웹서비스 개발 프로세스에서 구현전략 결정을 위한 평가 지침

(Assessment Guidelines for Decision Making of Implementation Strategy in Web Services Development Process)

김 유 경 † 윤 홍 란 †† 박 재 년 †††
(Yukyung Kim) (Hongran Yun) (Jaenyun Park)

요 약 이기종의 분산시스템들을 통합하여 기업 환경에 적용할 수 있도록 여러 다양한 연구들과 시도들이 이루어지고 있다. 그러나 웹서비스가 갖는 특별한 구조 즉, 서비스 제공자, 서비스 요청자, 그리고 서비스 저장소의 세 역할을 갖기 때문에, 웹서비스 기술에 기존의 시스템 개발 방법론을 직접 적용하기는 어렵다.

M4WSD(Method for Wes Services Development)는 웹서비스 개발 프로세스 모델로서, 요구사항 분석 단계에서 유도되는 유스케이스 모델을 기반으로 하는 웹 서비스 개발 절차 및 가이드라인을 포함하고 있다. 본 논문에서는 각 서비스를 구현하기 위하여 주요 구현 전략을 어떻게 결정하는지에 초점을 맞춘다. 본 논문에서 제시하는 평가 가이드라인은 구현 전략을 결정하는 문제를 구조화할 수 있도록 한다.

키워드 : 서비스지향 아키텍처, 웹서비스, 개발 프로세스, 서비스 평가

Abstract To integrate heterogeneous distributed systems, there exist various researches and developments for the purpose of its adoption into enterprise environment. However, when web service technologies are applied, it is difficult to adopt directly existing software system development methodologies, because of the peculiar architecture of web services, such as service provider, service requester, and service repository. M4WSD(Method for Wes Services Development) is a web service development process model and involves procedures and guidelines to develop web services based on a Use Case model that is elicited from a business domain in requirement analysis. In this paper, we focus on how to determine key realization decisions for each service. The assessment guidelines help you to structure the problem of determining implementation strategy.

Key words : Service-Oriented Architecture, Web Service, Development Process, and Service Assessment

1. 서 론

서비스 지향 아키텍처(Service-Oriented Architecture; 이하 SOA)가 기업 내 차세대 인프라구조의 핵심으로 자리잡아가면서, 이에 대한 관심이 증폭되고 있다. SOA는 한마디로 비즈니스 기능들을 네트워크 상에서 재사용이 가능한 공유서비스의 집합으로 구현한 소프트웨어 설계 패러다임이다. 따라서 기업들이 SOA를 도입하면 기존 소프트웨어의 재사용, 이기종 환경(heterogeneous) 통합, ROI(Return on Investment) 극대화 등 다양한 이점을 얻을 수 있다. 시장 조사업체인 가트너 그룹은 '2008년에는 60% 이상의 기업이 회사의 핵심 애플리케이션과 프로세스를 개발할 때 SOA를 소프트웨어 개발 원칙으로 사용할 것'이라고 예상하고 있다[1].

SOA는 표준화된 인터페이스와 메시징 프로토콜을 이용한 약결합(Loosely coupled) 방식을 사용함으로써 플랫폼에 보다 유연하면서 표준화된 개발 방법을 제공한다. 따라서 이기종 시스템 간의 통합 요구 및 비즈니스 환경 변화에 신속하게 대응할 수 있는 환경을 제공해 줄 수 있다.

· 본 연구는 한국전산원 위탁연구과제(2005-정평-B08)로 수행되었음
† 정 회 원 : 숙명여자대학교 컴퓨터과학과 교수
ykkim18@sookmyung.ac.kr
†† 정 회 원 : 숙명여자대학교 컴퓨터과학과
hryun@sookmyung.ac.kr
††† 종신회원 : 숙명여자대학교 컴퓨터과학과 교수
jnpark@sookmyung.ac.kr
논문접수 : 2005년 9월 28일
심사완료 : 2006년 3월 9일

SOA는 이미 CORBA나 DCOM 등의 분산 객체 기술에서 그 기본 개념이 사용 되었다. 그러나 표준화된 개발 방법의 부재로 인한 벤더 간의 상호운용 문제 때문에 주목 받지 못하였다. 최근 XML 기반의 웹서비스 기술이 등장하면서 SOA가 새롭게 조명되고 있으며, 웹서비스는 SOAP, WSDL, 그리고 UDDI 등의 공개표준을 사용하고 있다. 따라서 서로 다른 벤더 간의 상호운용이 용이하여, 최상의 SOA의 구현기술로 자리 잡고 있다[2].

하지만 SOA는 기업이 비즈니스 시스템과 애플리케이션을 설계하는 방식에 중요한 변화를 가져온다. SOA를 수용하기 위해 선결되어야 할 과제는 체계적인 SOA 수용 전략 가이드 부재의 문제이다. 기존 시스템(Legacy System)을 대상으로 한 연계 및 통합 방안을 제시할 수 있는 절차에 대한 가이드라인과 상호 운용성 보장을 위한 SOA 관점의 정보 시스템 개발 가이드라인이 없다는 것이다.

이것은 웹서비스 기반의 SOA 구현에 대한 세부 기술 적용 지침 및 개발 모델 부재에 기인하며, M4WSD는 이 문제에 대한 해결책으로서 웹서비스 적용을 위한 개발 절차 및 구현전략 결정에 필요한 구체적인 평가 지침을 제공하기 위한 개발 모델이다.

본 논문에서는 M4WSD의 구현 전략 결정 과정에서 사용되는 단위 서비스 평가 가이드라인을 제안한다. 이 가이드라인은 식별된 단위 서비스들에 대한 구현 전략들 가운데 가장 비용효과적인 방법을 결정할 수 있도록 제공되는 지침이다.

논문의 구성은 다음과 같다. 먼저 2장에서는 SOA와 웹서비스에 대한 개념적 소개와 기존의 SOA 지원 개발 방법론에 대한 연구들을 살펴본다. 3장에서는 SOA 구현을 위한 웹서비스 개발 방법론인 M4WSD에 대하여 설명하고, 4장에서는 M4WSD의 다섯 번째 단계인 단위 서비스 개발 여부 판단 단계에서 요구되는 구현 전략 결정을 위한 의사 결정 과정에 대한 가이드라인을 제시한다. 이것은 단위 서비스를 실체화하기 위한 구체적인 접근 방법을 제공함으로써 SOA 구현에 대한 세부 기술 적용 지침을 포함하는 정형화된 접근 방법이다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 SOA와 웹서비스

SOA는 소프트웨어 아키텍처의 일종이며, 서비스 요청자(Service Requester), 서비스 공급자(Service Provider), 서비스 저장소(Service Registry)의 3가지 역할을 포함하고 있다. 이런 구조를 통해서 SOA는 서비스의 발견과 동적 바인딩이라는 개념을 지원한다. 어떤 서비스를 필요로 하는 사용자는 작업 수행 중(Runtime)에 필요로 하는 서비스를 찾고, 그것을 사용할 수 있는 것이다[3].

