

# MDA를 적용한 웹서비스 개발 프로세스

윤 흥 란<sup>†</sup> · 박 재 년<sup>††</sup>

## 요 약

최근 XML표준기술을 활용한 웹서비스는 기업내 및 기업간의 정보시스템을 통합하는데 발생하는 문제점을 해결해 주는 차세대 e-비즈니스의 기반으로 급부상하고 있다. 정보기술이 지속적으로 변화하기 때문에 특정 기술에 기반한 시스템을 정보 기술 변화에 맞게 통합, 변화, 유지하는 것은 지속적인 문제로 고려되고 있다. 이러한 문제점을 해결하기 위해 OMG는 MDA(Model Driven Architecture)라는 새로운 소프트웨어 아키텍처를 제시하였다. MDA는 기존의 개발 절차에서 분석모델인 플랫폼 독립적인 모델(PIM, Platform Independent Model)을 구축하고 이를 기반으로 설계모델인 플랫폼 종속적인 모델(PSM, Platform Specific Model)로 자동 변환하는 프로세스를 가진다. 이러한 자동 변환을 통해 여러 플랫폼을 쉽게 지원하고 개발자의 입장에서는 코드 작성에 소요되는 시간을 줄일 수 있으며 개발 프로세스의 측면에서도 품질관리를 수월하게 할 수 있다.

이 MDA의 개발 프로세스를 웹서비스개발 프로세스에 적용하면 MDA의 PIM에서 목표플랫폼으로 웹서비스를 선택하고 웹서비스 모델인 WSDL로 PSM을 표현할 수 있다.

본 논문에서는 웹서비스 공급자와 요청자가 웹서비스를 개발하거나 통합할때의 절차를 분류하여 웹서비스 개발 프로세스유형을 만들고 웹서비스 개발시 MDA개발 프로세스를 적용하여 웹서비스공급자와 요청자가 참조가능한 새로운 웹서비스 개발 프로세스를 제안한다.

키워드 : 웹서비스, 개발프로세스, MDA, WSDL, PIM, PSM

## A Web Service Development Process with MDA Applied

Yun Hong-ran<sup>†</sup> · Park Jae-nyun<sup>††</sup>

### ABSTRACT

Being able to resolve huge problems deriving from integration of information systems in-house or business to business, the web service that uses the XML standard technology has recently taken a quick dominance the next generation e-business bases. It's one constant concern how to integrate, change, and maintain such systems as based on certain technologies according to the changes to information technology, which is on the ongoing process of evolution. To help solve those problems, OMG suggested a new software architecture called MDA(Model Driven Architecture). MDA runs a process that establishes a platform independent model(PIM), which is an analysis model used as part of the existing development procedures, and automatically converts it into a platform specific model(PSM), a design model, based on the established PIM. Such automatic conversion has lots of benefits including easy support for diverse platforms, reducing the coding time that usually consume a great deal of the developer's effort, and facilitating quality control in the aspect of development processes.

By applying the MDA development process to a new web service development, you can choose web service as the target platform at the PIM of MDA and express PSM with a web service model, WSDL.

This study set out to classify the web service development or integration processes by the provider and requester to identify the types of web service development processes, and to apply the MDA development process to web service development, thus suggesting a new kind of web service development process that can be referred to by both the web service provider and requester.

Key Words : Web Service, Development Process, MDA, WSDL, PIM, PSM

### 1. 서 론

최근 XML표준기술을 활용한 웹서비스는 기업내 및 기업

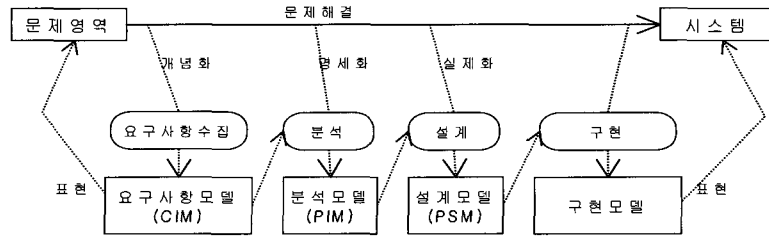
간의 정보시스템을 통합하는데 발생하는 문제점을 해결해 주는 차세대 e-비즈니스의 기반으로 급부상하고 있다. IT시스템은 과거 메인프레임이 모든 정보를 중앙에서 집중하여 처리하는 호스트 처리방식에서 90년대에 클라이언트-서버방식으로 발전되어 왔다. 이러한 IT시스템의 대부분은 시스템 자체의 유연성이나 확장성이 적어 고립된 구조의 하드웨어 및 소프트웨어 아키텍처로 이루어져 있다. 1990년대 후반에

※ 본 연구는 한국전산원 위탁연구과제(2005-정평-B08) 지원으로 수행되었음.

