

효율적인 프로세스 실행을 위한 웹서비스 기반의 실행 아키텍처 Web Service-based Enactment Architecture for the Efficient Process Execution

윤호성, 정지수, 최인준
포항공과대학교 산업공학과

Abstract

There have been numerous research and development efforts to automate business processes or execute business processes automatically. Especially, Web service is considered to be one of the most promising technologies. Existing Web service composition languages such as BPEL (Business Process Execution Language) and BPML (Business Process Modeling Language) do not provide full semantic support for business processes. This paper proposes a 3-layer model that consists of business process layer and service layer which can seamlessly integrates business process semantics with service semantics. The proposed model and architecture will enable effective management of business processes as well-defined corporate assets, services that execute tasks to achieve goals of business processes, and software components that comprise services.

1. 서론

정보기술의 발전은 기업의 경영환경에 많은 변화를 가져왔다. 고객들의 제품이나 서비스에 대한 요구 사항들은 급격히 증가하고 있는 반면, 상대적으로 제품의 수명주기는 짧아지고 있는 추세이다. 기업들은 새로운 경영환경에 대응하고 이윤을 극대화하기 위하여 제품과 서비스를 빠르게 제공하고자 노력하고있으며, 그 일환으로 조직 내부의 프로세스를 개선하거나 기업간 M&A 및 전략적 제휴를 추구하는 등 다양한 해결방안을 모색하고 있다[Will92]. 이와 같은 경영 전략의 변화와 더불어, 최근 들어 비즈니스 프로세스 관리의 중요성이 기업의 경쟁 우위를 유지하기 위한 수단으로서 널리 인지되고 있다. 조직 내부의 비즈니스 프로세스를 유연하게 변화시키고 혁신시키는 동시에, 다른 기업의 프로세스와 손쉽게 통합할 수 있게 하는 방법에 대한 기업들의 관심이 증가하고 있다.

그러나, 비즈니스 프로세스를 손쉽게 변화시키고 혁신시키는 등 유연하게 관리하는 것은 쉽지않다. 프로세스 설계에서 분석까지 프로세스 모델을 완성하는 과정에 많은 시간과 노력이 필요로 될 뿐만 아니라, 완성된 프로세스 모델을 수행하기 위하여 요구되는 시스템을 기업 내부에 구축하기까지 추가적인 노력과 시간과 자본이 필요로 되기 때문이다.

[WfMC00]에서는 이러한 문제를 극복하고, 비즈니스 프로세스를 손쉽게 설계하고 실현할 수 있게 하기 위한 수단으로 워크플로우(workflow) 개념을 제안하고 있다. 워크플로우는 비즈니스 프로세스를 정형화하고 수행을 자동화하고자 하는 개념으로, 워크플로우 관련 표준화 기관인 WfMC에서는 프로세스 정의와 실행 아키텍처에 대한 표준들을 제공하고 있다.

그러나, 현재 워크플로우 개념은 비즈니스 프로세스를 구성하는 단위 업무들이 수행되는 과정을 정의하거나 지원하는 것을 고려하지 않고 있다. 특히 사람의 참여가 필요하지 않고 기업 내부의 데이터 베이스나 기존 시스템(legacy system) 등에 의하여 수행되는, 어플리케이션 레벨의 복잡한 실행 로직을 갖는 단위 업무들을 지원하는 데는 미흡하다. 이러한 단위 업무에 대한 자세한 설계는 비즈니스 프로세스를 특정 어플리케이션에 종속 시키는 문제점을 유발하므로 고려대상에서 제외되어 왔기 때문이다. 그러나, 단위 업무의 실행 로직을 정의하고, 실행을 위한 어플리케이션을 효율적으로 빠르게 조합하는 것은 비즈니스 프로세스의 설계 및 실행의 정확성과 속도에 큰 영향을 미치게 된다. 새로 정의된 비즈니스 프로세스의 각 단위업무를 수행하는 어플리케이션을 개발하기 위하여 소요되는 시간과 비용을 기존 시스템을 활용하여 절약 할 수 있기 때문이다.