반면 웹서비스는 웹상에서 정의된 모듈화 된 소프트웨어 컴포넌트로서, 개방형 표준 데이타 표현 기법인 XML과 인터넷 프로토콜을 결합시킨 새로운 패러다임에 의해 탄생된 분산 컴퓨팅 기술이다. W3C는 웹서비스를 URI에 의해 인식되는 소프트웨어 애플리케이션으로서, 인터넷 기반의 프로토콜을 통하여 교환되는 XML 기반 메시지를 사용하여 다른 소프트웨어 에이전트들과의 직접적인 상호작용을 지원한다고 정의 한다[4].

웹서비스는 SOAP, WSDL, UDDI를 통해 SOA의 주요 요소인 메시지, 서비스 인터페이스, 서비스 공개 및 발견 체계를 구현한다. 따라서 웹서비스는 SOA 구축에 필요한 표준 기술들을 제공한다. 그림 1에서 볼 수 있듯이 웹서비스의 기본적인 아키텍처는 SOA를 채택하고 있다. 웹서비스의 기본 동작 방식은 서비스 기술, 서비스 등록 및 발견, 서비스 간의 통신 관점에서 정의된다.

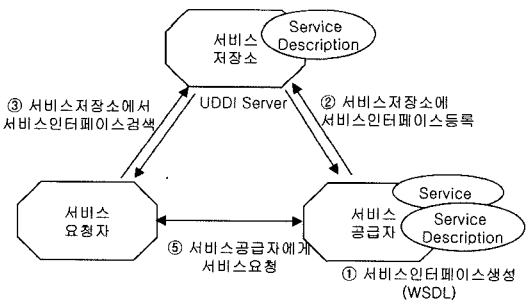

그림 1 웹서비스 아키텍처

SOA는 서비스 공급자, 서비스 요청자, 서비스 저장소라는 세 요소가 유기적으로 연계되어 있으며 이는 SOA를 구현하는데 있어 기존의 개발 방법론의 적용을 어렵게 한다. OOAD나 CBD 방법론의 경우 SOA의 구성요소들을 고려하지 않는다. 또한 서비스를 식별하고 명세하기 위한 구체적인 방법이나 절차를 제시하지 못해 그러한 지원이 필요하다.

게다가 SOA에 존재하는 두 가지 역할인 서비스 공급자와 서비스 사용자 사이의 상반된 요구사항을 적절히 수용하기 위한 체계가 필요하다. 따라서 SOA를 지원하는 개발방법론의 연구가 필요하다.

2.2 SOA 지원 방법론에 대한 기존 연구

SBD(Service Based Development)는 기존의 CBD (Component Based Development) 방법론에서 확장되었으며 SOA를 지원하기 위한 개발 방법론이다. SBD에서는 기존의 OOAD 나 CBD에서 찾을 수 없는 추가적

인 활동이나 산출물이 요구된다. 현재 소개되고 있는 SBD로서는 Gartner에서 정의한 SODA(Service Oriented Development of Applications)[5], RUP와 XP의 특징만을 모아서 만든 SOUP(Service Oriented Unified Process)[6,7]과 IBM에서 정의한 SOMA(Service-Oriented Modeling and Architecture)[8]와 SOAD(Service Oriented Analysis and Design)[9]가 있다.

SODA는 SOA를 구현하기 위해 서비스를 설계, 개발 및 조합 등 일련의 과정에 대한 원칙을 제공한다. 서비스를 개발하고 조합하여 비즈니스 프로세스를 지원할 수 있는 애플리케이션 개발 모델 제공을 목표로 한다. SODA는 CBD, 분산개발, SOA를 통합하여 공통적 개념들을 뽑아 적용하였다.

SOUP은 Knual Mittal이 제안한 것으로서, 소프트웨어 개발의 6단계를 정의하고, 각 단계에서 수행해야 되는 활동들을 포함하고 있다. SOUP의 프로세스는 SOA 배치(Deployment)와 배치 후의 SOA 관리의 두 부분으로 나누어진다. SOA 배치부분은 RUP에 기반을 두고 작성되었으며 SOA 관리부분은 XP에 기반 해서 만들어졌다[7].

SOMA는 SOA를 지원하는 모델링, 분석 설계 기술과 활동을 포함하며, SOA의 각 레이어 내의 요소를 정의하고 각 레벨에서 일어나는 아키텍처적 결정을 하도록 한다. SOMA는 상향식(Bottom-Up) 접근법과 하향식(Top-Down) 접근법을 통합하는 방법이다. 이러한 방식으로 상위수준의 비즈니스 프로세스 기능은 크기가 큰(Large-grained) 서비스로 식별되고 좀더 작은 단위(Smaller-grained) 서비스는 기존에 존재하는 기능을 중심으로 식별한다. 이로써 시스템 내에 있으나 외부 비즈니스 협력자나 사용자가 사용하고자 하는 기능을 외부로 끌어내어 어떻게 어댑터(Adaptors)나 래퍼(Wrappers)를 만들 건지 결정하게 하는 방법을 포함한다. SOMA는 서비스의 식별(Identification), 명세(Specification), 실현(Realization) 세 단계로 구성된다. 서비스 식별은 비즈니스 도메인(Domain)을 기능을 중심으로 하여 서브시스템으로 분해하는 하향식과 이미 가지고 있는 시스템의 분석을 통해 비즈니스 프로세스에서 수용 가능한 기능을 가진 후보 서비스를 상향식으로 식별한다. 앞의 상향식이나 하향식으로 식별되지 않은 다른 서비스를 찾아내는 목표 서비스 모델링(Goal-Service Modeling)으로 구성 된다[8].

SOAD는 OOAD와 EA 구조(frameworks), BPM과 같은 기존의 모델링 개념들을 SOA에 적용시킨 방법론이다. OOAD, EA, 그리고 BPM 개념들을 모두 포함하고 SOA에서 요구하는 사항들을 지원할 수 있도록 상향식 및 하향식을 포함한 방법론으로 제시되었다[9].

지금까지 대표적인 SOA 지원 개발 방법론에 대하여 살펴보았다. 다음 표 1은 이상에서 설명한 SOA 지원 개발 방법론들의 특징을 정리한 것이다.