† 정 회 원 : 숙명여자대학교 정보과학부 컴퓨터과학과 박사과정

†† 종신회원 : 숙명여자대학교 정보과학부 교수

논문접수 : 2005년 3월 23일, 심사완료 : 2005년 6월 20일



(그림 1) MDA의 개발 프로세스

들어되면서 많은 기업들이 기업의 생산성 향상 및 고객 서비스를 위해 IT투자를 확대해 감에 따라 기업 내에서도 다수의 이질적인 시스템 도입을 추진하게 되었고 그 결과 많은 기업 내에는 호환성이 없는 시스템이 산재해 업무추진에 어려움을 주고 있으며, 이러한 복잡하고 분산화된 시스템을 통합하고자 업계에서는 많은 노력을 기울이게 되었다. 그 결과로 등장한 기술이 웹서비스이며 이것은 분산화된 시스템을 통합하는 기존의 통합기술들의 문제점을 해결하고 분산된 애플리케이션을 저비용·고효율로 통합하기 위한 대안이 되었다[2]. 정보기술이 지속적으로 변화하기 때문에 특정 기술에 기반한 시스템을 정보 기술 변화에 맞게 통합, 변화, 유지하는 것은 지속적인 문제로 고려되고 있다. 이러한 문제점을 해결하기 위해 OMG는 MDA(Model Driven Architecture)라는 새로운 소프트웨어 아키텍처를 제시하였다. MDA는 시스템의 설계(design)와 명세(specification)를 정형화된 모델로서 기술 플랫폼과 분리하고 실제 구현과 관련된 모델은 변환규칙에 따른 매핑을 통해서 얻는 개발 방법이다. 이 방법은 기존의 개발 절차에서 분석모델인 플랫폼 독립적인 모델(PIM, Platform Independent Model)을 구축하고 이를 기반으로 설계모델인 플랫폼 종속적인 모델(PSM, Platform Specific Model)로 자동 변환하는 프로세스를 가진다. 이러한 자동 변환을 통해 여러 플랫폼을 쉽게 지원하고 개발자의 입장에서는 코드 작성에 소요되는 시간을 줄일 수 있으며 개발 프로세스의 측면에서도 품질관리를 수월하게 할 수 있다[6, 7].

이 MDA의 개발 프로세스를 웹서비스개발 프로세스에 적용하면 MDA의 PIM에서 목표플랫폼으로 웹서비스를 선택하고 웹서비스 모델인 WSDL로 PSM을 표현할 수 있다.

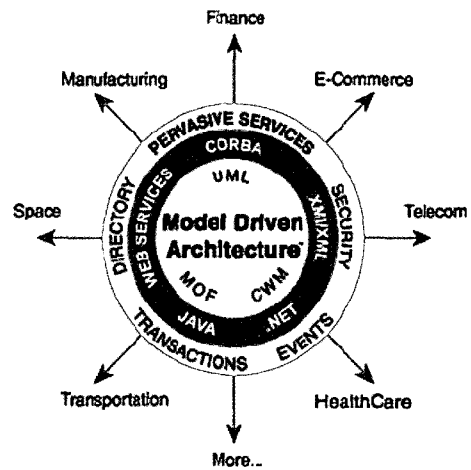
본 논문에서는 웹서비스 공급자와 요청자가 웹서비스를 개발하거나 통합할때의 절차를 분류하여 웹서비스 개발 프로세스유형을 만들고 웹서비스 개발시 MDA개발 프로세스를 적용하여 웹서비스공급자와 요청자가 참조가능한 새로운 웹서비스 개발 프로세스를 제안한다.

