개발된 워크플로우 어플리케이션을 다른 워크플로우 시스템에서 수정 없이 수행하거나 다른 워크플로우 시스템에서 처리된 결과를 다른 워크플로우 시스템의 입력으로 사용하려는 사용자의 요구가 많아지고 있다[17].

따라서, 기존에 회사 단위 또는 부서 단위로 독립적으로 운영되던 워크플로우 시스템들이 상호 운영되기 위해서는 상대 워크플로우 시스템들이 사용하는 데이터 구조, 메시지 형식, 프로세스 구조 및 정의 등에 대한 이해가 필요하기 때문에 메타데이터 수준에서 상호 이해할 수 있는 노력이 요구된다[15].

워크플로우 관련 단체의 모임인 WfMC(Workflow Management Coalition)에서는 서로 다른 벤더들의 상이한 워크플로우 관리 시스템들 간의 통신을 가능하게 하고자 이들의 상호운영성에 대한 논의를 해왔고, 그 결과로 워크플로우 시스템에 대한 표준 및 워크플로우 시스템의 상호운영성에 대한 표준[2]을 제정하였다.

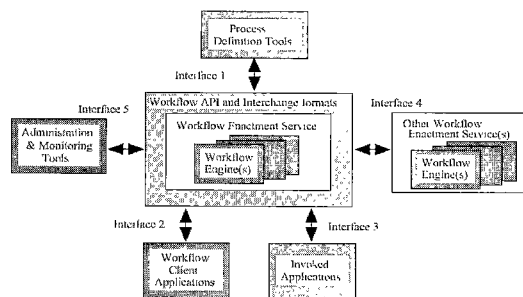


그림 1 워크플로우 참조 모델 - 구성요소 및 인터페이스

그림 1은 WfMC(Workflow Management Coalition)에서 표준으로 제정한 워크플로우 관리시스템을 위한 참조모델이다[5]. 이 구조는 주요 워크플로우 시스템의 주요 구성요소와 인터페이스를 보여준다. WfMC는 워크플로우 엔진과 나머지 구성 사이의 5가지의 인터페이스와 용어에 대하여 표준안을 제정하고 있으며, 이들 중 인터페이스 4에서 다른 워크플로우와의 상호운영성에 필요한 요소들을 정의한다.

현재 WfMC에서 제시한 워크플로우 상호운영성을 다루고 있는 워크플로우 표준 - 인터페이스 4(Workflow Standard - Interface 4)[2]에 대한 실질적인 구현 방안으로 MIME[3]을 이용한 방법과 WF-XML[4]을 이용한 방법이 있다. 하지만 언어의 정의만으로는 비즈니스의 종류 및 각 시스템의 특징을 담을 수 없고, 표준을 위한 가장 기본적인 내용들만을 대상으로 하고 있어 실제 시스템에 응용되기 어려우며, 특히 MIME의 경우는 구현도 어려워 효과적인 상호운영성을 제공하지 못한다.

전 세계적으로 활성화되고 있는 전자상거래의 종류는 너무나 방대하며, 각 기업들이 웹을 기반으로 한 워크플로우 관리 시스템들 간의 통신을 위해서는, 표준에서 제시하는 명세를 수용하는 것 외에도 시스템간의 효과적인 통신 인터페이스를 위해서 확장성 있는 방법론이 절실히 필요하다.

본 연구에서는 보다 확장성 있고 유연한 워크플로우 간의 데이터 교환을 지원하기 위하여, RDF와 메타데이터 레지스트리를 도입하였으며, 이를 기반으로 워크플로우간의 데이터 교환을 위한 프레임워크(WDEF : Workflow Data Exchange Framework)를 제안한다.

## 2. 관련 연구

### 2.1 워크플로우 시스템

워크플로우 시스템은 다양한 비즈니스 프로세스 자동화에 공통적으로 요구되는 기능들을 미리 갖추어 놓음으로써 비즈니스 프로세스의 자동화를 컴퓨터 상에서 효율적으로 구현하도록 도와준다. 초창기의 워크플로우 시스템으로 SCOOP[18], OfficeTalk-D[19] 등이 개발된 이후로 수많은 워크플로우 시스템이 발표되었다. IBM사는 워크플로우 시스템에 트랜잭션 개념을 도입한 FloMark를 개발하였으며 Keyfile사는 마이크로소프트사의 Exchange서버를 워크플로우 엔진으로 활용한 KeyFlow를 개발하였다. 이밖에서 HP사의 AdminFlow, Novell사의 GroupWise, Lotus사의 Notes등도 잘 알려진 워크플로우 시스템이다.

한편 대학교가 중심이된 워크플로우 시스템 개발도

활발히 전개되고 있다. 조지아 대학에서는 CORBA에 기반한 METEOR[20] 워크플로우 시스템을 개발하였고, Stuttgart 대학의 ConTracts[21]는 복수의 트랜잭션을 하나의 트랜잭션으로 통합하는 구조를 제공한다. 콜롬비아 대학에서는 워크플로우 모델로서 ICN(Information Control Network)를 제안하고 이것에 기반한 Chautauqua[22] 시스템을 개발하고 있다.

국내에서는 대부분의 기업들이 외국의 워크플로우 시스템을 구입하여 사용하고 있지만, 국내 비즈니스 프로세스와 다른 점이 많아 적용하는데 어려움을 겪고 있는 실정이다. 국내에서 개발되는 워크플로우 시스템은 WfMC 참조모델을 따라 구현된 세종 워크플로우 관리 시스템[16]과 웹기반으로 범용성을 갖도록 설계된 HiFlow[17]가 대표적이나 구현의 한계로 인해 실제 적용 단계까지는 이르지 못하고 있다.

국내의 대부분의 연구들이 워크플로우의 비즈니스 프로세스 정의 및 인스턴스 생성 및 실행에 중점을 두고 진행이 되고 있으며, 워크플로우 상호운영성에 관해서는 워크플로우 표준에서 정의한 상호운영성 명세를 그대로 구현해서 사용하는 수준의 연구가 진행되고 있다.

## 2.2 워크플로우 관련 표준

워크플로우 시스템을 위한 표준에는 워크플로우 자체 표준과 워크플로우와 외부 애플리케이션간의 연동을 위한 표준이 있다. 워크플로우 표준인 WfMC에서는 표준 워크플로우 구조를 정의하고 워크플로우에서 사용되는 인터페이스를 레퍼런스 모델이라는 이름으로 정의한다. CORBA와 같은 분산객체 관련 기술들의 표준을 제정하는 OMG에서는 분산객체 기술을 이용한 워크플로우와 인터페이스에 관한 표준을 JFlow(Joint Flow)라는 이름으로 정의한다.

워크플로우 상호운영성에 대한 표준화 노력은 WfMC와 OMG에서 추진해오고 있으며, WfMC는 추상적인 수준에서의 워크플로우 표준 명세를 정의해 왔으며, OMG에서는 XML이나 CORBA를 이용하여 워크플로우 시스템 사이의 상호운영성을 구현하려는 노력을 진행하고 있다.

WfMC 워크플로우 상호운영성 표준은 상이한 워크플로우 서비스 사이의 일대일 상호작용을 위한 추상적인 프로토콜을 정의한다[12]. WfMC 레퍼런스 모델 중에서 워크플로우 시스템들간에 프로세스 관련 정보를 주고받고, 프로세스를 연계하기 위한 표준과 API들의 집합이 인터페이스4이다. 인터페이스4는 서로 다른 워크플로우 엔진들이 상호 연동하기 위해 필요한 기능들의 규약을 설명한다.

인터페이스4를 통해 하나의 워크플로우 시스템은 다른 워크플로우 시스템에서 “정의된 프로세스의 선택”, “선택된 프로세스의 활성화”, “프로세스의 수행” 등의 작업을 수행한다. WfMC에서는 인터페이스4를 이용해 프로세스간의 상호연동을 설명한다.