객체지향 방법론 및 CBD의 개발 절차나 표기법 등은 SOA에서 요구하는 요구사항을 부분적으로는 지원하지만, 서비스 규약, 서비스 저장소나 서비스 버스와 같은 SOA에서 요구하는 사항에 대한 전반적인 지원이 미흡하므로, 추가적인 활동과 산출물들이 필요하다. 일반적인 SOA 모델링에서는 비즈니스 모델링에 CBM (Component Business Modeling)과 같이 상향식을 따르고 있어 기존 시스템이 가진 많은 양의 중요한 데이타와 비즈니스 로직에 대한 재사용을 고려하지 않는다. 따라서 하향식으로 래핑과 같은 전략을 적용해야 한다. 따라서 웹서비스 기반의 SOA 구현 모델의 제시와 함께 구체적 구현 기술 적용 및 활용에 대한 방안이 필요하다.

표 1 SOA지원 방법론의 특징

방법론	특 징
SODA	· CBD, 분산시스템 개발, SOA의 공통 요소를 뽑아 적용 · SOA를 구현하기 위한 서비스 설계, 개발, 조합에 관한 원칙을 정의 · 서비스 단위나 세부 절차에 대하여 구체적으로 명시하지 않음
SOAD	· OOAD, EA, BPM 같은 기존의 모델링 개념들을 SOA에 적용시킴 · 하향식과 상향식을 연결하여 기존시스템의 통합을 위한 아이디어 제시 · 세부적인 절차나 활동을 명시하지 않음
SOUP	· RUP와 XP의 특징만을 모아서 SOA에 적용시킨 방법론 · 소프트웨어 개발 6단계를 정의하고, 단계별 활동을 정의 · 비즈니스 프로세스, 서비스 규약(choreography)이나 서비스 저장소 같은 SOA에 대한 요구사항을 지원하지 않음
SOMA	· 기존 IBM의 SOAD를 참조하여 확장된 방법론 · 서비스 식별, 명세, 실현의 세단계로 구성 · 하향식과 상향식방식을 통합하여 정의 · 서비스 식별에 대한 구체적인 활동이 제시되지 않음

M4WSD[10]는 SOA 구현을 위한 웹서비스 개발 방법론으로서, 웹서비스 기반의 SOA 구현을 위한 프로세스 정의와 함께, 각 단계의 활동과 산출물 및 서비스 식별을 위한 지침과 구현 전략 결정을 위한 서비스 평가 지침을 포함하고 있다.

3. 웹서비스 개발 방법론 M4WSD

3.1 SOA 구현을 위한 개발 프로세스

개발시스템에 대한 요구사항 및 특성에 따라 적합한 개발 프로세스를 수립할 수 있도록 SOA 기반 개발 프로세스는 크게 3가지 유형으로 분류한다.

• 프로세스1

기존의 시스템을 고려하지 않고 서비스가 적용된 새로운 시스템을 개발하고자 하는 경우이다. 서비스개발을 통해 서비스공급자가 되는 경우와 서비스 개발여부 판단에 따라 서비스 요청자가 되는 경우가 포함된다.

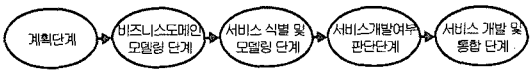

그림 2 프로세스1의 세부 절차

이 프로세스는 기존시스템과 독립적인 새로운 비즈니스 추가가 많이 요구되는 경우 그리고 기존시스템에 활용할 가치 있는 자원이 없는 경우에 해당한다.

• 프로세스 2

서비스적용을 위한 정보는 물론 서비스 식별을 위한 후보 서비스 재사용이 가능한 경우이다.

기존 시스템의 재사용이 전략적으로 필요한 경우나 서비스화를 위한 기존 시스템의 컴포넌트 분리가 가능한 경우에 해당된다.

그림 3 프로세스2의 세부 절차

• 프로세스 3

기존 시스템의 서비스 식별작업과 새로운 비즈니스에서의 서비스 식별 및 생성 작업이 명확하게 분리되는 경우이다.

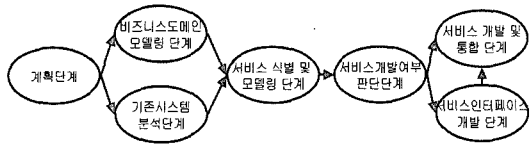

그림 4 프로세스3의 세부 절차

이 프로세스 유형은 계획단계에서 변환 전략 및 기법들을 명확히 결정할 수 있는 경우, 또는 목표시스템이 부분적으로 기존시스템의 재사용이 효율적이거나 새로운 비즈니스 확장을 요구하는 경우에 해당한다.

3.2 M4WSD 세부 단계 및 활동

M4WSD는 일반적인 소프트웨어 개발 방법론과 SOA의 특징을 고려한 개발 방법을 적용하여 새로운 시스템 개발 시 SOA를 적용하는 경우와 기존 시스템이 있는 경우 기존시스템 기반에 SOA를 적용하게 되는 기존시스템의 재활용을 고려해야 하는 경우 적용할 수 있다.

M4WSD은 총 6 단계(Phases)로 이루어져 있으며, 각각의 활동(Activities)과 세부 타스크(Tasks), 그에 따른 산출물로 구성된다. 각 단계는 계획단계에서 선택한 개발 프로세스에 맞추어 선택되어 적용될 수 있다. 아래의 그림 5는 M4WSD의 전체 개발 단계 및 활동들을

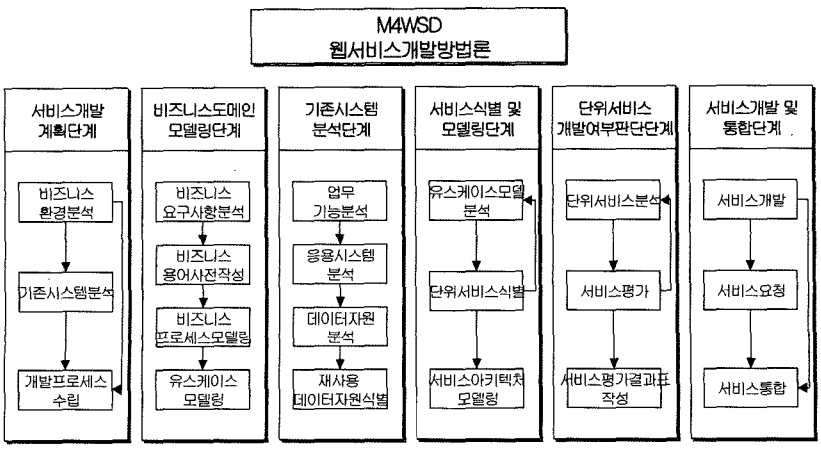

그림 5 M4WSD 전체 단계 구성도

나타낸다.

