

워크플로우 서비스 트랜잭션 처리 메커니즘 설계

안형진^o 박민재 김광훈

경기대학교 전자계산학과 워크플로우 기술 연구실
{ctrl_workflow^o, mjpark, kwang}@kyonggi.ac.kr

Design of Integrated Framework for Workflow Communication

Hyungjin Ahn^o Meanjae Park Kwanghoon Kim
Dept. Computer Science, Kyonggi University

요 약

현재의 비즈니스 환경은 조직의 점진적인 거대화, 조직들 간에 비즈니스 프로세스 상호운용을 통해 보다 광범위하고 복잡한 비즈니스 서비스들에 대한 처리를 요구하고 있다. 이것은 곧 워크플로우 프로세스를 워크플로우 도메인의 하나의 서비스 개체로서 인지하고, 워크플로우 프로세스와 관련되는 다양한 리소스들의 상호연계를 통해 질적으로 향상된 워크플로우 서비스를 제공하는 것이 가능해져야 함을 의미한다. 본 논문에서는 지속적으로 진화하고 있는 워크플로우 비즈니스 도메인에서 발생하는 단순 및 복합 워크플로우 서비스를 처리하는 서비스 지향 워크플로우 관리 시스템 구현의 기반 개념이 될 워크플로우 서비스 트랜잭션 처리 메커니즘에 대해 기술하고자 한다.

1. 서 론

비즈니스 정보 기술의 급속한 발전과 시장의 세계화는 비즈니스 환경에서 조직 내외부 간에 발생하는 업무들을 보다 신속하고 유연하게 처리하는 것을 요구하게 되었고, 이를 위해 업무 프로세스에 대한 전자적인 수행을 필요로 하게 되었다. 이러한 요구 사항을 반영한 개념이 워크플로우이다. 최근의 비즈니스 환경은 인터넷과 같은 네트워크 인프라의 급속한 확산과 워크플로우를 사용하는 조직의 점진적인 거대화, 그리고 조직들 간에 비즈니스 프로세스 상호운용을 통한 업무의 상호 연동 및 정보의 상호 교환등으로 인해 보다 광범위하고 복잡한 비즈니스 서비스들에 대한 처리를 요구하고 있다. 이것은 곧 워크플로우 프로세스를 조직이나 기업에서 일어나는 일련의 업무 처리들에 대한 절차를 정의하는 비즈니스 템플릿의 개념을 넘어서서, 워크플로우 도메인의 하나의 서비스 개체로서 인지하고, 워크플로우 프로세스와 관련되는 다양한 리소스들의 상호 연계를 통해 QoS(Quality of Service)가 향상된 워크플로우 서비스를 제공하는 것이 가능해져야 함을 의미한다. 현재의 워크플로우 시장은 이러한 비즈니스 요구 사항을 만족시키기 위하여 엔터프라이즈 시장을 이끌어나가는 기술들인 EAI, 웹 서비스, BPM 등 각종 기술들을 접목한 워크플로우 제품군을 내놓고 있지만, 이는 각 벤더들이 가지는 기반 기술력을 통해 독자적인 서비스를 구축해나가는 것일 뿐, 표준 프레임워크를 기반으로 제작되는 제품군들이 아니기 때문에 실질적인 워크플로우 비즈니스 서비스들을 통합적으로 상호 운용하는 것에는 한계가 있다.

본 논문에서는 단위 업무들의 순차 또는 병렬적 절차들의 나열로서 정의되던 워크플로우 프로세스를 더욱 다양하며 질적으로 향상된 비즈니스 서비스를 제공하는 서비스 개체로서 워크플로우 프로세스를 바라보고 있다. 이와 같이, 유연하고 광범위

한 서비스의 연계를 통해 이루어지는 워크플로우 서비스들을 효율적으로 처리하기 위한 메커니즘으로서, 서비스 지향 아키텍처에서 제시하는 개념들을 현재 가장 잘 반영하고 있는 “웹 서비스 아키텍처” 개념을 바탕으로 워크플로우 서비스 트랜잭션에 대한 처리 메커니즘을 기술하고자 한다.

2. 관련 연구

2.1. 워크플로우 표준 커뮤니케이션 인터페이스

2.1.1. WAPI 와 Wf-XML 2.0

WAPI (Workflow Application Programming Interface)는 워크플로우 엔진과 워크플로우 애플리케이션들 간의 상호 통신을 위해 사용되는 프로그래밍 인터페이스이며, 워크플로우 표준 기구인 WfMC에서는 인터페이스 2&3를 통해 WAPI 오피레이션들을 정의하고 있다[4]. Wf-XML 메시지 프로토콜은 프로세스 엔진 상에서 호출되어 사용되는 비동기적인 서비스들을 사용할 수 있도록 지원하는 OASIS의 비동기 웹 서비스에 대한 표준인 ASAP(Asynchronous Service Access Protocol)의 확장된 개념으로서의 워크플로우 표준 인터페이스 4의 프로토콜로 정의되어 있으며, 현재 2.0버전까지 개발된 상태이다[2]. Wf-XML은 워크플로우 엔진 간 또는 수행 서비스들 간의 상호운용성을 지원한다. 현재 Wf-XML 2.0은 지속적으로 표준을 위한 협의가 진행되고 있는 상태이나, 빠르게 변화하는 비즈니스 환경에 대처하기에는 현저히 미흡한 상태라 할 수 있다.