WfMC의 인터페이스4는 프로세스 수준에서의 상호작용을 정의하며, 비즈니스 도메인의 내부적인 처리 과정은 다루지 않는다. 여러 워크플로우 관리 시스템 사업자들은 MIME을 이용한 상호운영성 명세의 구현을 워크플로우 엔진에 적용해 왔으며, XML 기반으로 정의된 Wf-XML이 현재 표준화 작업 중이다.

OMG(Object Management Group)는 서로 다른 워크플로우 시스템들이 플랫폼에 상관없이 상호 연동될 수 있도록 워크플로우의 핵심구성 요소를 정의해 객체 지향 워크플로우의 표준(Joint Flow)을 정의한다. JFlow(Joint Flow)는 WfMC 표준을 기반으로 작성되었다[13,14]. JFlow 표준은 워크플로우에서 기본적으로 존재해야 하는 중요한 객체(워크리스트 핸들러, 액티비티, 워크아이템 등)들을 정의하고, 객체들간의 상호 연동을 위한 인터페이스를 정의한다. JFlow 제정에 참여하고 있는 기업은 모두 WfMC의 멤버들이며 JFlow는 WfMC의 인터페이스 1,2,3,4,5를 모두 충족시키는 워크플로우 객체 모델이다. 즉, WfMC의 참조 모델이 워크플로우의 인터페이스를 정의하는 것에 중점을 둔 반면, JFlow의 메타 모델은 객체 중심의 워크플로우에서 공통적으로 가져야 할 기본 객체(CORBA 객체)들을 정의한다.

JFlow는 워크플로우 서비스 요청자와 제공자를 기본 모델로 하여 WfRequester, WfProcess, WfProcessMgr, 그리고 WfActivity와 같은 인터페이스를 정의한다. 이 표준은 워크플로우 프로세스 정의를 제외한 다양한 WfMC의 표준들을 포함하는 통합된 객체 모델을 정의하며, 이 객체 모델은 SWAP이나 Wf-XML과 같은 WfMC의 구체적인 전개 방향의 기초로서의 역할을 한다.

## 2.3 워크플로우 상호운영성 구현 동향

WfMC에서 정의한 인터페이스 4를 인터넷 상의 워크플로우 시스템들 사이의 상호운영성에 적용하기 위한 구체적인 방법으로는 인터넷 전자우편 방식인 MIME을 이용하는 방법, 표준화된 DTD를 정의하여 교환할 데이터를 XML을 이용하여 전송하는 방법(Wf-XML) 그리고 워크플로우 사이의 간단한 프로토콜 정의인 SWAP(Simple Workflow Access Protocol)이 제안되어 있는 상태이다. 이 장에서는 이 세 가지 방식을 설명하고, 장단점을 분석한다.

Wf-XML[4]은 WfMC(Workflow Management

Coalition)의 상호운영성 추상 명세(인터페이스 4)의 데이터 전송 요구사항들을 지원하기 위하여 디자인된 XML 언어에 대한 명세이다. 이 언어는 WfMC의 워크플로우 참조모델에 정의된 인터페이스 4를 지원하기 위해 명세서에서 기술한 기능 및 요소들을 구현하기 위한 기본으로 사용한다.

Wf-XML은 상호운영성 추상 명세[2]에 정의된 상호운영성의 기본적인 2개의 타입인 단순 연쇄 워크플로우(Simple Chained Workflow)와 중첩된 워크플로우의 상호운영성을 제공하기 위한 언어이다. 또한 이들 두 가지 타입의 워크플로우에 대해 동기적 및 비동기적 데이터 교환을 모두를 지원한다. 구현에 있어서는 데이터 교환을 개방적이고, 표준 기반의 모습을 유지하여, 구조화되고 견고하며, 주문화된 통신 형식을 표현 가능하도록 하기 위해서 XML(eXtensible Markup Language)를 사용한다. 즉, 워크플로우 데이터 교환에 사용되는 XML 태그 집합을 통일하기 위해 표준화된 DTD(Document Type Definition)를 정의하고 사용하는 방식이다.

Wf-XML은 워크플로우간의 통신 메시지의 구조적 조직을 위해 XML을 도입하였다. 하지만, 현재는 물론 미래에 대한 확장이 불가피 한 상황에서 계속적인 언어의 확장을 계획 및 적용하는 것은 시간 및 노력 면에서 상당한 비용을 초래한다. 또한 Wf-XML과 같은 새로운 언어 이외에 전자 상거래를 위해 XML/EDI와 같은 다른 언어를 적용하기 위해서는 통신의 계층을 분리해야 하는 번거로움이 발생한다. 따라서, Wf-XML의 본래 목적대로 보다 구현 독립적인 모습을 갖추려면, 단순한 XML의 정의에서 끝날 것이 아닌 RDF와 같은 정형화된 메타데이터 기술을 위한 프레임워크를 도입하는 것이 필수적이다.

MIME(Multipurpose Internet Mail Extension)을 이용한 데이터 교환은 두 개의 워크플로우간의 통신 메시지 교환을 위한 전송 방식으로 인터넷 전자우편을 사용한다. 워크플로우 추상 명세 인터페이스 4에 해당하는 내용을 전자우편의 텍스트 인터페이스로 사상하는 것을 기본으로 하여, 기본적인 워크플로우간의 통신 메시지를 구성한다.

하지만, 전자메일은 신뢰할 수 없는 통신 채널이기 때문에 예기치 않은 문제가 발생할 경우 해결해야 하는 추가의 노력이 필요하다. MIME 결함에서는 기존 워크플로우 엔진의 운영에 의한 오류 및 예외 상황에 대한 해결책 이외에 전자우편을 프로토콜로 이용함으로써 발생하는 오류와 예외 상황들을 해결하기 위한 내용들도

포함한다. 이러한 오류 해결은 구현 방법에 귀속되어 또 다른 상호운영성의 문제를 야기한다. 또한 구현의 어려움을 가중 시켜 벤더들의 구현 의욕을 저해하는 요소이기도 하여 표준 프로토콜로서의 적용에 문제점을 내포하고 있다.

SWAP[6]은 서로 다른 워크플로우 시스템과 그 워크플로우 시스템이 지원하는 응용프로그램들 사이의 상호운영성을 위해 IETF(Internet Engineering Task Force)에서 제안된 간단하고, 가벼우며, 확장 가능한 프로토콜이다.

SWAP은 XML과 HTTP와 같은 다른 프로토콜 표준화 노력과 연관되어 있다. HTTP는 프로그래머로 하여금 관계된 작업을 지원하기 위하여 다른 프로토콜 엔진을 작성하도록 하지 않으며, 프로토콜 확장 프로토콜(Protocol Extension Protocol)은 HTTP의 확장을 가능하게 하기 때문에 여러 목적에 대하여 유용한 점을 가지고 있다.

SWAP의 기본 동작은 프로세스 정의(Process Definition)의 URI로부터 시작한다. createProcess Instance라는 명령이 호출되면, ProcessInstance의 URI를 얻게 된다. 그 다음에 관찰자(Observer)의 URI를 ProcessInstance에 넘겨준다. 만약 ProcessInstance의 행위가 종료되면, 관찰자의 completed라는 명령을 호출한다. SWAP 명세에서는 이러한 동작을 관찰자/서비스 모델(Observer/Service Model)이라고 설명하며 그림 2는 이 모델을 도식화하여 설명한다[6].