(1) 서비스 개발 계획(SDP:Service Development Planning) 단계

이 단계의 목표는 개발 프로세스를 수립하는 것이다. 새로운 시스템을 개발해야 하는 경우 비즈니스 환경 분석을 통해 개발 프로세스를 선택한다. 기존 시스템을 기반으로 개발해야 하는 경우, 비즈니스 환경 분석과 기존 시스템의 분석을 통하여 개발 프로세스를 수립한다. 개발 프로세스 수립 활동은 적합한 개발 프로세스 유형을 선택하는 것이다. 프로세스 3의 경우엔 기존시스템의 재활용 뿐 아니라 새로운 비즈니스에 대한 요구사항이 추가되는 경우에 프로세스 1과 프로세스 2가 조합된 형태이다.

업무현황과 조직체계 업무 프로세스에 대한 관련 정보를 참고하여 정확한 업무 프로세스파악을 위한 담당자 면담과 그 결과서, 조직 구성도, 조직간의 업무 흐름도를 산출하고 기존시스템이 있는 경우 이에 대한 환경과 구성도등을 포함하는 기존 시스템 분석서를 산출한다.

(2) 비즈니스 도메인 모델링(BDM:Business Domain Modeling) 단계

이 단계에서는 비즈니스 프로세스에 참여하는 스테이크홀더들이 비즈니스 도메인을 공통적으로 이해할 수 있도록 4개의 활동을 통해 비즈니스 도메인을 상세히 분석한다. 비즈니스 요구사항을 입력으로 하여, 관련 당사자 목록, 비즈니스 도메인 기술서, 용어 사전, 그리고 유스케이스 다이어그램을 산출한다.

(3) 기존 시스템 분석(LSA:Legacy System Analyzing) 단계

계획단계에서 기존시스템의 재활용이 가능하다고 결정되어진 경우 수행해야 하는 단계이다. 기존 시스템 분석서를 기반으로 재활용 전략을 수립한다. 재활용전략을 통하여 애플리케이션의 재사용인지 데이타차원의 재활용인지가 결정되면 이에 따라 필요한 형태의 재사용 자원이 식별된다. 이 단계를 통하여 재사용 가능한 자원의 목록을 산출한다.

(4) 서비스 식별 및 모델링(SIM:Service Identification & Modeling) 단계

비즈니스 프로세스와 비즈니스 유스케이스 모델을 기준으로 서비스를 식별하고 식별된 단위서비스에 대한 서비스모델이 생성되는 단계이다. 이 단계에서는 이전 단계에서 만들어진 유스케이스 다이어그램을 사용하여, 서비스를 식별하게 되고, 식별된 서비스에 대한 서비스 인터페이스 명세서, 서비스 아키텍처 다이어그램, 그리고 단위 서비스 목록을 산출한다.

(5) 단위 서비스 개발여부 판단(SDM:Service Decision Making) 단계

서비스모델을 참조하여 이 서비스를 개발할 것인지 기존시스템의 래핑을 통해 서비스화 할 것인지, 서비스 저장소에 공개된 서비스를 사용할 것인지를 판단하는 단계이다. 이 단계의 판단에 따라 서비스를 개발하거나 레거시 래핑(legacy wrapping), 서비스를 요청하는 단계로 넘어간다. 이 단계의 산출물은 이전 단계에서 식별된 단위 서비스들의 목록을 바탕으로 하여, 서비스 평가 기준을 정의한 평가 기준서와 서비스 평가 결과 테이블을 산출한다.

(6) 서비스 개발 및 통합(SDC:Service Development Composition) 단계

신규 개발된 서비스와 기존시스템을 래핑해서 생성된 서비스, 서비스저장소를 통해 얻은 정보를 기반으로 요청하는 서비스를 통합하기 위한 단계이다. 각 서비스들에 대한 구현 전략이 결정된 서비스 평가 결과를 통해, 각 서비스에 대한 구체적인 명세서와 서비스 통합 명세서가 산출된다.

M4WSD의 6단계는 다음 그림 6과 같은 실행 흐름을 갖게 되며, 이는 서비스 개발 계획 단계의 개발 프로세스 수립 활동에 의해 정의된 프로세스 유형들에 따라 달라진다. 즉, 새로운 시스템 개발 시 SOA를 적용하는 경우와 기존시스템 기반에 SOA를 적용하게 되는 경우

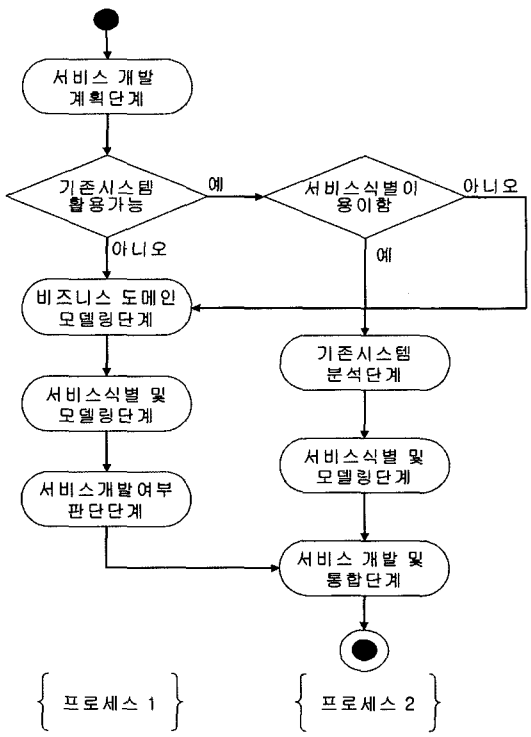

그림 6 프로세스 유형에 따른 흐름

에 따라 흐름이 나뉘게 된다. 기존 시스템 기반에 SOA를 적용하게 되는 경우에는 기존시스템의 재활용을 고려해야 하는 활동들이 필요하게 된다.

4. 구현 전략 결정을 위한 평가 가이드라인

본 논문에서 제안하는 평가 가이드라인은 M4WSD의 다섯 번째 단계인 서비스 개발 여부 판단 단계에서 이루어지는 활동들을 위한 지침으로서, 식별된 단위 서비스에 대한 구현 전략을 결정하기 위한 절차를 제공한다.