2.2. SOA와 웹 서비스

SOA (Service-Oriented Architecture)는 소프트웨어 아키텍처로 특정 기술의 집합이 아니며 기술로부터 독립적인 개념을 나타내는 아키텍처이다. 비즈니스 환경에서 SOA의 순수한 정

의는 “ 호출 가능한 잘 정의된 인터페이스를 갖는 독립된 기능의 서비스로 정의한 애플리케이션 아키텍처”이다. 반면, 웹 서비스는 기술의 집합체이며 SOA의 개념을 구체화한 것이다. 하지만 웹 서비스는 단순한 SOA의 구현만은 아니다. 웹 서비스는 SOA 구현의 Best Practice라 할 수 있다. 웹 서비스는 HTTP, XML, SOAP, WSDL, UDDI 등의 표준을 기본 구조로 이용하고 있으며 이들 기술들은 특정 플랫폼에 대하여 독립적이며, 표준화된 기반을 가지고 있으며, 또한 상호운용성을 충분히 보장하고 있는 기술 아키텍처라 할 수 있다.

3. 워크플로우 서비스 트랜잭션 처리 메커니즘 설계

3.1. 워크플로우 서비스

워크플로우 도메인에서 발생하는 단순 또는 복합 형태의 워크플로우 구성 리소스들의 연동을 기본으로 하여 워크플로우 관련 리소스들이 제공하는 기능들을 "워크플로우 서비스"라는 개념으로 접근한다.

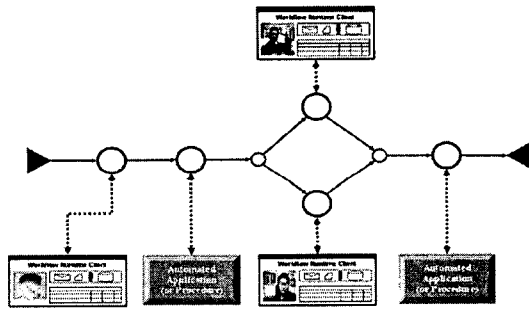


그림 1 단순 형태 워크플로우 서비스

위의 그림 1에서의 “ 단순 형태 워크플로우 서비스”는 워크플로우 프로세스 내에 정의된 워크플로우 액티비티마다 연계된 단위 애플리케이션 및 여러 리소스들의 연동을 통해 서비스를 제공하는 서비스를 말한다.

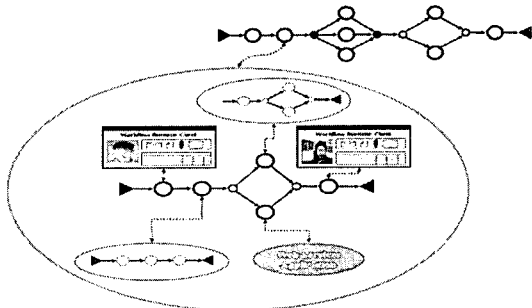


그림 2 복합 형태 워크플로우 서비스

그림 2에서 보여주고 있는 “ 복합 형태 워크플로우 서비스”는 워크플로우 액티비티 뿐만 아니라 워크플로우 프로세스를 서비스 개체로 인식하고, 컴포넌트 서비스 및 여러 워크플로우 인젝트먼트 서비스들의 광범위한 서비스 결합을 통해 보다 복잡하고 다양한 질적 향상 서비스를 제공해주는 것이 가능하다.

그림 4에서와 같이, 파트너들 간에 중단점을 알고 있을 경우에

3.2. 워크플로우 서비스 메타-모델

아래의 그림 3은 워크플로우 도메인 상에서 발생하는 각종 서비스 개체들의 상관 관계를 나타내는 서비스 지향 워크플로우 메타-모델을 나타내고 있다.