## 2. 연구 배경

### 2.1 MDA(Model Driven Architecture)

#### 2.1.1 MDA개요

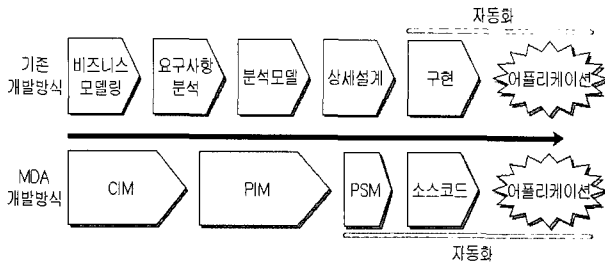
MDA는 OMG(Object Management Group)가 그동안 만들어낸 플랫폼 기술과 표준모델링언어(UML)등을 이용하여



(그림 2) OMG의 MDA 로고

구현된 여러 산업의 표준안을 결합한 모델 방식의 새로운 소프트웨어 아키텍처이다[6, 7]. MDA는 (그림 1)에서 보는 바와 같이 기존의 개발 절차에서 요구사항모델인 CIM(Computation Independent Model)을 구축한다. 이어서 분석모델인 플랫폼 독립적인 PIM(Platform Independent Model)을 구축하고 이를 기반으로 설계모델인 플랫폼 종속적인 PSM(Platform Specific Model)으로 변환하고 최종 애플리케이션을 구현하는 프로세스를 갖는다. (그림 2)는 MDA가 다양한 방향으로 적용 가능함을 보여주는 OMG의 MDA로고이다.

MDA는 MDA표준을 적용해서 모델의 자동화와 변환(transformation)을 통해 여러 플랫폼을 쉽게 지원하고 개발자의 입장에서는 코드 작성에 소요되는 시간을 줄일 수 있으며 개발 프로세스의 측면에서도 품질관리를 수월하게 할 수 있다. MDA에서는 PIM과 PSM을 분리했기 때문에 PIM을 변경하지 않고도 기술 플랫폼의 변화나 요구사항 변화에 발 빠르게 대처할 수 있다. 플랫폼 의존적인 시스템은 기존의 아키텍처가 새로운 요구사항을 만족시키지 못하는 경우가 많이 발생한다. MDA를 이용하면 기능과 아키텍처를 분리해서 정의하기 때문에 아키텍처의 변화가 있더라도 변환과정을 거쳐 구현과정으로 쉽게 진행할 수 있다. 따라서 시스템의 기능과 요구사항 변화에 의한 아키텍처 변경에 비교적 자유롭다. 한번 작성된 PIM의 경우 비즈니스 핵심 부분에 대한 모델이기 때문에 향후 다른 시스템에서도 쉽게 이용될 수 있으며 재사용성이 높아지게 된다. MDA의 PIM모델을 활용하여 재사용이 가능한 서비스컴포넌트를 식별할 수 있다.



(그림 3) 기존 개발방식과 MDA개발방식

2.1.2 기존 개발방식과 MDA개발방식

MDA는 B2B 애플리케이션같은 소프트웨어시스템을 구현하기 위하여 웹서비스에 적용할 수 있다. MDA의 경우 기존의 개발방식과 아래 (그림 3)과 같은 차이가 있으며 MDA는 기존 개발방식에 비해 많은 부분을 자동화에 의존한다.

기존 개발방식대표적인 객체지향 방법론인 UP(Unified Process)에서는 비즈니스 모델링을 통해 전체 비즈니스를 파악하고 유스케이스 다이어그램을 통해 요구사항을 모델링하여 이것으로 분석모델을 만든다. 이때의 분석모델은 비기능적 요구사항이 전혀 반영되지 않은 순수한 비즈니스를 모델링하고 있으며, 이 분석 모델을 기반으로 적절한 플랫폼과 설계 메커니즘을 적용해 플랫폼 종속적인 설계 모델을 만들고 이를 기반으로 구현한다. 요구사항 모델은 시스템을 개발자 관점으로 가져오기 전에 고객과 의사소통하기 위한 수단이고 초기분석 모델에서는 어떤 환경에서 어떤 제약조건을 갖고 구현될지는 생각하지 않고 순전히 비즈니스만을 반영한 모델을 모델링한다. 그리고 설계모델에서는 비기능적 요구사항을 반영해 세부 설계를 하고 이 모델을 이용해 구현 소스 코드를 만든다.

기존의 소프트웨어 개발 방법은 분석, 설계 작업을 통해 나온 모델을 개발자가 직접 구현하는데 반해 MDA는 개발작업의 중심이 플랫폼 독립적인 모델인 PIM을 구축하는 것에 맞추어져 있고 이후에는 MDA툴을 이용해 변환 작업을 하여 최종 소스코드, 애플리케이션을 만들어내는 것이다. MDA중심에는 PIM이 있고 툴과 변환작업이 애플리케이션을 만들어내는 데 중요한 역할을 한다. 따라서 MDA에서는

PIM을 PSM으로 자동으로 변환하도록 하는 변환규칙이 매우 중요하다.