서비스 식별 및 모델링 단계에서 단위 서비스가 식별되고 나면, 서비스에 대한 상태를 평가하여 구현에 대한 의사 결정이 이루어져야 한다. 즉, 서비스가 개발되기 위해서는 비즈니스 도메인에서 식별된 단위 서비스 각각에 대해 이 서비스 컴포넌트가 새로 개발될 것인지, 기존에 존재하는 시스템 컴포넌트의 래핑을 통해 서비스화 될 것인지, 또는 다른 서비스 공급자에 의해 개발된 서비스를 요청해서 사용할 것인지를 먼저 결정해야 한다.

서비스 개발 전략을 결정하기 위해서는 무엇보다 서비스 평가에 대한 구체적인 지침이 필요하다. 본 가이드라인은 이런 과정에서 단위 서비스에 대한 개발 전략을 보다 정성적인(Qualitative) 방법을 통해 결정할 수 있도록 도와줄 것이다.

평가 과정은 각 스테이크 홀더들이 다양한 평가 기준(Assessment Criteria, 이하 AC)에 대해 다섯 가지 스케일로 투표를 하고, 이 결과를 정량적인 값으로 변환하여 각 서비스에 대한 이익을 추정하는 단계로 이루어져 있다. 각 AC들의 중요도에 따라 가중치를 반영할 수 있고, 이 가중치는 시스템의 중요한 목적(Goal)에 따라 달라질 수 있다. 그림 7은 평가 과정의 각 단계를 보여주고 있다.

4.1 평가 기준 정의하기

이 단계는 각 서비스에 대해 이 서비스가 개발되었을 경우 어떤 투자 가치가 있는지를 먼저 조사하는 것이다. 투자 가치란 비용뿐만 아니라 서비스의 비즈니스 프로세스와 관련된 가치 및 전체 시스템 차원의 기술적인 가치와 같은 다양한 요소가 있다. 필요하다면, 개발자는 시스템 분석가와 상의하여 평가 기준을 추가할 수 있을 것이다. 이런 평가 요소들은 비즈니스 프로세스, 서비스의 품질(Quality of Services), 전체 시스템과의 적절성 등에 대한 평가 기준이 될 수 있다.

비즈니스 프로세스 가치(Business Process Values)는 하나의 단위 서비스에 대해, 비즈니스 프로세스 상에서의 효용 가치를 추정하기 위한 평가 기준으로서, 본 논문에서는 다음과 같은 네 가지 AC를 정의하였다. 이들 AC들은 시스템의 목적에 따라 평가자와 개발자등스테이크홀더들을 통해 수정 또는 추가 될 수 있을 것

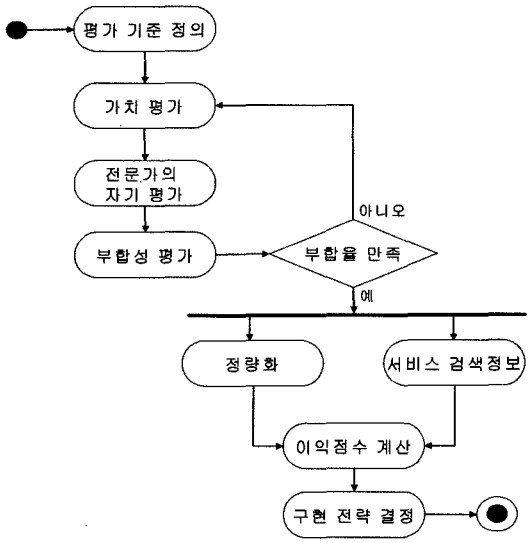

그림 7 구현 전략 결정을 위한 서비스 평가 과정

이다.

- 서비스의 중요도(Importance) : 이 평가 기준은 비즈니스 도메인 내에서 그 단위 서비스가 얼마나 중요한 역할을 담당하는지를 나타낸다.
- 서비스 업무의 영속성(Persistency) : 이 기준은 그 단위 서비스가 얼마나 오랫동안 그 시스템에서 지속적으로 사용될 가능성이 있는지를 나타낸다.
- 위험도(Risks) : 이 평가 기준은 그 단위 서비스가 원활하게 제공되지 않을 경우의 위험 정도를 나타낸다. 서비스 제공에 실패할 경우, 전체 시스템에 미치는 영향력을 평가하는 기준이다.
- 사용빈도(Frequency of Usage, 이하 FOU) : 이것은 그 서비스가 제공하는 기능이 얼마나 많이 사용되었는지를 표현한다. 서비스가 더 빈번하게 사용되어질수록, 비즈니스적 가치는 높아지게 된다.

4.2 가치 평가(Rating)

스테이크 홀더들은 각 서비스에 대해 AC들을 기준으로 다섯 가지 스케일(LL, L, M, H, HH)의 점수를 할당한다. 이때 각 AC들에 가중치를 부여할 수 있으며, 이들 가중치의 총합은 100이 되도록 한다.

아래의 표 2는 비즈니스 가치에 대한 평가 결과에 대한 예를 보여준다.

괄호에 부여된 숫자는 각 AC에 대한 가중치로서, 시스템의 중요한 목표(Goal)에 따라 달라질 수 있다. 이들 AC에 대하여 가중치를 할당하는 이유는 스테이크 홀더들이 소프트웨어 시스템의 목표에 대한 중요성을 인식할 수 있도록 우선 나열된 AC들에 대한 아웃라인을 제공하기 위한 것이다.

표 2 비즈니스적 가치에 대해 유도된 정보들

	중요도(20)	영속성(40)	위험도(20)	사용빈도(20)
Stk 1	HH	H	M	H
Stk 2	H	H	LL	HH
Stk 3	H	M	L	M
Stk 4	LL	H	M	H
Stk 5	M	M	L	H
Stk 6	H	HH	L	HH
Stk 7	H	H	L	H
Stk 8	M	H	LL	H
Stk 9	M	M	L	HH
Stk 10	H	M	M	M
Stk 11	M	H	H	L

4.2.1 자기 평가 과정

스테이크 홀더들은 일반적으로 같은 조직 내에 속한 전문가들로 구성되며, 따라서 시스템의 목표를 이해하는 정도가 유사하다. 종종 각자 다른 의견을 가지기도 하지만 의견에 큰 차이를 보이는 경우는 드물다. 의견에 차이를 보이는 경우는 유도된 결과 값이 매우 높은 다양성을 갖게 된다. 이러한 다양성은 시스템 내에 잠재하는 불확실성이나 또는 스테이크홀더들이 평가 기준의 정의에 대해 완전하게 동의하지 않는 경우에 나타날 수 있다.