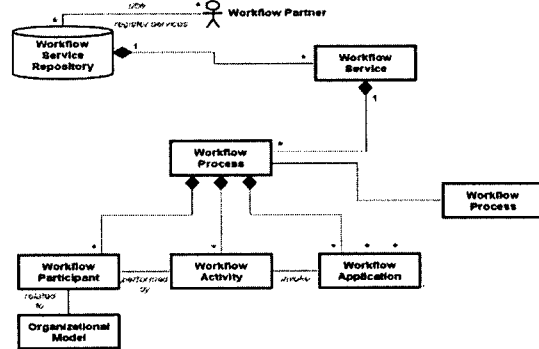


그림 3 워크플로우 서비스 메타-모델

워크플로우 액티비티는 워크플로우 엔진에 의해 자동적으로 처리되는 자동 액티비티, 사용자에게 작업인 워크아이템으로서 할당되어 처리되는 수동 액티비티, 사설(private) 또는 공개(public) 워크플로우 프로세스를 호출하여 업무를 처리하는 액티비티등의 여러 액티비티 타입이 존재한다. 액티비티와 가장 밀접하게 관련되는 속성 개체들이 워크플로우 참여자와 워크플로우 애플리케이션이다. 워크플로우 참여자는 액티비티의 수행자로서 비즈니스에 참여하게 되며, 이 때 수행자 자신에게 할당된 업무인 워크아이템을 처리하는데 필요한 것이 워크플로우 애플리케이션이다. 워크플로우 애플리케이션은 런타임 클라이언트, 모니터링 클라이언트, 기타 애플리케이션 등 여러 형태의 애플리케이션들을 통해 실제적인 서비스의 수행을 구체화한다.

3.3. 워크플로우 서비스 트랜잭션 처리 메커니즘

워크플로우 도메인에서의 서비스 상호 연동은 워크플로우 파트너들 간의 서비스 바인딩에 의해 이루어진다. 서비스 지향 워크플로우 환경에서 발생하는 서비스 바인딩의 경우로서 하나는 워크플로우 서비스 파트너들 간 중단점에 대한 URL을 알고 있는 경우, 다른 하나는 파트너들 간에 서비스 중단점 URL을 알지 못하는 경우들이 있다.

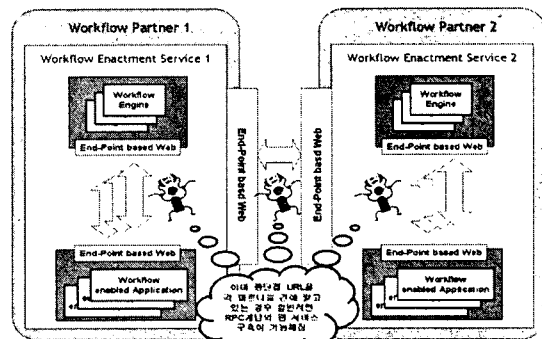


그림 4 서비스 중단점을 알고 있는 워크플로우 파트너들 간의 서비스 연계

는 일반적인 RPC 형태의 SOAP 바인딩을 통한 요청 및 응답을 통해 통신하거나 한 파트너가 다른 파트너의 종단점에 직접 연결하여 WSDL 파일을 다운로드하여 해당 서비스 목록에 기술된 워크플로우 서비스들을 호출하는 방법들이 존재한다.

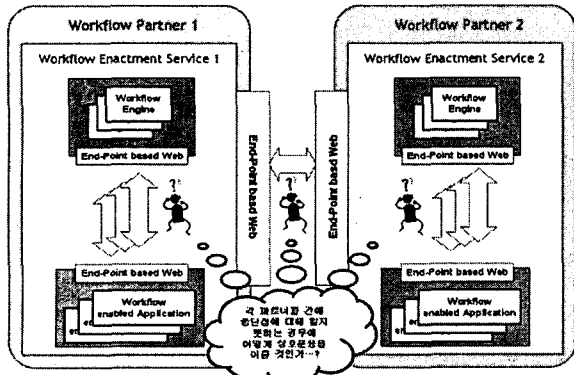


그림 5 서비스 종단점을 알지 못하는 워크플로우 파트너들 간의 서비스 연계

위의 그림 5에서는 워크플로우 파트너들 간에 사용되는 사실 또는 공용 레지스트리를 통해 파트너에 대한 종단점 URL을 얻어내고, 해당 URL을 이용하여 파트너들 간의 바인딩을 이루게 되며, 서비스 기술 명세인 WSDL을 다운로드 받아 원하는 서비스를 호출하는 것을 보여주고 있다. 위에서 설명한 두 가지 경우의 공통점은 서비스를 요청하는 파트너 측이 상호간의 바인딩을 통해 워크플로우 서비스들에 대해 기술한 명세인 WSDL을 다운로드 받아 서비스를 호출하는 시점부터 워크플로우 관련 서비스의 본격적인 요청 및 응답이 이루어진다는 것이다. 워크플로우 서비스 교환에서 사용되는 SOAP 메시지는 Wf-XML 2.0의 룰을 바탕으로 만들어진다. Wf-XML 2.0을 이용한 서비스 요청 및 응답에 대한 실제적인 비동기 처리는 "메시지 큐"를 통해 구체화하게 된다. 다음의 그림 6은 메시지 큐잉 메커니즘을 통해 서비스 요청과 응답에 대해 처리하는 것에 대해 보여주고 있다.