2.2 웹서비스

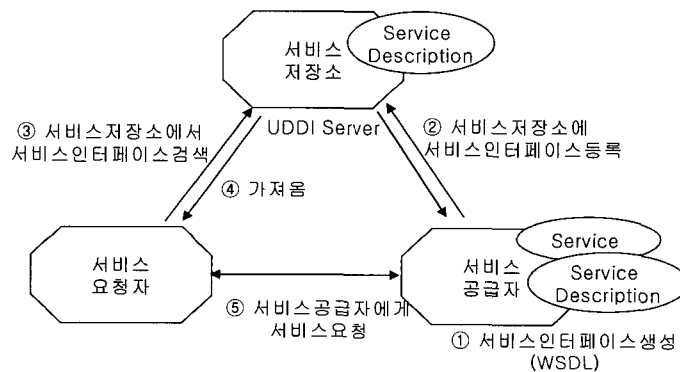
2.2.1 웹서비스 개요

웹 서비스는 SOAP(Simple Object Access Protocol), WSDL(Web Services Description Language), UDDI(Universal Description Discovery and Integration)라는 XML 기반의 세 가지 공개 표준을 이용하여, XML 메시지를 웹을 통하여 전송함으로써 이기종 시스템간의 상호 작용을 돕는 소프트웨어 시스템을 일컫는다[1]. SOAP은 웹 서비스 제공자와 웹 서비스 요청자가 분산 환경에서 어떤 방식으로 통신을 해야 하는가에 대해 주고받는 메시지의 형식을 정의한 메시지 교환 프로토콜이다. WSDL은 해당 웹 서비스에 대한 상세한 설명을 포함하고 있는 서비스 기술 언어로, 개발자가 웹서비스의 기능이나 인터페이스 정의 등 웹 서비스를 설명할 때 사용한다. 그리고 UDDI는 웹 서비스를 등록하고, 검색할 수 있는 일종의 디렉터리 서비스와 같은 웹서비스 레지스트리에 대한 명세이다[4].

2.2.2 웹서비스 아키텍처

웹서비스는 서비스지향아키텍처라는 큰 그림의 일부이며, 플랫폼 독립(platform independent), 느슨한 연결(loosely coupled), 서비스 기반검색 같은 기술상의 이점을 제공한다. 웹서비스는 인터페이스의 정형화 및 표준화를 통해 공급자와 소비자 사이의 분리를 가능하게 한다[2]. 웹서비스는 서비스지향아키텍처를 구현하는데 사용되어질 수 있는 유일한 기술은 아니며, 서비스지향아키텍처를 웹서비스 기술이 아닌 다른 기술을 이용하여 구현할 수 있다. 웹서비스 또한 단순히 서비스지향아키텍처를 구현한 것만은 아니다. 웹서비스는 서비스지향아키텍처의 기본적인 요구사항을 만족하는 가장 좋은 예가 될 수 있다.

(그림 4)에서 보면 서비스 공급자는 서비스를 개발하여 WSDL을 서비스저장소에 등록을 하게 되고 서비스 요청자는 서비스 저장소에서 WSDL을 검색하여 사용할 서비스를 선택하고 WSDL에서 제시한 정보를 가지고 SOAP을 통해



(그림 4) 웹서비스 아키텍처

서비스공급자가 제공하는 서비스를 사용할 수 있다.

웹서비스는 서비스요청자가 필요할 때 원격호출 가능한 명세를 WSDL로 제공하며, 효율적인 분산/협업시스템구축을 위한 효율적 방안을 제시하게 된다. 웹서비스는 서로 다른 시스템 간 유연한 상호연계가 가능하며, 서로 다른 각각의 서비스 모듈들을 통합하여 새로운 웹서비스로 만들 수 있다는 특징이 있다. 웹서비스의 장점을 보면 조직이 기존에 가지고 있는 전산자원을 재활용(reuse)하여 지속적으로 가치를 창출할 수 있도록 도와주며, 시스템 주변 환경변화에 대한 기존시스템의 변화를 최소화할 수 있도록 돕는다. 웹서비스는 급격한 시장변화에 맞닥뜨림없이 비즈니스의 요구사항에 빠르게 대처할 수 있도록 함으로써 기업경쟁력강화에 크게 기여할 수 있다.