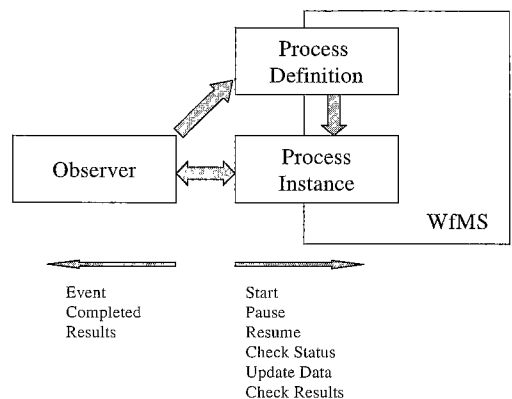


그림 2 SWAP의 관찰자/서비스 모델

SWAP 역시 인코딩 데이터 형식으로 XML을 이용하고 있고, MIME과 같은 프로토콜로서의 접근 방법을 택하고 있다. 그렇지만 MIME의 경우와 같이 비신뢰성 프

로토콜을 이용하지 않으며, 웹 기반의 시스템에서 이용하고 있는 HTTP를 이용한다는 점이 큰 장점이라 할 수 있다.

하지만, Wf-XML과 같이 데이터 교환에 필요한 요소들을 프로토콜에 정의하는 것으로만 한정시키고 있기 때문에 확장하기 어려운 문제가 있으며, 현재 논의 단계의 프로토콜이라 구현을 위한 정확한 명세를 제공하지 못한다.

### 3. 워크플로우 데이터 교환 프레임워크(WDEF :Workflow Data Exchange Framework)

#### 3.1 WDEF 구조

본 논문에서 제안하는 WDEF는 상이한 벤더들의 워크플로우 엔진간의 상호운영성을 제공하기 위한 프레임워크이다. WDEF에서는 워크플로우간 통신을 위한 메타데이터들을 데이터 레지스트리에 데이터 요소로서 정의하고, 정의된 데이터 요소들을 RDF 스키마로 자동 생성하여 워크플로우 시스템 사이의 통신 메시지를 구성한다.

그림 3은 WDEF 전체 구조를 보여준다. WDEF는 워크플로우 관리 시스템, 데이터 레지스트리, 그리고 워크플로우 데이터 교환 프로토콜(WDEF:Workflow Data Exchange Protocol)로 구성된다. WDEF는 워크플로우 엔진간의 데이터 교환을 위한 프로토콜이다.

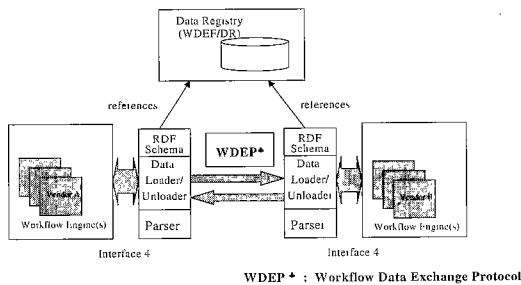


그림 3 WDEF 구조

WDEF는 다음과 같은 순서에 의해 동작한다.

① 데이터 요소 등록 : 데이터 레지스트리에 기존에 등록된 데이터 요소 이외에 필요한 요소가 상호운영성을 유지하는 데에 필요하다면 새로운 요소를 등록한다. 데이터 요소의 등록은 등록 위원회에서 담당하며, 여기에서는 WfMCA가 등록에 관한 권한을 가지고 있다고 가정한다.

② 데이터 레지스트리로부터 데이터 요소 추출 : RDF 스키마로 변환되어 워크플로우간 통신 메시지의 주형을 마련할 데이터 요소들을 검색하고 추출한다.

③ RDF 스키마를 이용한 어휘집(Vocabulary) 생성 : 추출된 데이터 요소들을 바탕으로 RDF 스키마를 이용하여 어휘집을 생성한다.

④ 통신 메시지 작성 : 교환될 데이터들을 RDF형식에 담아 워크플로우간의 통신 메시지를 구성한다.

⑤ 메시지 전송 : 구성된 통신 메시지를 WDEF를 통하여 통신하고자 하는 워크플로우 엔진에 전송한다.

⑥ 메시지 분석 : 전송된 RDF 문서를 분석하여 현재 워크플로우의 상태, 필요한 프로세스 정의 및 인터페이스 등과 같은 시스템에 필요한 정보들은 물론 실제 업무 데이터를 추출한다.

⑦ 워크플로우 서비스 : 정의된 업무 규칙에 의해 전송된 데이터를 이용하여 워크플로우 서비스를 실시한다.

WDEF는 워크플로우 사이에 주고받는 메시지 형식으로 RDF를 이용하기 때문에 다양한 네트워크상의 자원을 이용하는 전자상거래에 적용될 경우 많은 장점이 있다. 가령, 워크플로우의 프로세스에 필요한 데이터가 더블린 코어와 같은 RDF로 기술된 나름대로의 문맥을 가지고 있는 경우, 이러한 문맥 및 의미를 WDEF를 통해서 해석할 수 있으므로, 의미 모호성을 제거하며 인간 가독성(human-readability)과 기계 이해성(machine-understandability)을 제공한다.

#### 3.2 WDEF를 위한 데이터 레지스트리(WDEF/DR)

메타데이터는 데이터를 설명하는 데이터이다[7]. 워크플로우 시스템에서 메타데이터는 워크플로우 시스템 사이에 주고받는 메시지의 내용을 설명하는 데이터라고 볼 수 있다. 즉, 메시지를 주고받는 두 시스템 사이에 메타데이터를 이해할 수 있으면, 자동화된 데이터의 교환이 가능하다. ISO/IEC 11179[7]에서는 메타데이터를 데이터 요소라 명명하며, 그 표준화된 구조와 속성을 정의한다. 그림 4는 데이터 요소의 구조를 보여준다.

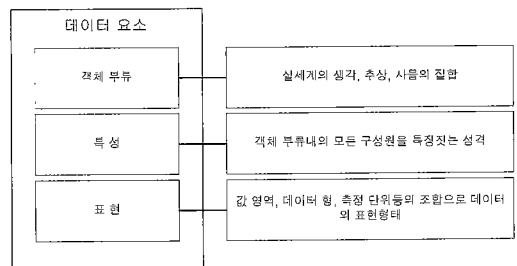


그림 4 데이터 요소의 구성요소

그림 4에서 보듯이 ISO/IEC 11179는 데이터 요소를 객체 부류(Object Class), 특성(Property), 표현(Representation)의 세 부분으로 나누어 정의한다. 객체 부류는 자동차, 사람, 고객, 주문, 사원 등과 같이 우리가 수집하거나 저장하려는 데이터를 지칭한다. 특성은 색, 모델, 성, 나이, 수입, 주소, 가격 등과 같이 사물을 구별하거나 설명하는데 사용되는 특징이다. 표현은 해당 데이터 요소가 가지는 값에 대한 도메인 및 표현 형태를 기술한다. 예를 들어, "사람"이라는 객체부류와 "나이"라는 특성을 합쳐 사람에 대한 나이를 지칭하며, 여기에 "0살부터 200살까지"라는 표현이 합쳐져 사람의 나이를 설명하는 하나의 데이터 요소가 된다.

데이터 요소를 등록, 관리, 서비스하는 데이터베이스를 데이터 레지스트리라고 하며, ANSI X3.285[10]에서 그 구조를 정의한다. 즉, 데이터 레지스트리는 데이터의 특성에 대한 사실(표준 데이터 요소)들을 저장하는 장소이다. 데이터 레지스트리는 시스템들 또는 조직 사이의 데이터 공유를 지원하며, 공유 데이터의 사용자로 하여금 데이터의 의미, 표현, 식별 등에 대한 일반적인 이해를 돕는다. ANSI X3.285에서는 데이터 레지스트리의 구조(메타 모델)를 정의하고 있으며, 이 ANSI X3.285를 기반으로 ISO/IEC 11179에서도 데이터 레지스트리의 기능 및 명세를 명시한다.

본 논문에서는 워크플로우 시스템에서 사용되는 메타 데이터를 공동으로 등록하고 활용하기 위해서 데이터 레지스트리를 이용한다.