본 연구에서는 이와 같은 사항에 주목하고 비즈니스 프로세스를 빠르고 유연하게 관리하기 위한 하나의 방안으로서 어플리케이션 기반의 단위 업무 수행 과정을 정의하고 지원할 수 있는 비즈니스 프로세스 실행 아키텍처를 제시하고자 한다. 이를 위하여 본 연구는 최근 주목 받고 있는 서비스 기반 아키텍처(Service Oriented Architecture)와 웹 서비스(Web Service) 개념을 도입한다. 본 연구에서 제시하는 비즈니스 프로세스 실행 3 레이어 모델은 비즈니스 프로세스가 정의되는 비즈니스 공간과 기업의 어플리케이션들이 위치하는 컴퓨팅 공간 사이에 새로운 서비스 공간을 추가함으로써 비즈니스 프로세스와 어플리케이션의 효율적인 통합을 실현할 수 있게 한다. 3 레이어 모델은 비즈니스 프로세스의 독립성이 유지하는 동시에, 어플리

케이션 기반의 단위 업무들이 기존 시스템을 활용하여 정의되고 실행되도록 지원 한다.

이후 구성은 다음과 같다. 2 장에서는 관련연구를 소개하고, 3 장에서는 3 레이어 모델을 설명한 후, 4 장에서는 이 3 레이어 모델을 실현할 수 있는 구체적인 프로세스 실행 아키텍처에 대하여 설명한다. 5 장에서 프로토타입 시스템에 대해 간략히 살펴본 후, 6 장에서 결론을 정리한다.

2. 관련연구

웹서비스는 e-Business 표준을 따르면서 인터넷을 통해 제공되는, 비즈니스 로직을 수행하는 소프트웨어 컴포넌트라고 정의할 수 있다. 웹서비스는 기술이나 컴퓨팅이 아닌 서비스로서 이용된다는 점에서 기존의 어플리케이션이나 소프트웨어 컴포넌트 개념과 구별된다. BPEL4WS와 BPML은 웹 서비스 기반의 프로세스 모델링 언어로서 비즈니스 프로세스를 XML 형태로 표현하여 서비스 개발과 프로세스 실행의 편의를 제공한다 [Sander01]. 한편, 비즈니스 프로세스 관리에 웹서비스 개념을 도입하기 위한 시도는 다음과 같은 연구들을 통해 진행 중이다.

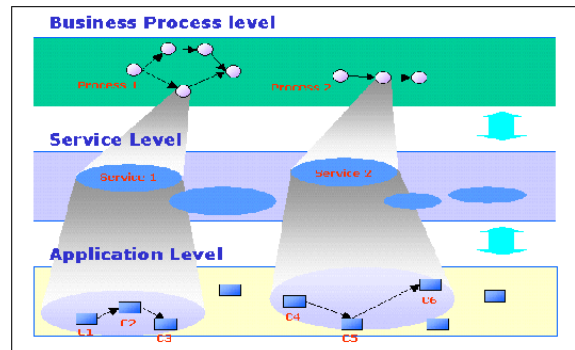
- 통합에 대한 연구 : 웹서비스들을 하나의 프로세스로 통합하고자 하는 연구. 웹서비스 Orchestration 을 정의하기 위한 프로세스 실행 정의 언어인 BPEL4WS와 BPML 이외에도 Message Choreography (WSC1, BPEL), Message Transaction (WS-C, WS-T, BTP), Agreement (CPA, CSWS) 등에 대한 연구가 진행되고 있다.
- 웹서비스 구성에 대한 연구 : 웹서비스들을 구성하여 하나의 프로세스를 생성하는 방법에 대한 연구. 필요에 따라 UDDI 에서 웹서비스를 손쉽게 검색, 선택하여 교환할 수 있는 컴포넌트 기반의 비즈니스 워크플로우 관리에 대한 연구가 있다 [YD03].
- 웹서비스 프로세스의 유효성에 대한 연구 : 웹서비스들의 복잡한 연결들로 이루어지는 프로세스에 대한 신뢰성과 타당성 검증 방안에 대한 연구. 신뢰성 있는 웹서비스 구성을 만들기 위한 접근 방법들을 분류하고 이에 대한 방법론을 제시한 연구 [Muh02] 와 기존의 소프트웨어 모델 검증 기술을 이용하여 유효성을 검증하는 연구 [Shin02] 등이 이루어지고 있다.

그러나, 이러한 연구들은 웹서비스 자체를 구현하는 방법론이거나 기존의 웹서비스들을

연결하여 비즈니스 프로세스를 구성하고자 하는 연구들이며, 비즈니스 프로세스와 어플리케이션들을 효율적으로 통합하는 수단으로 웹서비스를 활용하고 있지는 않다.