### 3. MDA를 적용한 웹서비스 개발 프로세스

#### 3.1 웹서비스 개발 프로세스 유형(WSPT, WebService Process Type)

웹서비스의 아키텍처는 (그림 4)와 같이 서비스저장소, 서비스공급자, 서비스요청자의 세가지 역할을 가지는 컴포넌트로 구성된다. 웹서비스의 개발 절차상에 서비스공급자와 서비스요청자가 어떻게 상호작용을 하는지가 중요하며, 웹서비스의 입장에서 보면 하나의 서비스는 서비스공급자에게도 서비스요청자에게도 모두 의미를 가질 수 있다.

웹서비스 아키텍처에서 볼 때 웹서비스 개발 생명주기는 개발, 배포, 실행의 단계를 포함하며 개발(bulid)은 서비스공급자가 웹서비스를 개발하고 웹서비스 인터페이스를 생성하고 이것을 공개하는 단계이며 배치(deploy)는 이렇게 생성된 웹서비스가 배치되는것이고 실행(run)은 공개된 웹서비스 인터페이스가 서비스의 요청자에 의해 호출되어지는 단계이다.

웹서비스를 개발하기 위한 개발 프로세스는 서비스 공급자나 요청자에 따라 다른 개발 절차를 따르게 된다. 웹서비스는 서비스모델링부터 시작하여 처음 개발하는 서비스 공급자의 경우엔 일반적인 소프트웨어 개발 프로세스를 따르게 되지만 서비스저장소를 검색하여 외부 웹서비스를 요청

<표 1> 웹서비스개발 프로세스 유형

	웹서비스인터페이스	
래거시애플리케이션	없음	있음
없음	WSPT1	WSPT2
있음	WSPT3	WSPT4

하여 사용하려는 웹서비스 요청자의 경우엔 서비스저장소에 존재하는 WSDL문서를 검색하여 조건에 맞는 서비스를 찾아야 하고 찾는 웹서비스가 있느냐 없느냐에 따라 다른 프로세스를 가지게 된다.

따라서, 이러한 웹서비스 개발 프로세스는 웹서비스가 될 서비스 인터페이스(WSDL)유무와 래거시(legacy) 애플리케이션의 유무에 따라 <표 1>의 4가지 프로세스 유형(WSPT)으로 분류할 수 있다.

##### 3.1.1 WSPT1(WebService Process Type 1)

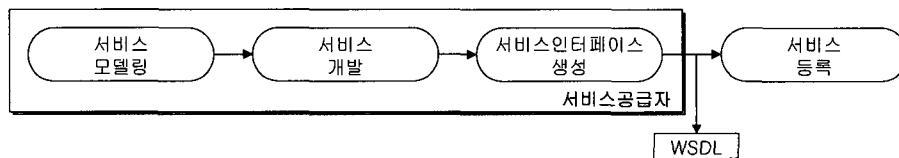
서비스공급자의 입장에서 새로운 서비스개발하는 경우이다. 이경우엔 일반적인 소프트웨어 개발 프로세스가 그대로 적용된다. 서비스모델링의 단계가 MDA의 PIM모델 생성 단계로 볼 수 있으며 이렇게 생성된 PIM모델은 웹서비스형태의 PSM으로 전환되어 공개된다. 이 경우 제안된 방법은 서비스 모델링 단계 이후에 WSDL을 생성하여 배포함으로써 서비스 요청자가 참조하여 WSPT2의 절차를 따를 수 있도록 해 준다.

##### 3.1.2 WSPT2(WebService Process Type 2)

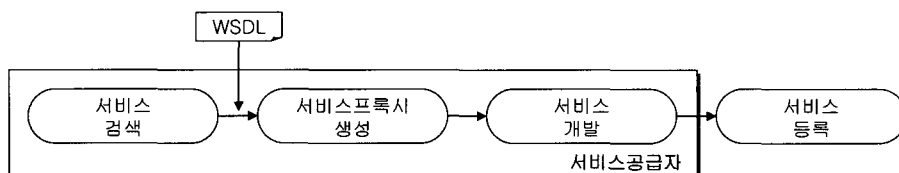
WSPT2는 서비스 공급자가 다른 서비스 공급자가 개발한 서비스를 이용하여 새로운 서비스를 개발하는 경우이다. WSPT1단계의 프로세스를 따라 서비스 모델링된 PIM에서 생성된 WSDL을 검색(discover)하고 그 서비스에 접근하기 위한 프록시를 생성하여 새로운 서비스를 개발한다. 이렇게 생성된 서비스는 다시 서비스저장소에 등록(publish)될 수 있다.