유도된 값의 다양성을 최소화하기 위해서, 각 전문가들에 대한 자기평가 방법을 사용할 수 있다. 이것은 각 AC에 대한 의미를 이해해 보는 것으로서, 네 가지 스케일(전혀 이해 못함, 이해 못함, 약간 이해함, 잘 이해하고 있음)로 전문가 자신을 스스로가 얼마나 이해하고 있는지를 판단해 보는 것이다. 이들 스케일과 함께, 각각의 AC에 대한 코멘트를 첨부할 수 있도록 한다. 그리고 나면 모든 전문가들에게 평가 결과에 대한 범위(Range)와 코멘트 정보를 제공하고, 이를 통해 자신들의 평가 결과에 대한 재검토를 하도록 유도한다. 이런 과정을 반복적으로 적용함으로써 AC들에 대한 최종 결과 값을 얻어낼 수 있다.

이들 절차의 목적은 전문가들이 AC에 대한 이해의 부족에 기인하여 잘못 평가하게 되는 오류를 줄일 수 있는 기회를 제공하기 위한 것이다.

4.2.2 부합성 평가 과정

전문가들의 자기 평가 과정을 통해, AC에 대한 이해를 돕는다면, 부합성 평가 과정은 각 전문가들이 AC 각각의 중요도에 대해 얼마나 일치하는 의견을 보이고 있는지를 확인하는 과정이다. 전체 스테이크 홀더들이 얼마나 많이 AC들의 중요도에 대해 동의하고 있는지를 시험하기 위하여, 켄달의 부합상관계수를 사용할 수 있다. 이 상관계수는 전체 그룹 중에 부합하는 쌍의 비율을 계산하는 척도로서 그룹 내에서 AC의 중요도 순위에 대해 동의를 많이 할수록, 부합계수의 값이 높아진다.

$$R=\frac{(P-Q)}{(P+Q)} \quad (1)$$

식 (1)에서 R은 상관계수, P는 부합하는 관측 쌍의 수, Q는 비부합 관측 쌍의 수이다. 여기서 부합은 n개의 변량 관측 값 중에서 임의의 두 관측치가 부합하는 쌍의 수를 말한다. 예를 들어, stk1의 평가 결과 치는 (HH, H, M, H)이고 stk2와 stk3의 관측치는 각각 (H, H, LL, HH)와 (H, M, L, M)이라고 했을 때, stk1의 중요도 순위는 (1, 2, 3, 2)이고 stk2와 stk3는 각각 (2, 2, 3, 1)와 (1, 2, 3, 2)로 중요도를 부여했다. 이 경우 stk1과 stk3의 AC 중요도 순위는 일치하므로 부합하는 쌍이 되며, stk1과 stk2은 비부합쌍이 된다.

즉, 모든 관찰 치들의 가능한 조합으로부터 관측치 쌍의 값들을 비교해서 얻은 수 P와 Q를 구하여 부합적 경향과 비부합적 경향의 상대적인 차이를 정량화 한 값이다.

이렇게 P와 Q가 구해지면, 다음 식 (2)와 같이 부합율을 계산할 수 있다.

$$\frac{P}{(P+Q)}=\frac{(1+R)}{2} \quad (2)$$

식 (2)의 결과를 통해, 얼마나 많은 전문가들이 시스템의 목적에 따라 가장 중요한 AC들이 어떤 것인지에 대해 동의하고 있는지를 알 수 있다.

표 2로부터 계산된 R은 0.35이며, 부합율은 0.675 즉, 67.5%이다.

전체 평가자의 부합율 목표를 정한다면, 가치 평가 과정의 반복에 대한 기준이 될 수 있다.

4.3 서비스 저장소 검색

식별된 단위 서비스에 대한 구현전략을 결정하기 위해서는 서비스를 요청할 경우, 직접 개발할 경우 또는 기존의 컴포넌트를 래핑하여 서비스화 할 경우에 대한 비교가 필요하다. 즉, 어떤 전략이 가장 비용 효과적인 전략이 될지에 대한 비교가 이루어져야 한다. 비교를 통해 평가자들은 각 구현전약에 따른 효용성을 고려하게 되고, 그 결과로서 각 구현 전략에 대한 이익 점수가 계산될 것이다.

각 구현전략에 대한 비교를 위해서는 평가자들에게 식별된 단위 서비스 각각에 대해, 다음과 같은 정보가 제공되어야 한다.

• 단위 서비스 개발에 필요한 추정된 비용 정보

식별된 단위 서비스를 실제 개발하게 될 경우 필요한 비용을 말한다. 이는 COCOMO 또는 function point와 같은 기존의 비용 산정 기법을 사용하여 추정할 수 있다. 추정된 비용에 대한 정보가 평가자들에게 제공되어야 한다.

• 서비스 저장소에 등록되어 있는 사용가능한 서비스 목록

식별된 단위 서비스에 대한 사용 가능한 서비스에 대한 목록이 만들어져야 한다. 이러한 정보를 제공하기 위해서는 서비스 저장소는 서비스 제공자가 등록한 서비스에 대한 설명 정보(descriptions)를 저장하고 있으므로, 서비스 저장소 안에 보관되어 있는 서비스 명세서의 집합에서 식별된 단위 서비스가 구현되어 등록되어 있는지를 검색해야 한다.

• 서비스 명세 및 비용 정보를 포함하는 서비스 프로파일

일단 서비스 검색을 통해 사용가능한 서비스 목록이 제공되면, 서비스 저장소에 등록되어 있는 서비스 설명 정보 및 비용을 포함하는 프로파일이 필요하다.

• 서비스화 할 수 있는 사용가능한 컴포넌트 목록

기업이 기존에 가진 애플리케이션을 리엔지니어링 하여 서비스 래핑을 할 수 있는 사용가능한 컴포넌트들에 대한 정보가 필요하며, 이를 위해 사용 가능한 컴포넌트 목록이 만들어져야 한다.

• 각 컴포넌트 프로파일

서비스화 할 수 있는 사용가능한 컴포넌트 목록이 만들어지면, 각 컴포넌트에 대한 보다 구체적인 정보가 필요하다.

이러한 정보들을 토대로 평가자들은 다음 단계에서 이익 점수를 계산하는 과정을 수행하게 된다. 이 과정을 통하여, 단위 서비스를 어떤 방법으로 구현하는 것이 가장 비용 효과적일지에 대한 의사 결정이 이루어지게 될 것이다.