3. 3-레이어 모델

3 레이어 모델은 비즈니스 프로세스 레이어와 어플리케이션 레이어 사이에 서비스 레이어를 추가함으로써 둘 사이의 괴리를 극복하고 효율적인 비즈니스 프로세스 실행을 가능하게 한다. [그림 1]은 3 레이어 모델을 도식화한 것이다.



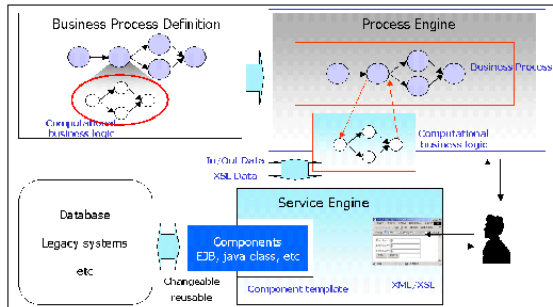
[그림 1] 3-레이어 모델

비즈니스 프로세스 레이어에서는 비즈니스 골(Goal)을 가진 비즈니스 프로세스 모델이 설계된다. 비즈니스 프로세스는 단위 업무와 그들간의 선후 관계로 정의되며, 기존의 워크플로우 프로세스 모델과 동일한 정보들을 갖는다. 어플리케이션 레이어에는 기업 내부의 기존 시스템들의 기능들이 활용 가능한 컴포넌트 형태로 존재한다. 컴포넌트 기반의 개발 개념은 시스템의 재사용성을 증가시킬 뿐만 아니라 빠르고 유연한 어플리케이션 구축에 매우 중요하다. 한편 서비스 레이어는 두 레이어 사이의 브로커 역할을 수행 한다. 하나의 서비스는 여러 컴포넌트들의 논리적 결합에 의하여 생성되며, 생성된 서비스들은 또한 여러 프로세스들에 의해 활용될 수 있다.

4. 프로세스 실행 아키텍처

본 논문에서 제시하는 프로세스 실행 아키텍처는 프로세스 실행 엔진과 서비스 엔진 그리고 컴포넌트 풀로 구성된다. 특히 어플리케이션 기반의 단위업무의 수행을 지원하는 서비스 엔진을 제시하고 이 엔진을 웹서비스로 공개함으로써 임의의 프로세스 엔진에서 서비스 엔진을 통하여 단위 업무를 처리할 수 있게 한다. 어플리케이션 기반 단위 업무는

기존의 프로세스 정의언어로는 표현할 수 없었던 컴포넌트들의 실행 정보로 정의된다. 서비스 엔진은 기업의 필요한 컴포넌트들을 실행시킬 뿐만 아니라, XML 과 XSL 정보로 이루어진 화면정보를 통해 필요한 사용자와의 상호작용을 제공하기도 한다. [그림 2]는 전체적인 프로세스 실행 아키텍처를 나타낸다.



[그림 2] 비즈니스 프로세스 실행 아키텍처

4.1. 비즈니스 프로세스 정의

본 논문에서는 단위 업무 수준의 실행 방안을 정의할 수 있도록 기존의 프로세스 실행 정의언어를 확장한다. 확장된 프로세스 실행정의언어는 'PROCESS'요소를 통해 비즈니스 프로세스를 정의하며, 어플리케이션 기반의 단위업무에 대해서는 'COMPONENT_FLOW'를 정의한다. 단, 필요한 컴포넌트들은 미리 구현되어 있어야 한다. 프로세스 실행정의언어는 웹서비스 파트너 정보, 파트너에게 보내지는 메시지 정보, 파트너와의 인터페이스를 정의하는 메세징 정보를 포함할 수 있도록 확장된다. 기존의 프로세스 실행정의언어가 프로세스 레벨에서 실행되었던 언어라고 한다면 확장된 프로세스 실행정의언어는 프로세스 레벨과 서비스 레벨 그리고 어플리케이션 레벨 모두에서 실행되는 언어라고 할 수 있다.

4.2. 프로세스 엔진

프로세스 엔진은 프로세스 정의 언어로 정의된 프로세스 모델을 실행시키는 시스템이다. 확장된 프로세스 실행 정의언어를 실행시키기 위해서는 단위 업무의 수행 과정을 메시지에 담고 웹서비스를 통해 서비스엔진으로 보내는 추가적인 기능이 필요하다. 다음은 기존의 프로세스 엔진에 확장되어야 하는 기능들이다.