##### 3.1.3 WSPT3(WebService Process Type 3)

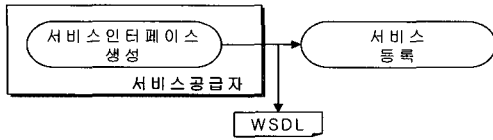
EJB, JavaBean, Servlet, C++, Java class, COM class로



(그림 5) WSPT1



(그림 6) WSPT2



(그림 7) WSPT3

구현된 레거시 애플리케이션이 있는 경우 서비스 인터페이스만 생성함으로써 웹서비스를 등록(publish) 할 수 있다.

3.1.4 WSPT4(WebService Process Type 4)

레거시 애플리케이션을 웹 서비스로 전환하여 서비스를 개발하는 경우이다.

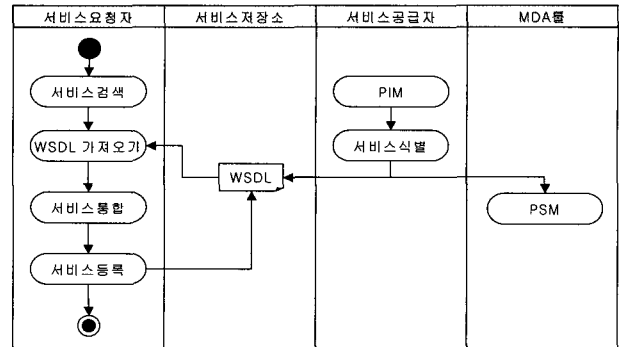
3.2 MDA를 적용한 웹서비스 개발 프로세스

웹서비스를 개발하여 공급하는 웹서비스공급자의 입장에서 일반적인 웹서비스 개발 프로세스는 (그림 3)에서 보여주는 일반적인 개발 프로세스에서 크게 벗어나지 않는다.

웹서비스의 여러 가지 구성 요소 중 웹서비스에 대한 정보를 표현하는 WSDL모델은 그 웹서비스가 제공하는 기능과 인터페이스 정보를 웹서비스 요청자에게 제공한다. MDA 중심에는 PIM이 있고 툴과 변환작업이 애플리케이션을 만들어내는 데 중요한 역할을 한다. 따라서 MDA에서는 분석 모델인 PIM을 설계 모델인 PSM으로 자동으로 변환하도록 하는 변환규칙이 매우 중요하다. 웹서비스개발 프로세스에서 서비스공급자가 웹서비스모델링의 결과로 PIM을 만들고, 이것에 웹서비스를 위한 모델인 WSDL을 자동으로 생성해 줄 수 있는 변환규칙이 제공되면 웹서비스의 공급자와 요청자가 동시에 웹서비스개발과 통합에 참여할 수 있으며 이를 통해 생산성과 효율성을 높일 수 있다. (그림 9)는 이렇게 MDA를 적용하고 웹서비스공급자와 요청자를 고려한 웹서비스 개발 프로세스를 보여준다. (그림 9)에서 웹서비스 공급자와 웹서비스 요청자의 관점에서 본 웹서비스 사용 프로세스를 보여주며 서비스모델링과 WSDL등록은 서비스 공급자입장에서 이루어지는 작업이며 보통의 경우엔 일반적인 웹서비스 개발툴을 사용함으로써 구현한 후 WSDL을 자동으로 생성하게 된다.