그림 5는 워크플로우 엔진간의 통신을 위한 데이터의 종류를 설명한다. 워크플로우간 통신 데이터는 크게 두 가지이다. 첫 번째는 워크플로우간 프로세스 제어에 이용되는 데이터(Process Control Data)이며 다른 하나는 응용프로그램에서 이용되는 데이터(Application Data)이다. 전자의 경우는 비즈니스 프로세스 정의 및 인스턴스에 대한 내용과 더불어 인터페이스, 메소드, 결과값, 파라미터 등의 제어를 위한 데이터로 구성되어 있다. 응용프로그램에서 이용되는 데이터와 달리 제어 데이터들은 반드시 표준 수준의 메타데이터로 조직되어야 워크플로우

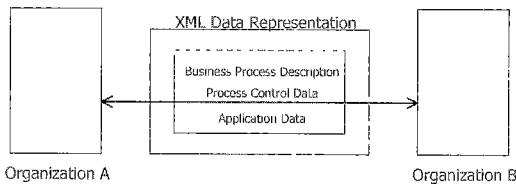


그림 5 워크플로우 시스템 사이의 통신 데이터

상호운영성을 제공할 수 있으므로, 본 논문에서 제시하는 워크플로우 데이터 교환을 위한 프레임워크에서 이용할 데이터 레지스트리는 제어 데이터만을 등록 대상으로 삼는다.

응용프로그램에서 이용되는 데이터는 필요에 따라 더블린 코어(Dublin Core), 또는 해당 분야별로 구성된 메타데이터 표준을 이용함으로써 서로간의 의미와 문맥을 전달할 수 있다.

### 3.3 RDF 스키마를 이용한 메시지 자동 구성

#### (1) 메시지 구조

워크플로우간의 통신 메시지의 구조는 그림 6과 같다. WDEF/DR은 통신 메시지의 각 요소들을 데이터 요소로서 등록해야 하며, 각 등록된 데이터 요소들은 효율적으로 RDF 스키마로 재구성 될 수 있도록 구조적으로 구성되어야 한다.

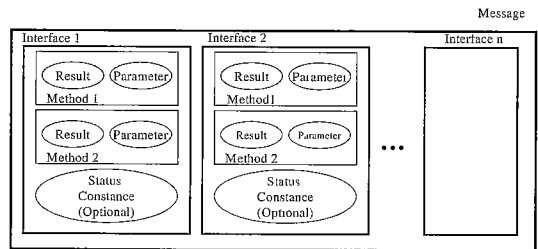


그림 6 워크플로우간 통신 메시지의 구성

메시지(Message)는 하나 이상의 인터페이스(Interface)로 구성되어 있으며 인터페이스는 수 개의 메소드(Method)와 상태 상수(Status Constance)들을 포함한다. 또한 각 메소드(Method)들은 결과값(Results)과 매개변수(Parameters)를 지정할 수 있다.

메시지는 워크플로우 통신의 기본 단위이다. 이러한 메시지를 구성하는 주요 요소는 인터페이스로 워크플로우 프로세스 인스턴스에 대한 접근 환경을 제공한다. 상태 상수는 프로세스 인스턴스의 상태를 기술하는 상수이며 메소드는 워크플로우 서비스를 위한 각종 행위를 직접 기술한다. 매개변수와 결과값은 메소드에서 이용되는 변수 및 인터페이스를 정의한다.

이러한 메시지의 구조는 WDEF/DR에 등록되는 각 데이터 요소들을 조직하는 기본 개념이 되며 RDF 스키마를 이용한 어휘집을 생성하는 데에 중요한 정보를 제공한다.

#### (2) RDF 데이터 모형

메타데이터가 데이터에 대한 데이터(data about data)를 의미한다면, RDF(Resource Description

Framework)는 웹 자원을 기술한 데이터(data describing web resource)로 정의할 수 있다[8]. RDF는 메타 데이터의 기술과 교환을 위한 구조로서, 웹 상의 메타 데이터를 지원하는데 필요한 구조를 정의하기 위해 W3C(World Wide Web Consortium)에서 제안한 표준이다. RDF는 상이한 메타데이터간의 어의, 구문 및 구조에 대한 공통적인 규칙을 지원하는 메커니즘을 통해 웹 상에 존재하는 기계이해형(machine-understandable) 정보를 교환하는 구조화된 메타데이터간의 상호운영성을 지원한다.

RDF 데이터 모형의 기술에 필요한 세 가지 요소는 자원(Resource), 특성 유형(Property Type), 특성 값(Value)이다. 자원은 일반적으로 자원의 형태에 관계없이 URI(Uniform Resource Identifier)로 식별 가능한 모든 객체를 의미한다. 하나의 자원은 여러 특성 유형이나 여러 특성 값을 가질 수 있다. 자원을 표현하는 속성은 특성 유형으로 구분된다. 특성 유형은 '저자' 또는 '서명' 등과 같이 자원의 속성을 적절한 이름으로 표현한 것이다. 인터넷상에서는 특성 유형 자체가 자원이 되는 경우가 있다. 따라서 특성 유형 역시 각각의 고유한 특성을 가지고 있다. 또한, 특성 유형은 그에 상응하는 값으로 표현되는데, 이를 특성 값이라고 하며, 문자열이나 숫자 등과 같이 자연어로 상세히 기술된다. 특성 값 역시 특성 유형처럼 자원이 될 수 있으며, 고유한 속성을 가지고 있다. 그림 7은 이들 각 요소들 간의 관계를 도식화한 것이다.

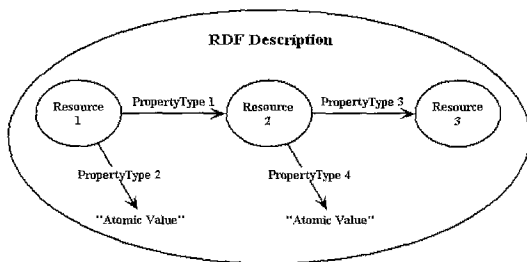


그림 7 RDF 데이터 모형의 각 요소간 관계

RDF 스키마는 특정 메타데이터에서 의미적 특성 유형의 집합이라 할 수 있는 어휘집을 선언하는데 사용된다. RDF 스키마는 특성 유형의 값의 특성이나 제한사항 뿐만 아니라 주어진 RDF 기술에서의 유용한 속성들을 정의하도록 한다. RDF 스키마를 식별하기 위해 XML(eXtensible Markup Language) 네임스페이스(namespace) 메커니즘이 사용되며, URI를 이용하여

RDF 스키마를 인간과 기계가 동시에 처리할 수 있도록 기술한다.

WDEF는 워크플로우 시스템 사이의 교환 메시지 형식으로 RDF를 사용한다. 그 이유는 다음과 같다.

- 데이터 요소와 RDF 데이터 모형의 유사성

데이터 요소는 객체 부류, 특성, 표현으로 구성되어 있다. 또한 RDF 데이터 모델은 자원, 특성 유형, 특성 값을 그 구성요소로 한다. 데이터 요소를 표현하는 객체 부류는 자원을 표현하는 개념과 유사하며, 특성은 특성 유형, 표현은 특성 값과 유사성을 가지고 있다.

- RDF 스키마의 데이터 요소 표현에 대한 적합성

RDF 스키마는 RDF 데이터 모형을 기술하는 어휘집을 만들기 위해 고안된 것이며, RDF 스키마는 데이터 요소를 설명하기 위한 항목들과 데이터 레지스트리 구축을 위한 메타모형을 구성할 수 있는 틀을 제공한다.

(3) RDF 스키마를 이용한 어휘집의 생성

RDF 스키마를 이용한 어휘집의 생성은 데이터 요소로부터 어휘집으로의 사상을 통해 이루어진다. 어휘집의 생성은 3단계를 필요로 하며 그림 8은 그 과정을 설명하고 있다.