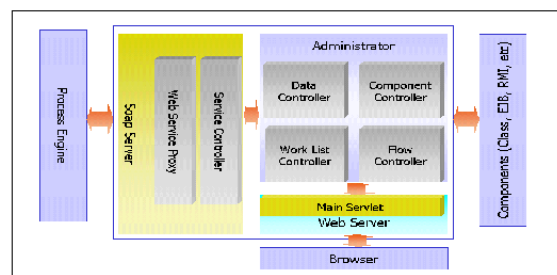
- 서비스 관리 기능 : WSDL에 명시된 웹 서비스 파트너의 정보를 이용하여 웹서비스를 실행한다. 웹서비스 실행에 필요한 메시지와 Proxy 정보가 필요하다.
- 컴포넌트 관리 기능 : 컴포넌트 플로우

인스턴스를 생성 관리 할 수 있어야 한다.

- 업무 관리 기능 : 어플리케이션 기반 단위 업무에 대해서 필요한 메시지를 만들고 서비스 관리 기능을 통해 메시지 교환을 실행하는 기능이 추가 되어야 한다.

4.3. 서비스 엔진

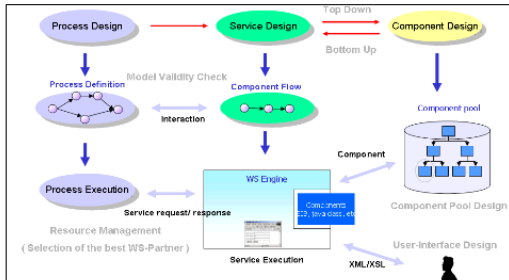
서비스 엔진은 프로세스 실행 엔진으로부터 어플리케이션 기반 단위 업무의 수행 정보를 받아 이를 실행 시키는 시스템이다. 서비스 엔진의 아키텍처는 인터넷 프로토콜, JAVA Servlet 등의 인터넷 기술과 SOAP, WSDL 등의 웹서비스 기술, RMI 나 EJB 등의 분산 컴퓨팅 기술들을 바탕으로 제시되었다. 엔진의 내부는 [그림 3]와 같이 크게 세 부분으로 구성된다. SOAP서버를 통해 프로세스 엔진과 상호 작용하는 부분, 컴포넌트들과 연결되어 실제로 서비스를 실행하는 부분, 그리고 웹서버를 기반으로 작업자와 대화하는 부분이다.



[그림 3] 서비스 엔진의 아키텍처

4.4. 비즈니스 프로세스 설계 및 실행 방법론
이 장에서는 제시된 아키텍처를 기반으로 실제 비즈니스 프로세스의 설계와 실행이 이루어지는 방안에 대하여 논의한다 ([그림4] 참조). 비즈니스 프로세스 정의 단계에서는 비즈니스 프로세스가 설계되며, 어플리케이션 기반의 단위 업무에 대해서는 각각에 요구되는 서비스가 설계된다. 서비스 제공에 필요한 컴포넌트들을 설계하는 과정에서는 서비스의 필요에 따라 컴포넌트를 새롭게 설계하는 Top Down 방식과 이미 설계된 컴포넌트들을 재활용하는 Bottom Up 방식이 이용된다. 설계된 컴포넌트들은 미리 구현되어 컴포넌트 풀에 저장되며, 컴포넌트 풀은 저장과 검색이 용이하도록 설계되어야 한다. 이렇게 비즈니스 프로세스와 서비스의 설계가 끝나면 그 산출물로서 각각 비즈니스 프로세스 정의와 컴포넌트 플로우가 생성된다. 이들은 분석단계에서 유효성이 검증되어야 하며 비즈니스 프로세스 정의에는 필요한 곳에 적절히 컴포넌트 플로우가 참조 되어야 한다. 실행단계에서 프로세스 정의는 프로세스

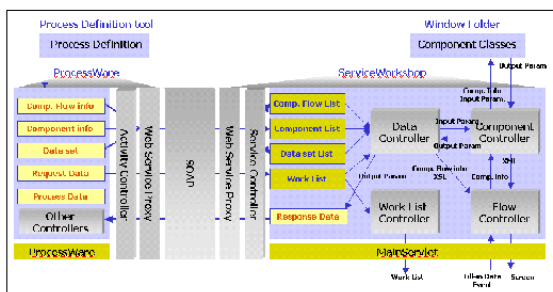
엔진에 의해 실행되며 참조된 컴포넌트 플로우를 웹서비스를 통하여 서비스 엔진에서 실행된다. 서비스 엔진에서는 컴포넌트 플로우 정보를 받아, 정의된 컴포넌트를 실행시키거나 사용자와의 상호작용을 통하여 결과를 산출하고 이를 프로세스에 반환한다.