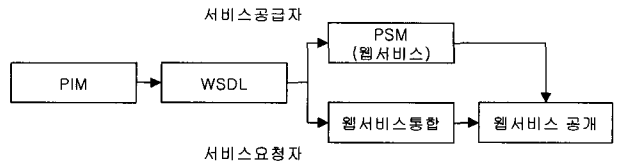
3.3 제안된 웹서비스 개발 프로세스의 장점

지금까지도 소프트웨어 시스템 개발시, MDA나 기존의 객체지향 분석 설계 방법론을 쓰지 않고 간단한 시스템 분석 후에 구현을 바로 하는 경향이 있다. 웹서비스의 개발에서도 이것은 예외가 아니며 현재 많은 웹서비스 지원 개발



(그림 9) MDA를 적용한 웹서비스 개발 프로세스

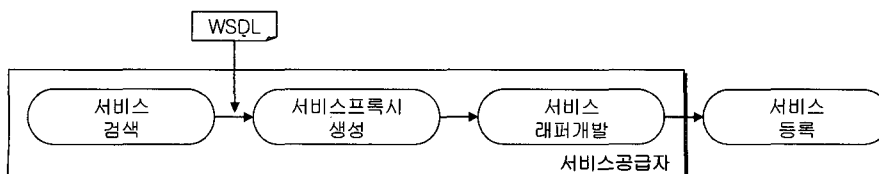
툴에서도 구현되어진 단위 웹서비스에서 WSDL을 자동으로 생성하게 된다. 이러한 방법을 사용하는 경우 구현되어진 소스코드에서 자동 생성된 WSDL은 원래의 XML스키마에서 생성된 WSDL보다 덜 형식화되어 있다. XML스키마의 많은 특징들은 자동 WSDL생성기로는 지원되지 않으므로 데이터 정의를 하는데 있어서 XML기반의 접근에 많은 부분이 손실되어지게 된다. 또한, 소스코드의 구현물에서부터 WSDL을 생성하게 되면 다양한 플랫폼 사이의 상호운용성 문제를 야기하게 한다. 예를 들어 JAVA Remote Reference를 사용한 서비스와 .NET기반 클라이언트와의 상호작용이 어렵다. 제시한 웹서비스 개발 프로세스를 사용하면 이러한 상호작용성의 문제들은 서버나 클라이언트가 공통의 XML스키마 타입을 사용함으로써 해결할 수 있다.



(그림 10) 웹서비스 공급자와 요청자의 웹서비스 개발

(그림 10)에서 보는바와 같이 본 논문에서 제안하는 방법은 MDA의 PIM모델에서 WSDL을 생성하여 그 결과물은 WSDL을 서비스공급자와 서비스 요청자가 동시에 사용할 수 있도록 한다. WSDL을 기존의 웹서비스 개발툴에서처럼 마지막 구현물을 가지고 생성하지 않고 PIM에서 생성해주는 경우 장점은 다음과 같다.

첫째, WSDL을 먼저 생성할수 있다면 그 다음 단계에는 자동화된 툴을 사용하여 서비스를 구현하는 클라이언트와 서버쪽의 코드를 생성해낼 수 있다. 이것은 많은 양의 코드



(그림 8) WSPT4

를 작성하는 시간을 줄임으로써 생산성을 향상시킨다. 그리고 복잡한 인터페이스를 가지는 경우 코딩을 먼저 하는 것은 매우 어렵다. MDA의 장점을 살려 복잡한 스키마에서 모든 데이터타입에 대한 인터페이스를 자동으로 생성하여 생산성을 향상시킬 수 있다.

둘째, 디자인과 개발을 분리시킬 수 있다. 서비스 디자인은 서비스개발자에게 WSDL을 통하여 요구사항을 명시할 수 있고 그에 따라 개발자가 구현하게 할 수 있다.

셋째, 서비스공급자와 요청자는 동시에 작업을 할 수 있다. WSDL로 서비스 인터페이스를 정의함으로써 client-side 개발자는 서버쪽의 개발자와는 독립적으로 작업을 진행할 수 있다. 또한 클라이언트쪽 개발자와 서버쪽 개발자가 같은 개발언어를 사용할 필요도 없다. 따라서 분산애플리케이션 개발에 참여할 수 있는 개발자의 수를 늘릴 수 있다.

넷째, WSDL은 웹서비스 프레임워크 사이에서 이식가능하므로 효과적으로 서비스디자인을 분리하여 웹서비스를 구현할 수 있도록 한다.

#### 4. 결론 및 향후 연구 방향

본 논문에서는 웹서비스 공급자와 요청자가 웹서비스를 개발하거나 통합할때의 절차를 분류하여 웹서비스 개발 프로세스 유형을 만들고 웹서비스 개발시 MDA개발 프로세스를 적용하여 웹서비스공급자와 요청자가 참조가능한 새로운 웹서비스 개발 프로세스를 제안하였다.