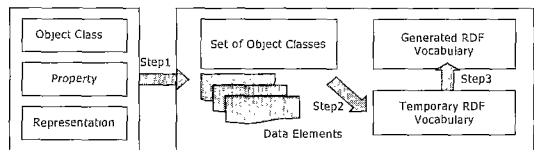


그림 8 데이터 레지스트리에서의 RDF 어휘집 생성 과정

RDF로 표현된 어휘집은 데이터 레지스트리의 다른 표현으로 이해할 수 있다. 데이터 레지스트리는 워크플로우 통신에 필요한 메타데이터를 데이터 요소로서 유기적으로 조직하고 있으며 이렇게 조직된 데이터 요소들은 RDF로 표현된 어휘집으로서 그대로 반영된다.

① 단계 1 : 데이터 요소의 검색

RDF 스키마는 정기적으로 재생성 될 수 있으며, 통신하고자 하는 워크플로우 엔진간에는 해당 데이터 요소의 버전만을 일치시키면 자동화된 데이터 교환이 가능하다. RDF 스키마는 데이터 레지스트리의 한 시점에서의 상태를 그대로 반영하고 있다. 각 워크플로우 엔진들은 일정 시점의 데이터 레지스트리로부터 통신을 위한 스키마를 구성하고 이를 통하여 통신 메시지를 구성하게 된다. 단계 1은 데이터 레지스트리에서 먼저 객체 클래스에 대한 탐색으로부터 시작하여, 필요한 데이터 요소들을 추출하는 과정을 포함한다.

② 단계 2 : 데이터 요소의 선택 및 재조직

데이터 레지스트리의 검색을 통하여 얻어진 데이터 요소들은 통신 메시지를 구성하기 위하여 재조직된다. 재조직의 기준은 데이터 레지스트리 안에 객체 클래스들 간의 관계로 표현되어 있다. 각 객체 클래스들 간의 관계를 바탕으로 RDF를 위한 어휘집을 생성하기 위한 속성 및 표현에 대한 정보 수집 작업이 단계 2에서 수행된다.

③ 단계 3 : RDF 스키마를 이용한 어휘집 생성

단계 2의 결과인 재조직된 데이터 요소들과 객체 클래스들 간의 관계 정보를 바탕으로 RDF 스키마를 이용하여 어휘집을 생성한다.

그림 9는 데이터 요소와 데이터 요소로부터 생성된 어휘집의 예를 보여준다. "Interface"는 "Message"를 구성하는 요소이므로 "Message"의 하위클래스의 관계에 있다. 이것은 RDF 스키마의 "subClassOf"를 이용하여 표현할 수 있다.

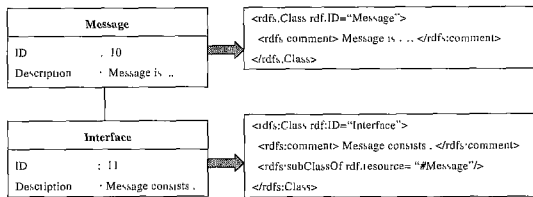


그림 9 데이터 요소로부터 어휘집 생성의 예

3.4 WDEF(Workflow Data Exchange Protocol)

WDEF는 SWAP을 기본으로 하여 RDF를 이해하여 워크플로우 엔진에 필요한 데이터의 전달을 지원하는 프로토콜이다. SWAP은 현재까지 개발 중인 프로토콜로서 명세를 확립해 나아가고 있다. 하지만 SWAP은 간단해야한다는 점을 주요 목적으로 삼고 있기 때문에, 현재까지의 SWAP의 범위가 확장될 것으로 전망되지 않는다. SWAP 이외의 추가되는 부분은 응용프로그램 수준에서 구현되어야 한다.

본 논문에서는 WDEF를 SWAP 위에 RDF를 이해하는 구문분석기(Parser)를 도입한 프로토콜로 정의한다. 만약 어떤 웹 기반의 워크플로우 시스템이 RDF로 기술된 응용프로그램 데이터를 이용하고자 할 때 SWAP의 확장 부분을 포함한 WDEF가 응용 프로그램 데이터를 이해하는 역할을 한다. 또한, 이러한 확장 구조는 WDEF가 RDF를 이용하여 응용프로그램 데이터를 기술하지 않는 시스템, 즉 SWAP만을 이용하는 시스템과

도 호환할 수 있는 구조를 제공한다.

4. 워크플로우 데이터 교환을 위한 데이터 레지스트리(WDEF/DR)의 설계

4장에서는 본 논문의 3장에서 설명한 WDEF의 핵심적인 부분을 구성하고 있는 데이터 레지스트리의 설계에 대해서 설명한다. 본 논문에서는 ISO/IEC 11179와 데이터 레지스트리에 대한 메타모델인 ANSI X3.285[10]에 기초한 웹 기반 워크플로우 데이터 교환 프레임워크를 위한 데이터 레지스트리(WDEF/DR)를 구현하였다.

4.1 레지스트리 메타 모델

ANSI X3.285[10]는 데이터 레지스트리를 위한 메타 모델이다. 이는 ANSI Accredited Standards Committee X3 Technical Committee L8에서 관리하는 표준으로 데이터 레지스트리의 구조(레지스트리의 스키마)를 나타내고 있으며, 개념적 데이터 모델을 소개하고 있다. 메타모델이란 메타데이터에 대한 데이터 모델을 의미한다. ANSI X3.285의 목적은 ISO/IEC 11179에 이해하기 쉬운 구조를 제공하는 것이며, ISO/IEC 11179의 각각의 부분들 간의 관계성들을 모델로서 표현하고 있다.

ANSI X3.285에서는 ISO/IEC 11179와 마찬가지로 데이터 레지스트리에 대한 가장 최소한의 모습들만 결정하고 있으며, 특정한 시스템 환경이나, DBMS, 데이터베이스 설계 패러다임, 시스템 개발 방법론 등은 명세하지 않는다. 단지 11179에서 제시하는 데이터 요소의 조직을 모델로서 표현하고자 하는 의도만이 담겨 있다. 따라서 이를 정확히 구현한 시스템은 존재하지 않으며, 각각의 시스템들은 이 모델을 기본으로 필요에 따라 확장을 할 수 있다. 그러나 핵심적인 부분은 삭제하지 못한다.

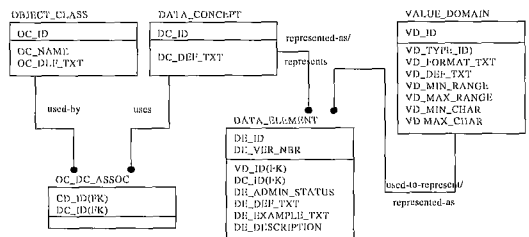


그림 10 WDEF/DR의 데이터 요소 관리 부분 메타모델

그림 10은 WDEF/DR의 메타모델 중 데이터 요소 관리 부분을 보여준다[10]. 데이터 요소는 'DATA\_



ELEMENT' 엔티티(Entity)에 의해 표현된다. 데이터 요소는 데이터 요소 식별자('DE\_ID')와 버전('DE\_VER\_NBR')을 기본키(Primary Key)로 갖는다. 데이터 요소들을 표현하기 위한 객체 클래스('OBJECT\_CLASS')와 데이터 개념('DATA\_CONCEPT')간의 연계는 'OC\_DC\_ASSOC' 엔티티에서 관리한다. 데이터 개념은 데이터 요소에 의해 표현되며, 데이터 요소의 표현은 값 도메인('VALUE\_DOMAIN') 엔티티가 수행한다.

4.2 워크플로우 데이터 요소

WfMC는 상호운영성 명세에서 워크플로우간의 메시지 구성을 위한 메타데이터들을 정의하고 있으며, 이러한 메타데이터들을 WDEF/DR을 위한 데이터 요소로서 추출할 수 있다. 각 데이터 요소들은 레지스트리 메타모델에 의해 조직된다.