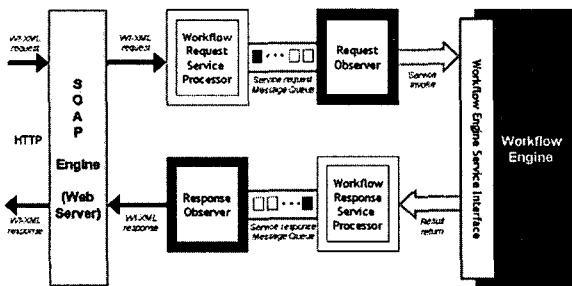


그림 6 워크플로우 서비스 트랜잭션 처리 메커니즘 개념도

위의 그림에서 보는 바와 같이, 종단점에 위치한 "워크플로우 요청 서비스 프로세서"는 워크플로우 서비스 제공자 측으로부터

터 전달된 Wf-XML 기반의 요청 메시지를 해석하여 적절한 서비스 메소드 실행을 위한 메시지를 요청 메시지 큐에 적재하게 된다. 메시지는 큐의 이벤트를 감지하고 있는 옵저버 컴포넌트에 전달되어 워크플로우 엔진의 서비스 메소드 수행을 통해 실질적인 작업을 처리하게 된다. 수행 완료된 결과 데이터는 "워크플로우 응답 서비스 프로세서"에 의해 Wf-XML 기반의 응답 메시지로 구성되어 응답 메시지 큐에 적재된 후 옵저버에게 전송하여 서비스 요청 측으로 최종 전달하게 된다. 만약 Wf-XML 메시지를 받는 파트너 측에서 예외 상황이 발생할 경우, 실패된 메시지는 삭제되지 않고 메시지 큐에 남아 재전송 메커니즘을 통해 다시 보내어지게 된다.

4. 결론 및 향후 발전 방향

기존의 워크플로우 환경에서의 이슈가 되는 부분은 "워크플로우 프로세스"였다. 워크플로우 프로세스는 비즈니스 환경에서 일어나는 일련의 업무들에 대한 절차를 정의한 것으로서, 워크플로우 인에이블먼트 서비스가 구성되는 비즈니스 도메인의 업무 자동화 및 사용자와의 업무 인터페이스를 능동적 형태로 제공해주는 개념을 제공하였다. 그러나 점차적으로 최근의 워크플로우 환경은 프로세스를 구성하는 액티비티에 연계된 애플리케이션 또는 리소스들의 순차적 또는 병렬적 연동을 통해 이루어지는 형태의 워크플로우 뿐만 아니라, 조직과 기업 간의 원활한 상호운용을 지원하며 서비스들의 유연한 통합 및 확장을 통한 형태의 워크플로우 서비스를 제공하는 것을 원하고 있다. 본 논문에서는 워크플로우 비즈니스 도메인에서 발생하는 서비스들을 단순 형태 워크플로우 서비스 및 복합 형태 워크플로우 서비스로 분류하여 정의된 서비스 개념들을 바탕으로 워크플로우 서비스 메타-모델을 정의하여 워크플로우 서비스들을 형성하는 개체들 간의 관계에 대해 기술하였으며, 이러한 개념들을 토대로 현재 비즈니스 환경에서 이슈가 되고 있는 서비스 지향 아키텍처 개념과 웹 서비스 기술 아키텍처를 접목하여 워크플로우 서비스 환경에서 일어나는 서비스 트랜잭션에 대한 처리 메커니즘 설계 내용을 이론적으로 제시하였다. 향후에는 본 논문에서 제안한 워크플로우 서비스 트랜잭션 메커니즘을 기반으로 하여 서비스 지향 워크플로우 관리 시스템을 구축하기 위한 시스템 아키텍처를 설계하고 아키텍처를 바탕으로 구현하고자 한다.

참고 문헌

- [1] Keith Swenson, "Asynchronous Service Access Protocol (ASAP)", Organization for the Advancement of Structured Information Standards (OASIS), June 2004
- [2] Kieth Swenson, Mike Gilger and Sameer Predhan, "Wf-XML 2.0, XML Based Protocol for Run-Time Integration of Process Engines", Workflow Management Coalition (WfMC), October 2003
- [3] Keith Swenson, "ASAP/Wf-XML 2.0 Cookbook", Fujitsu Software Corporation, June 2004
- [4] The Workflow Management Coalition, "Reference Model", WfMC-TC-1003, November 1994
- [5] 이경하, 이규철 "SOA(Service-Oriented Architecture)와 웹 서비스", 정보과학회지 제22권 10호, 2004.10