4.4 정량화 및 이익점수 계산과정을 통한 의사 결정

일단 다양한 AC들에 대한 평가가 이루어지면, 정성적인 스케일들은 비즈니스 가치와 기술적 가치에 대한 정량적인 추정 값으로 변환시킨다. HH, H, M, L, LL에 대해 각각 5에서 1까지의 값을 부여하고, 각 AC의 가중치에 따라 값을 변환한다. 예를 들어, 첫 번째 스테이크 홀더인 stk1의 경우, 중요도에 HH, 영속성에 H, 위험도에 M, 그리고 FOU에 H를 부여했으므로, 다음과 같이 계산된다.

$$\text{중요도} = 20\times\frac{5}{5}=20 \quad \text{영속성} = 20\times\frac{4}{5}=16$$

$$\text{위험도} = 20\times\frac{3}{5}=12 \quad \text{사용빈도} = 20\times\frac{4}{5}=16$$

따라서 총 합계는 80이 된다. 다음의 표 3은 비즈니스 가치에 대해 정량화 한 결과를 보여준다.

변환된 값들과 함께 각 구현전략에 대한 이익점수를 계산하여야 한다. 이익 점수란 투자 대 구현 비용에 대한 효용성을 나타내는 개념으로서, 각 구현전략에 대하여 -1, 0, +1까지의 범위 내에서 각 전략에 대한 가중치를 부여하여 계산한다. +1은 그 전략이 그 시스템에 대해 가장 좋은 이익을 줄 가능성을 표현하고, -1은 반대의 개념을 표현한다. 표 4는 이익점수 계산을 위한 가중치를 부여한 평가지의 템플릿이다.

표 3 비즈니스 가치에 대한 변환 결과

	중요도	영속성	위험도	사용빈도	총합계
Stk 1	20	32	12	16	80
Stk 2	16	32	4	20	72
Stk 3	16	24	8	12	60
Stk 4	4	32	12	16	64
Stk 5	12	24	8	16	60
Stk 6	16	40	8	20	84
Stk 7	16	32	8	16	72
Stk 8	12	32	4	16	64
Stk 9	12	24	8	20	64
Stk 10	16	24	12	12	64
Stk 11	12	32	16	8	68

표 4 이익점수 평가를 위한 템플릿

AC / 전략	중요도 (20)	영속성 (40)	위험도 (20)	사용빈도 (20)	참고사항
개발	+1	+1	0	+1	
래핑	0	-1	+1	0	
요청	-1	0	-1	-1	

이렇게 각 구현전략에 대한 가중치를 부여하고 나면, 각 AC에 대한 이익점수를 계산한다. 이익점수는 각 스테이크 홀더가 내린 비즈니스 가치에 대하여, 구현 전략에 따른 가중치를 곱한 결과 치들의 합으로 계산된다. 예를 들면, stk1의 비즈니스 가치 평가는 (20, 32, 12, 16) 이므로, 각 구현전략에 따른 이익점수는 다음과 같이 계산된다.

개발 = (20 * 1) + (32 * 1) + (12 * 0) + (16 * 1) = 68
래핑 = (20 * 0) + (32 * (-1)) + (12 * 1) + (16 * 0) = -28
요청 = (20 * (-1)) + (32 * 0) + (12 * (-1)) + (16 * (-1))
= -48

이익점수 계산 결과에 따라, 각 스테이크 홀더의 구현전략이 결정되고, 이들 스테이크 홀더들의 전체 의견에 대한 부합율을 통해 최종적인 구현전략을 결정하게 된다. 이렇게 계산된 이익점수를 통해, 전문가들이 평가한 서비스의 비즈니스 가치와 비용을 고려한 이익이 가장 클 것으로 예측되는 전략을 선택하게 될 것이다.

이 평가 가이드라인은 비즈니스와 기술적 가치의 정량화 개념을 통해 M4WSD에서 구현 전략을 결정하는 문제를 구조화도록 도와줄 것이다. 이들 과정이 개략적인 아웃라인 정도를 제공하기는 하지만, 중요도를 추정하거나 정량적인 측정의 순서를 제공함으로써 의사결정 과정에 충분한 가이드라인의 역할을 할 수 있으리라 기

대한다.

4.5 구현전략 결정의 적용 과정

일단 정량화되어 각 식별된 서비스에 대한 구현전략 결정이 이루어지고 나면, 구현전략에 따라 다음과 같은 단계를 거치게 된다.

• 서비스를 개발하는 경우

서비스 개발을 위해서는 프로젝트 계획이 수립되어야 하며, 이 과정에서 개발비용 및 시간과 인력에 대한 추정 및 위험 분석이 이루어져야 한다. 이 경우 COCOMO와 같은 기존의 비용 산정 모델을 사용하여, 비용에 대한 추정을 수행할 수 있으며, 그 결과를 토대로 개발에 대한 타당성이 검증되어야 한다. 만약 타당성 검증에 실패한다면, 예를 들어 개발비용이 너무 많이 산정된다거나, 위험요소가 많은 경우라면, 구현전략에 대한 재평가가 이루어져야 한다.

• 서비스를 래핑하는 경우

래핑 전략은 레거시 컴포넌트를 유지하면서, 컴포넌트에 대한 접근 방법으로 새로운 아키텍처에 맞는 래퍼 컴포넌트를 생성하여 연동하게 된다. 이 경우 기존 컴포넌트에 대한 가용성(availability)과 신뢰성(reliability)에 대한 평가가 이루어져야 한다. 가용성이나 신뢰성이 낮은 컴포넌트는 서비스로 구현되었을 경우, 그 서비스 자체에 대한 가용성과 신뢰성에 문제를 야기 시킬 수 있다. 이럴 경우, 구현전략에 대한 재평가가 이루어져야 한다.

• 서비스를 요청하는 경우

서비스 제공자를 통하여 서비스를 공급받는 경우, 서비스에 대한 가용성과 신뢰성 및 사용품질은 중요한 요소가 된다. 따라서 요청 서비스에 대한 가용성과 신뢰성에 대한 평가를 수행하여야 한다. 서비스의 가용성과 신뢰성이 낮은 경우, 시스템 전체의 품질을 저하시키는 요인이 된다. 따라서 서비스에 대한 가용성 및 신뢰성이 낮은 서비스의 경우, 구현전략에 대한 재평가가 이루어져야 한다.

구현전략에 대한 재평가가 이루어지는 경우, 스테이크홀더들은 시스템의 특성에 따라 서비스 개발 전략 평가 기준 항목을 조절한다. 즉, 항목의 일부를 삭제하거나 일부 값을 변경할 수 있다. 변경된 서비스 구현 전략 평가 기준에 따라 각 서비스를 다시 평가한다. 재평가 이후에는 평가된 각 서비스에 대한 서비스 평가표를 작성한다.