표 1 메시지 데이터 요소

데이터 요소 이름	설 명
Message	RDF 어휘집의 클래스를 표현
Request	워크플로우 서비스 요청
Response	워크플로우 서비스 응답
SessionID	세션 번호
Exception	예외 상황

표 1은 메시지 클래스를 구성하는 데이터 요소를 나타낸다. "Message"라는 데이터 요소는 어휘집을 구성하기 위한 RDF 스키마의 Class를 설정하는데 이용되며, 그 아래에 보이는 요소들은 모두 이 클래스의 속성으로 파악할 수 있다. 이러한 점은 메시지, 인터페이스, 프로세스 상태, 메소드 등의 데이터 요소들에 모두 해당된다.

표 1에서 설명한 데이터 요소 이외에도 결과값(Results), 매개변수(Parameters)에 대한 데이터 요소들도 다수 존재한다. 이들 데이터 요소는 상호운영성 명세가 점차 확장되어감에 따라 데이터 요소에 대한 변경 및 증감이 발생하게 되며, WDEF는 이러한 표준의 변경 및 개선이 즉시 반영될 수 있는 기반 구조를 갖추고 있다.

5. 적용 예 : 전자상거래에서의 WDEF

WDEF는 웹 기반 워크플로우 관리시스템들 간의 상호운영성을 제공하여 서로 통신할 수 있도록 하는 프레임워크이다. 5장에서는 전자상거래의 예제를 통하여, WDEF가 어떻게 운영되며, 어떻게 보다 유연하고 확장성 있게 되는 지에 대해서 설명한다.

한국에 있는 어떤 회사 A는 컴퓨터에 관한 서적을 수입하는 업체라고 가정하자. 또 미국의 다른 어떤 회사 B는 인터넷을 통하여 프로그래밍 언어에 관한 서적을 판매하는 회사라 하자. 만약 A가 B로부터 "Java on the XML"이라는 "Knuth Ullman"이라는 저자의 책을 수입하고자 한다면, A의 워크플로우 엔진은 B의 엔진에 특정한 프로세스를 가동하도록 할 필요가 있다. 그림 11은 워크플로우 엔진 A와 B가 서로 통신하는 모습을 나타낸다.

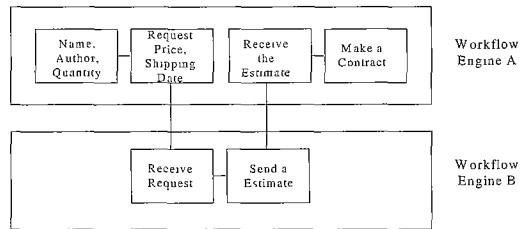


그림 11 전자상거래에서의 프로세스 예

워크플로우 엔진A는 사용자나 응용 프로그램에서 책명, 저자, 수량에 대한 정보를 얻게 된다. 이것을 이용하여 선적 일자와 가격에 대한 견적서를 워크플로우 엔진 B에 요구하게 되며, B는 사용자로부터 견적에 관한 사항을 입력받아 A에게 전달해줌으로써 계약에 대한 준비를 한다.

WDEF는 회사 A와 B는 자신의 워크플로우 시스템 상에서 상대 회사의 워크플로우 시스템의 프로세스를 실행시켜, 두 워크플로우 시스템이 상호연동 할 수 있는 환경을 제공한다. WDEF는 다음의 단계를 거쳐 동작하며, 각 단계는 3장에서 설명하였다. 이 장에서는 실제 예제를 통해 WDEF의 각 단계별 동작을 설명한다.

(1) 데이터 요소 등록(WDEF/DR)

워크플로우 시스템 사이에 주고받는 메시지의 내용을 설명하는 데이터를 데이터 요소(Data Element)라는 개념으로 등록한다. WfMC의 워크플로우 참조모델에서 정의하는 사용하는 데이터요소들과 표준에서 제시하지 않지만 각 워크플로우 시스템에서 정의하고 사용하는 프로세스들을 등록하면, 상호 연동 하고자 하는 다른 워크플로우 시스템에서 WDEF/DR를 참조하여 상대 워크플로우 시스템에서 제공하는 프로세스를 정확히 파악할 수 있다.

표 1의 데이터 요소 중 Request 데이터 요소의 내용은 표 2와 같다. 데이터 요소를 설명하는 21개의 속성들은 ISO/IEC 11179[7]에서 정의하고 있으며, WDEF/

DR은 이들 속성을 기본으로 하여 Request 데이터 요소에 대한 정확한 정의와 표현 범위, 문맥적 위치 등의 정보를 등록하고 서비스한다. 표 2는 WDEF/DR에 등록된 Request 데이터 요소에 대한 정보를 21개의 속성을 이용해서 표현한 것이며, 실제 WDEF/DR의 저장구조는 ANSI X3.285에서 정의하는 구조를 따라 설계, 구현된다.

표 2 WDEF/DR에 등록된 Request 데이터 요소

데이터 요소속성	속성 값	데이터 요소속성	속성 값
명칭	Request Message	표현범주	XML
식별자	wfmc.ref.0003	표현양식	Request
버전	1.0 - 2000	데이터 유형	RDF
등록 기관	Workflow Management Coalition	최대크기	10
동의명	Order	최소크기	1
문맥	Message	표현매치	-
정의	a message for requesting other company's information	이용 가능값	-
분류 체계	Workflow Message	책임조직	WfMC
핵심어	Message, Request	등록상태	Complete
관련 데이터	Response	발의조직	Company A
관련성 유형	the result of request		

(2) 데이터 요소 추출 및 RDF스키마를 이용한 어휘집(Vocabulary) 생성

WDEF/DR에는 공통적으로 사용하는 워크플로우 상호운영성에 대한 표준 및 확장 데이터 요소가 등록되어 있다. 워크플로우 엔진의 자동화된 검색과 이용을 위해서 추출된 데이터 요소를 RDF 어휘집으로 변환하는 과정이 필요하다. 그림 12는 WDEF/DR의 "Request", "processDefinition" 그리고 "ProcessInstance" 데이터 요소를 RDF 어휘집으로 생성한 결과를 보여준다.

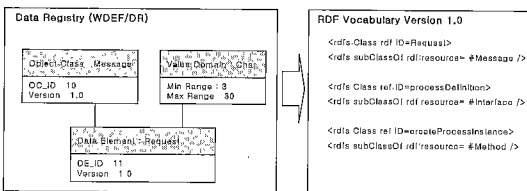


그림 12 WDEF/DR로부터 RDF 어휘집 생성

그림 12의 생성된 RDF 어휘집에서 Request는 메시지의 하위클래스 관계에 있다는 것을 RDF 스키마를 이용해서 표현한 것이며, "#Message"는 Message에 대한 설명이 문서의 나머지 부분에 있다는 것을 의미한다. Message에 대한 설명은 WDEF/DR에 객체 부류(Object Class)로 등록되어 있으며, 그 형식은 ISO/IEC 11179에서 정의한 표준을 따른다.

워크플로우 시스템은 생성된 RDF 어휘집을 사용하여, 상대 워크플로우 시스템에 요청할 메시지를 구성한다.

(3) Request 메시지 생성 및 전송

그림 13은 워크플로우 A가 B에게 책명과 저자에 대한 정보를 제공하는 부분에 대한 메시지를 생성한 결과를 보여준다. 접두어 'wdef'는 XML 네임스페이스(namespace) 메커니즘을 이용하여, WDEF에 의해 생성된 RDF 어휘집임을 표시한다. 'request', 'process Instance', 'context' 등은 WDEF/DR에 등록된 데이터 요소로부터 생성된 RDF 어휘이며, 메시지를 구성하는 프로세스 인스턴스 제어 데이터이다.