[그림 4] 프로세스 설계 및 실행 방법론

5. 프로토타입 시스템

프로세스 정의 도구에 의해 생성된 비즈니스 프로세스 모델은 프로세스 웨어에서 실행된다. 비즈니스 프로세스 모델에 정의된 컴포넌트 플로우 정보, 컴포넌트 정보, 데이터 셋 정보는 프로세스가 수행되는 도중에 Activity Controller와 SOAP Proxy에 의해 서비스 워크숍에 전달된다. 서비스 워크숍은 이 정보를 리스트에 등록시키고 Data Controller를 이용하여 관리한다. Component Controller는 Data Controller나 Main Servlet을 통해 작업 수행자에게 정보를 입력받고 컴포넌트를 실행시킨다.

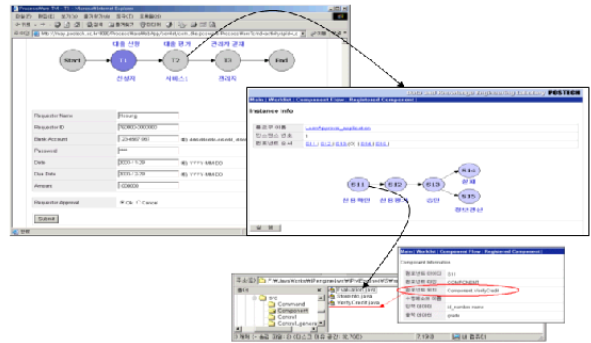


[그림 5] 프로토타입 시스템 구성도

미리 정의된 컴포넌트들은 Component Controller에 의해 사용 된다. Flow Controller는 Data Controller로부터 컴포넌트 플로우 정보를 받아 정해진 순서대로 컴포넌트 정보를 Component Controller에 제공 하며, Data Controller로부터는 XSL 정보를 Component Controller로부터는 XML정보를 받아 서블릿 환경을 통해 수행자에게 화면 정보를 제공한다. 작업수행자는 Work List Controller 로부터 작업목록을 얻고, 필요한 작업을 선택하여 실행한다.

[그림6]은 프로토타입 시스템이 수행

되는 과정에 나타나는 주요 화면들이다. 예제 비즈니스 프로세스의 단위 업무 T2는 어플리케이션 기반 단위 업무이므로, 프로세스 엔진은 T2의 컴포넌트 플로우 정보를 서비스 엔진으로 넘겨 수행하게 된다. 각 컴포넌트는 아래 그림과 같이 기업내부의 적절한 위치에 미리 정의되어 있다.



[그림 6] 프로토타입 시스템

6. 결론

본 연구에서는 웹서비스와 컴포넌트 기반 개발 기술을 기반으로, 비즈니스 프로세스 실행을 위한 3-레이어 모델을 제안하였다. 이 모델은 비즈니스 프로세스 레이어와 컴포넌트 형태의 어플리케이션 레이어 사이에 웹서비스 형태의 서비스 레이어를 추가함으로써 두 레이어를 효율적으로 통합시킬 수 있게 한다. 서비스 레이어는 비즈니스 프로세스 수행에 필요한 어플리케이션 기능을 효과적으로 조합할 수 있게 하여 어플리케이션에 독립된 비즈니스 프로세스 설계를 가능하게 하고, 어플리케이션 재사용성을 증가시킴으로써 개발에 소요되는 시간과 비용을 감소시킨다.

참고문헌

- [Muh02] Muhammad F. Kaleem, "A Classification Framework for Approaches and Methodologies to make Web Services Compositions Reliable", Technical University Hamburg-Harburg, 2002.
- [Sander01] Sander Duivestijn, "Web Services and Workflow", <http://www.webservicesarchitect.com>, Web Services Architect, September, 2001.
- [Shin02] Shin NAKAJIMA Hosei University and PRESTO, JST, "Verification of Web Service Flows with Model-Checking Techniques", Proceedings of the First International Symposium on Cyber Worlds (CW'02), IEEE Computer Society, 2002.
- [Will92] William Davidow, "The Virtual Corporation", HarperCollins Publishers, 1992.
- [YD03] Yu-Liang Chi and et. al, "Component-based business workflow management for web services collaboration", IEEE Computer Society, 2003.