웹서비스 개발 프로세스 유형은 웹서비스의 공급을 위해 개발하는 개발자나, 웹서비스를 통합하여 새로운 웹서비스를 만드는 요청자 모두에게 개발 프로세스를 선택하기 위한 가이드를 줄 수 있다. 또한 웹서비스를 개발하기 위하여 MDA를 적용한 효율적인 개발 프로세스를 제시함으로써 웹서비스 공급자와 요청자의 웹서비스 개발에 활용할 수 있으며 이것은 전체적인 웹서비스 개발 프로세스 개선에 기여할 수 있다. MDA는 소프트웨어 개발에 있어서 매우 흥미롭게 접근되어지고 있고 더욱더 많은 연구가 이루어질 전망이다.

향후에는 제시한 웹서비스 개발 프로세스 내에서 식별된 웹서비스가 모델링된 PIM을 WSDL로 변환시키기 위한 변환규칙을 정의함으로써 제안한 웹서비스 공급자 입장에서의 웹서비스개발프로세스에 MDA의 PIM에서 웹서비스모델인 WSDL을 자동으로 생성할 수 있도록 하는 MDA들을 만드는 데 기여할 수 있다.

#### 참고 문헌

[1] W3C Web Services Activity, "http://www.w3.org/2002/ws"  
 [2] David Sprott, "Understanding the Component and Web Services Market", CDBI Journal 2001, 5  
 [3] W3C Web Services Description Language(WSDL) 1.1, "http://www.w3.org/TR/2001/NOTE-wsdl-20010315"  
 [4] Scott W. Ambler, "Deriving Web Services from UML models", IBM, Mar 2002, http://www-106.ibm.com/

developerworks/library/ws-uml1/  
 [6] Anneke Kleppe, Jos Warmer, Wim Bast, "The Model Driven Architecture : Practice and Promise", Addison-Wesley, April, 2003.  
 [7] Joaquin Miller and Jishnu Mukerji, "MDA Guide version 1.0.1", 2003 OMG  
 [8] Peter Brittenham, "Web Services Development Concepts (WSDC 1.0)", IBM, May, 2001.  
 [9] Jean Bezivin, Slimance Hammoudi, Denivaldo Lopes, Frederic Jouault, "Applying MDA Approach for Web Service Platform", EODC 2004, IEEE  
 [10] P. C'aceres et al. "A MDA-Based Approach for Web Information System Development", Workshop in Software Model Engineering in conjunction with UML Conference, SanFrancisco, 2003.  
 [11] E. Marcos; P. C'aceres and V. de Castro, "An Approach for Navigation Model Construction from the Use Case Model", The 16th Conference On Advanced Information Systems Engineering, CAISE'04



#### 윤 홍 란

e-mail : hryun@sookmyung.ac.kr  
 1992년 숙명여자대학교 전산학과(학사)  
 1995년~1997년 (주)트라이콤 솔루션사업부  
 1997년 숙명여자대학교 교육대학원  
 전산교육학과(석사)  
 1998년~1999년 숙명여자대학교 가상교육  
 센터 연구원

2000년~2001년 (주)이나우테크놀로지 제품개발팀  
 2001년~2003년 안양과학대학 초빙전임강사  
 2000년~현재 숙명여자대학교 정보과학부 컴퓨터과학과 박사과정  
 관심분야 : 소프트웨어 공학, 시스템 개발 방법론, 웹 서비스, 서비스지향아키텍처(SOA), 서비스 식별, MDA



#### 박 재 년

e-mail : jnpark@sookmyung.ac.kr  
 1966년 고려대학교 물리학과(이학사)  
 1969년 고려대학교 대학원(이학석사)  
 1972년 독일 함부르크대학 전산학 전공  
 1981년 고려대학교 대학원 물리학 전공  
 (이학박사)  
 1970년~1972년 독일 국립 입자 가속기  
 연구소(DESY) 연구원

1972년~1979년 고려대학교 전자계산소 총간사  
 1978년~1983년 전남대학교 계산통계학과 교수  
 1987년~1998년 숙명여자대학교 전산원 원장  
 1991년~1991년 숙명여자대학교 사무처장  
 1994년~1996년 정보과학회 이사  
 1994년~1996년 전산교육 연구회 운영위원장  
 1983년~현재 숙명여자대학교 정보과학부 교수  
 2002년~현재 숙명여자대학교 이과대학 학장  
 관심분야 : 시스템 분석 및 설계, 정보구조 모델링, 컴포넌트 기반 개발 방법론