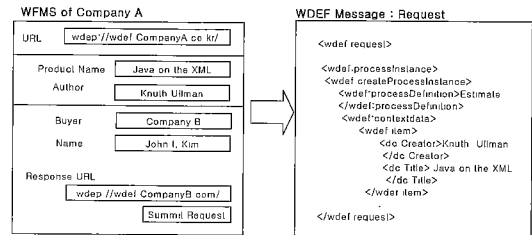


그림 13 Request 메시지 생성

접두어 'dc'는 더블린 코어임을 나타낸다. 'Creator'와 'Title'은 더블린 코어에서 정의하는 요소이며, 각각 지정한 자원의 저자와 제목을 표현하는데 이용되는 메타 데이터이다.

생성된 메시지는 WDEP(Workflow Data Exchange Protocol)을 이용해서 전송된다. WDEP는 SWAP을 확장하여 SWAP 명세에 RDF 문서를 이해하는 구분분석기(Parser)를 도입한 프로토콜로서 SWAP의 단순성으로 인한 메시지 표현의 한계를 데이터 레지스트리를 사용하여 확장성을 부여한다.

(4) 메시지 분석 및 워크플로우 서비스

그림 14는 수신된 메시지를 분석해서 처리하는 과정을 보여준다. Request 메시지를 받은 B회사의 워크플로우 시스템은 Request 메시지를 분석하여 견적을 산정하는 Estimate 프로세스를 실행해야 함을 인식하고 워크플로우 시스템의 프로세스 정의 저장소에서 Estimate라

는 프로세스 정의를 찾은 다음 실제 수행 가능한 프로세스 인스턴스를 생성한다. 그리고, Request 메시지에 포함되어 있는 데이터(책이름, 저자, 수량, 선적일 등)를 생성한 인스턴스에 넘겨준다.

워크플로우 시스템은 생성된 인스턴스를 실행시켜 데이터베이스에서 해당하는 책의 견적에 대한 정보를 찾아내 보여준 다음 선택된 결과를 Response 메시지로 구성하여 워크플로우 시스템 A에 전송한다. 결과 값을 전달하는 Response 메시지의 생성 역시 Request 메시지의 생성과정과 동일하며 상호운영성을 가지기 위해서 WDEF/DR에 등록된 데이터 요소를 기반으로 RDF 형식으로 메시지를 생성한다.

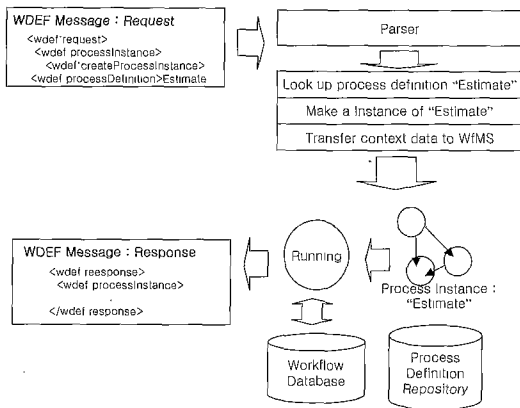


그림 14 메시지 분석 및 워크플로우 프로세스 실행

6. 비교 및 평가

본 논문에서는 워크플로우간의 상호운영성을 지원하기 위한 프레임워크를 제안하였고, 이것을 가능하도록 하는 기반 기술로서 ISO/IEC 11179 기반 데이터 레지스트리와 RDF를 도입하였다.

WfMC에서는 워크플로우간의 상호운영성을 제공하기 위하여, 다양한 노력들을 기울여 왔으며, 그 대표적인 예로 인터넷 전자우편 MIME 결합, Wf-XML 결합, SWAP 등을 들 수 있다. 이 절에서는 이들과의 비교 분석을 통하여 WDEF가 워크플로우 상호운영성 향상을 위해 기여하는 바를 설명한다.

(1) 인터넷 전자우편 MIME 결합과의 비교

WDEF는 SWAP을 그 기본 프로토콜로 사용한다. 따라서 프로토콜 측면에서 살펴보면, WDEF는 SWAP이 가지는 장점을 그대로 포함하고 있다. SWAP은 HTTP를 그 기본 프로토콜로 이용하고 있으므로, 전자

우편 프로토콜의 비 신뢰성으로 인한 추가적인 예외 상황 발생의 여지가 없고, 따라서 그에 대한 부가적인 처리를 필요로 하지 않는다.

또한 SWAP은 차세대 웹 문서로도 각광을 받고 있는 XML을 인코딩 데이터로 이용하기 때문에, 웹 기반의 응용프로그램에 적용되기도 적절하다. 워크플로우 트랜잭션은 그 성격상 긴 시간을 요하기도 한다. 이러한 트랜잭션을 지원하는 메커니즘을 MIME에서는 구현하기 어렵다. SWAP에서는 프로토콜 수준에서 이러한 기능을 제공한다. MIME이 가지고 있는 한계는 메시지 통신 시에 발생하는 지연시간을 예측할 수 없다는 점이다. 이점은 구현의 어려움을 가중시키기 때문에 워크플로우 구현자들의 환영을 받지 못하고 있다.

(2) Wf-XML과의 비교

Wf-XML은 XML을 이용하여 워크플로우간 통신 메시지를 구성한다는 점에서 WDEF와 동일하다. 하지만 Wf-XML은 표준화된 DTD(Data Type Definition)을 정의한 고정된 언어이며, WDEF에서 이용하는 RDF는 유동적인 표준의 반영이 가능하다는 점에서 차이점이 있다. 또한 WDEF는 XML 그 자체로도 통신을 할 수 있는 확장 가능한 구조로 설계되었기 때문에, Wf-XML과의 호환성을 유지할 수 있지만 Wf-XML은 그 자체의 언어만으로 통신을 해야하는 제약점을 안고 있다.

Wf-XML과 WDEF 모두 SWAP을 그 기본 프로토콜로 이용할 수 있다. 따라서 프로토콜 수준에서의 상호운영성을 모두 획득할 수는 있지만, 계속되는 표준의 변경 및 확장을 수용하는 측면에서는 MIME 결합의 경우와 마찬가지로 Wf-XML은 많은 비용을 요구한다.

WDEF의 확장성은 데이터 레지스트리를 이용한 메타데이터의 공유 환경 제공을 통해서 이루어진다. 인터넷상의 전자상거래는 불특정 다수간의 상거래를 지원해야 한다. 현재의 WDEF는 프로세스 정의를 공유하는 시스템들간의 통신만을 고려하지만, EDI 또는 CALS/EC에서 지원하듯 표준화된 메시지와 같이 프로세스 정의를 기반으로 확장해 나가는 환경의 변화를 수용할 수 있는 프레임워크이다.

7. 결론

본 논문에서는 웹 기반 워크플로우 데이터 교환을 위한 프레임워크를 제안하였다. 워크플로우 뿐만 아니라 현재의 모든 정보 시스템들은 상호운영성의 확보를 그 주요한 요구조건으로 삼고 있다. 웹 기반 워크플로우 데이터 교환 프레임워크(WDEF)는 데이터 레지스트리를 이용하여 워크플로우간 통신을 위한 메타데이터를 정의

하여 등록된 메타데이터들을 공유할 수 있는 환경을 구성하고, 워크플로우간 데이터 교환을 위한 메시지를 구성을 RDF를 이용함으로써 상호운영성을 확보한다. 또한 워크플로우간의 통신 프로토콜로서 SWAP을 기본으로 확장된 프로토콜을 이용함으로써 RDF를 사용하지 않는 시스템들간의 호환성을 유지하며, SWAP이 제공하는 워크플로우간 통신 메커니즘의 장점을 그대로 수용한다.