4.6 기존 SOA지원 개발 방법론과의 비교 분석

M4WSD와 기존 SOA지원 개발 방법론에 대한 비교를 위해, 아래 표 5와 같이 참조, 모델링 기술, 산출물, 프로세스, 서비스 평가 항목을 이용하였다.

5. 결 론

SOA는 기업의 생산성과 유연성을 극대화 할 수 있는 새로운 IT 전략으로서, 기업 인프라의 복잡성 및 유지비용을 최소화 할 수 있는 기술을 제공함으로써 기업 내부는 물론 협력사 및 고객에 대한 표준화 된 e-비즈니스 환경을 구축할 수 있도록 한다. 특히 SOAP, XML, UDDI, WSDL 등 표준 기술을 제공하는 웹서비스 발전을 계기로 관심이 고조되고 있다.

본 논문은 SOA와 웹서비스의 정의 및 특징을 살펴보고, 웹서비스 기술 표준을 기반으로 SOA를 구현하기 위한 개발 방법론인 M4WSD를 살펴보고, 서비스 구현 전략 결정을 위한 가이드라인을 정의하였다. 이 가이드라인은 M4WSD의 다섯 번째 단계인 단위 서비스 개발 여부 판단 단계의 활동들을 지원하기 위한 것으로, 단위 서비스에 대한 개발 전략을 보다 정성적인 방법으로 통해 결정할 수 있도록 도와준다. 평가 과정은 각 스테이크홀더들이 다양한 평가 기준에 대하여 가중치와 정성적 값을 할당한다. 그 결과를 정량적인 값으로 변환하여, 각 구현전략에 대한 이익 점수를 계산하여 가장 이익이 클 것으로 예측되는 전략을 결정하도록 한다. 그러

표 5 SOA지원 개발 방법론의 비교

방법론 / 비교기준	SODA	SOAD	SOMA	SOUP	M4WSD
참조	CBD, SOA	OOAD, BPM, EA	SOA	RUP, XP	CBD, SOA
모델링기술	언급 없음	UML	언급 없음	언급 없음	UML
산출물	언급 없음	언급 없음	언급 없음	언급 없음	단계별 산출물 정의
프로세스	통합된 생명주기에 대한 언급 없음	하향식+상향식	3단계, 세부 활동 정의 없음	6단계 정의	6단계 정의
			하향식+상향식	세부 활동 정의 없음	세부 활동 정의 및 지침 제공
서비스평가	언급 없음	언급 없음	언급 없음	언급 없음	개발전략 결정을 위한 평가방법 제시

나 이 과정에서 서비스 평가에 참여한 전문가들이 부여하는 가중치에 대한 좀더 객관성을 유지할 필요가 있다. 따라서 현재 전문가 그룹의 현실적인 문제점을 완화하기 위한 방법으로 널리 알려진 AHP(Analytical Hierarchy Process)등을 도입하여 현재의 평가 과정에 대한 보완이 이루어져야 할 것이다.

현재 정형화된 개발 체계의 부재로 인하여 SOA를 도입하려는 기업들은 해결해야 할 많은 문제점을 가지고 있다. 이들 문제점 해결을 위해 제시된 M4WSD와 가이드라인들은 웹서비스 기반의 SOA 구현에 대한 세부 기술 적용 지침 및 개발 모델로서 활용될 수 있을 것으로 기대된다.

현재 M4WSD에서 제안한 서비스식별 지침과 본 논문에서 제안한 평가 지침을 적용한 프로젝트가 진행되고 있으며, 세부 진행 연구가 통합되어진 M4WSD는 SOA지원 개발 방법론으로서 TTA표준으로 제출 할 예정이다.

참 고 문 헌

[1] David Smith, "Web Services : The Next Holy Grail ?," Gartner Group SSA Research Note, 2002, 5, 1.

[2] 이경하, 이규철, "SOA(Service-Oriented Architecture)와 웹서비스," 정보과학회지, 제22권, 제10호, pp.5-10, 2004.

[3] M.P. Papazogglou et al. "Service-Oriented Computing," Communications of ACM Vol. 46, No. 10, pp.25-28, Oct. 2003.

[4] W3C Web Services WG, "Web Services Architecture," W3C Working Group Note 11 February 2004.

[5] Samir Nigam, "Service Oriented Development of Applications in Sybase Workspace," Sybase Inc. whitepaper, 2005.

[6] Kunal Mittal, "Service Oriented Unified Process (SOUP)," IBM Journal, 2005년 6월.

[7] Kunal Mittal, "Build your SOA : Maturity and methodology, Part1," IBM Journal, 2005년 5월.

[8] Ali Arsanjani, "Service-Oriented Modeling and Architecture : How to identify, specify, and realize services for your SOA," IBM developerWorks, 2004년 11월.

[9] Ash Parikh, Rajesh Pradhan and Nirav Shah, "Modeling of Web Services : A Standards-Based Approach," Software Magazine, 2004년 5월.

[10] 윤홍란, 박재년, "MDA를 적용한 웹서비스 개발 프로세스", 정보처리학회 논문지, 제12-D권, 제4호, 2005년 8월.

김 유 경

1991년 숙명여자대학교 수학과(학사). 1994년 숙명여자대학교 컴퓨터과학과(석사) 2002년 숙명여자대학교 컴퓨터과학과(박사). 2002년~2005년 숙명여자대학교 컴퓨터과학과 초빙교수. 2005년~현재 University of California Davis 컴퓨터과학과 research fellow. 관심분야는 웹서비스, SOA, MDA, S/W 품질평가

윤 홍 란

1992년 숙명여자대학교 전산학과(학사) 1995년~1997년 (주)트라이콤 솔루션 사업부. 1997년 숙명여자대학교 전산교육과(석사). 1998년~1999년 숙명여자대학교 가상교육센터 연구원. 2000년~2001년 (주)이나우테크놀로지 제품개발팀. 2001년~2003년 안양과학대학 초빙전임강사. 2006년 숙명여자대학교 컴퓨터과학과(박사). 관심분야는 웹서비스, SOA, MDA, S/W 개발방법론

박 재 년

1966년 고려대학교 졸업(학사). 1969년 고려대학교 석사. 1981년 고려대학교 박사. 1979년~1989년 전남대학교 전산통계학과 교수. 1983년~현재 숙명여자대학교 정보과학부 교수. 관심분야는 소프트웨어 공학, 시스템 개발 방법론, 웹 서비스, 품질 평가, 시뮬레이션, 서비스지향아키텍처(SOA)