웹 기반 워크플로우 시스템들이 WDEF를 도입함으로써 얻을 수 있는 이점은 다음과 같이 정리할 수 있다. 첫째, RDF로 기술한 인터넷상의 다양한 자원에 대한 접근이 가능해짐에 따라 전자상거래 상에서 발생하는 의미적 호환성을 제거할 수 있다. 둘째, 응용프로그램 데이터에 기계 해독성 및 인간 해독성을 동시에 제공하여 비즈니스 환경을 보다 효율적으로 구성할 수 있도록 한다. 셋째, WDEF/DR을 참조할 수 있는 웹 기반 워크플로우 시스템들은 프로세스 정의를 공유할 수 있다면, 서로 통신할 수 있는 기본 환경을 제공한다. 넷째, 워크플로우 상호운영성 표준 관리, 변경 및 적용을 쉽게 한다.

현재 데이터 레지스트리의 구축과 RDF문서 처리기, XML 형식의 데이터 교환 메시지 생성 및 구문 분석기에 대한 구현을 진행하고 있다. 데이터 레지스트리의 구축을 위해서 ANSI X3.285의 레지스트리 메타모델을 기반으로 설계하였으며, 등록되는 데이터 요소는 WfMC에서 정의한 참조모델의 인터페이스4 부분을 수용할 계획이다. XML, RDF에 관한 부분은 상이한 워크플로우 시스템에서 적용가능해야 하므로 Java언어를 기반으로 구현중이다.

WDEF가 보다 본격적 의미의 전자상거래 환경을 제공하기 위해서는 불특정 다수간의 거래를 가능하게 해야 한다. 이를 위해서는 전자상거래의 분야에 맞는 문서 및 메타데이터 표준, 표준화된 프로세스 정의 등이 필요하다. 현재 EDI와 CALS/EC 등의 분야에서는 제공하는 메타데이터들에 대한 표준작업과 XML 적용 작업이 이루어지고 있다. 이들 작업과 더불어 RDF 스키마의 완성 및 SWAP의 완성과 더불어 WDEF를 적용한 시스템을 구축하는 것이 후후 연구 사항으로 남아있다.

## 참 고 문 헌

- [1] Workflow Management Coalition, "A WfMC White Paper : Workflow and Internet: Catalysis for Radical Change," Workflow Management Coalition, June 1998.
- [2] Workflow Management Coalition, "Workflow Standard - Interoperability Abstract Specification," Workflow Management Coalition, 20, October 1996.
- [3] Workflow Management Coalition, "Workflow Standard - Interoperability Internet e-mail MIME Binding," Workflow Management Coalition, September 1998.
- [4] Workflow Management Coalition, "Workflow Standard - Interoperability Wf\_XML Binding," Workflow Management Coalition, July 1998.
- [5] David Hollingsworth, "The Workflow Reference Model," Workflow Management Coalition, January 1995.
- [6] K. Swenson, "Simple Workflow Access Protocol (SWAP)," Strawman document, August 1998.
- [7] ISO/IEC FCD 11179 Part 1 :Specification and Standardization of Data Elements, 1994.
- [8] W3C, "Resource Description Framework(RDF) Model and Syntax Specification," W3C, February 1999.
- [9] W3C, "Resource Description Framework(RDF) Schema Specification," W3C, March 1999.
- [10] American National Standard for Information Technology, "Metamodel for the Management of Shareable Data," ANSI X3.285, Working Draft, 1998.
- [11] James G. Hayes, Effat Peyrovian, Sunil Sarin, Marc-Thomas Schmidt, Keith D. Swenson, Rainer Weber, "Workflow Interoperability Standards for the Internet," IEEE Internet Computing, Vol. 4, No. 3, pp.37-45, May/June 2000.
- [12] Workflow Management Coalition, "Workflow Standard - Interoperability Abstraction Specification," WfMC-TC-1012, Version 1.0, October 1996. available at <http://www.aiim.org/wfmc/standards/docs/if14-a.pdf>.
- [13] "Workflow Management Facility," Object Management Group Joint Submission, revised submission, July 1998. available at <ftp://ftp.omg.org/pub/docs/bom/98-06-07.pdf>.
- [14] M.-T. Schmidt, "Building Workflow Business Object," Business Object Design and Implementation II, D. Patel, J. Sutherland, and J. Miller, eds, Springer-Verlag, London, pp.64-76, 1998
- [15] Mike Anderson, Rob Allen, "Workflow Interoperability - Enabling E-Commerce," WfMC White Paper, April 1999.
- [16] 신화중, 조일제, 김기남, 지석진, 신동일, 신동규, "세종 워크플로우 관리 시스템의 설계 및 구현", 한국정보과학회 불 학술발표논문집, Vol. 26, No. 1, pp.644-646, 1999.
- [17] 한동수, 박향재, "비즈니스 프로세스의 자동화를 지원하는 웹기반 HiFlow시스템의 설계 및 구현", 정보과학회 논문지(C), 제4권, 1호, pp.50-62, 1998.
- [18] M. Zisman, "Representation, Specification, and Automation of Office Procedures," Ph.D. dis-

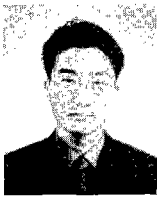
sertation, Wharton School, Univ. Pennsylvania, 1977.

- [19] C. Ellis, M. Bernal, "Officetalk-D: An Experimental office information system," Proceedings of the ACM-SIGOA Conference on Office Information Systems, pp.131-140, June 1982.
- [20] ORBWork: A Distributed CORBA-based Engine for the METEOR Workflow Management System, University of Georgia, Athens, GA, URL: <http://lsdis.cs.uga.edu/>.
- [21] A. Reuter, F. Schwenkreis, "ConTracts - A Low-Level Mechanism for Building General-Purpose Workflow Management Systems," Bulletin of the Technical Committee on Data Engineering, IEEE, Vol. 18, No.2, March 1995.
- [22] C. Ellis, C. Maltzahn, "Chautauqua: A Flexible Workflow System," Proc. of the 30th HICSS Conference, Jan., 1997.



백 두 권

1970년 ~ 1973년 고려대학교 수학과.  
 1974년 ~ 1976년 고려대학교 산업공학과(석사). 1981년 ~ 1983년 Wayne State Univ. 전산학 석사. 1984년 ~ 1985년 Wayne State Univ. 전산학 박사. 1986년 ~ 현재 고려대학교 컴퓨터학과 교수. 1989년 ~ 현재 한국정보과학회 평의원/이사. 1991년 ~ 현재 ISO/IEC JTC1/SC32 국내위원회 위원장. 관심분야는 소프트웨어 공학, 데이터 공학, 메타데이터, 컴포넌트 시스템, 데이터베이스



나 홍 석

1990년 ~ 1993년 고려대학교 컴퓨터학과. 1994년 ~ 1995년 고려대학교 컴퓨터학과(석사). 1996년 ~ 1998년 고려대학교 컴퓨터학과(박사수료). 1996년 ~ 현재 고려대학교 기초과학연구소 연구원. 2000년 ~ 현재 (주)라임미디어테크놀러지스 대표. 관심분야는 메타데이터, 데이터레지스트리, 워크플로우, XML 문서 유통 시스템



김 중 일

1992년 ~ 1997년 고려대학교 원예학과. 1998년 ~ 1999년 고려대학교 컴퓨터학과(석사). 1998년 ~ 1999년 고려대학교 컴퓨터과학기술연구소 연구원. 2000년 ~ 현재 (주)라임미디어테크놀러지스 기술이사. 관심분야는 메타데이터, 워크플로우, 원격교육시스템



이 이 섭

1984년 ~ 1998년 서강대학교 이공대학 수학과. 1998년 ~ 1999년 서강대학교 전자계산학과(석사). 1990년 ~ 1992년 삼성종합기술원 CAE 센터. 1993년 ~ 1995년 삼성SDS 정보기술연구소. 1996년 ~ 1999년 삼성경영기술대학 정보통신학과 조교수. 1999년 ~ 현재 삼성 SDS 지식관리시스템, 워크플로우 시스템 개발 총괄. 관심분야는 워크플로우, 지식관리시스템, 원격교육시스템