

서비스 지향 아키텍처 기반의 서비스모델링방법론 개발

변성욱*, 서경애**

*투이컨설팅 상무, **투이컨설팅 수석컨설턴트

A development of service modeling methodology Based on Service Oriented Architecture

Byun, Sung-Yook, Seo, Kyung-Ae

2e Consulting, 2e Consulting

E-mail : bpm@2e.co.kr, kaseo@2e.co.kr

요 약

국내 외 많은 컨설팅 및 SI회사들이 나름대로의 SOA방법론을 개발 및 적용하고 있다. 하지만 각 회사의 방법론마다, SOA 사상 관점에 따라 접근 방향이 상이하고, 수행 방법과 태스크 또한 실제 적용하기에는 상세성이 부족하다.

이 연구에서는 국내 외 SOA 방법론을 비교 분석하고, 실제 적용된 사례를 통하여 특정 도메인에 무관한 현업 사용자 위주의 서비스 식별이라는 명제를 반영한 유저인터페이스 기반의 서비스 분석 및 모델링 방법론을 제시한다.

키워드: SOA, Component, Service Identification, Biz Component, Enterprise Architecture, Service Modeling

1. 서론

최근 들어 비즈니스 관점에서 IT 아키텍처를 구성하는 SOA가 IT 업계의 새로운 패러다임으로 부상하고 있다. 이는 SOA가 기업 IT 인프라의 복잡성 및 유지비용을 최소화하고, 기업의 생산성과 유연성을 극대화 할 수 있는 새로운 IT 전략으로 대두되고 있기 때문이다[20]. 그러나 이러한 IT전략으로서의 SOA는 어플리케이션 중심의 서비스 또는 비즈니스 프로세스 기반의 태스크 중심 서비스로는 구현이 불가능하다[17].

SOA 개발방법론의 경우 이전 객체지향 개발이나 CBD와는 다르게 개발 단위가 변화되었으며 따라서 클래스나 컴포넌트보다는 좀 더 추상화된 상위 개념의 컴포넌트와 SOA의 레이어를 고려하여 적용 되어야 한다.

SOA의 레이어의 경우 프로세스 하위의 액티비티들을 정보 중심으로 묶어주는 베이직 서비스와 이

러한 베이직 서비스의 조합인 컴포지트 서비스로 구성 되어야 한다[17]. 이러한 방법론을 적용은 서비스의 모델링 과정에 현업의 적극적인 참여가 필수적이며, 모델링의 세부적인 과정은 가이드 없이 수행하기 어려운 실정이다.

본고에서는 국내 SOA 서비스 모델링 방법론의 현황을 파악하고 현업 부서의 호응을 얻기 어려운 차세대 프로젝트 현실 속에서 참여를 유도하기 위한 전사적 아키텍처 관점의 유저인터페이스 기반 SOA 서비스 모델링 방법론을 제시하고자 한다.

본 연구는 한국정보사회진흥원 서비스지향 아키텍처 기반의 서비스모델링 방법론개발 연구과제(과제번호:xxxxxxx)의 연구비지원에 의하여 연구되었음

2. 국내의 SOA 방법론 현황

국내에서 SOA의 전자적 차원의 도입은 일부 프로젝트에서 진행 중이며, 초기 시장이기 때문에 아직 성공적으로 종료한 프로젝트를 찾기 어렵다. 타당성 검증을 위한 파일럿 형태나 단순한 시스템 연계 차원 또는 웹서비스의 차원에서 구현된 사례만 존재한다. 그리고 SI회사나, 벤더사들에서도 각자의 방법론을 제시하고 있지만, SI업체들은 <표1>와 같이 구체적인 가이드와 템플릿을 파악하기 힘들고 기존 소프트웨어 개발 방법론과의 선후관계 또한 모호하다[21][15][14]. 또한 벤더사들은 비즈니스 변화를 민첩하게 수용하기 위한 시스템 재사용 관점의 적용이 아닌 시스템간의 연계 측면 자사의 솔루션 적용을 위한 방법론을 주장하고 있다. 이는 국내의 SOA 적용에 혼란을 주고 있으며, 도입에 대한 필요성을 어렵게 하고 있다.

방법론	특징	이슈
8사	<ul style="list-style-type: none"> ●비즈니스 프로세스 위주의 서비스 식별방법 ●SOA의 공통요소를 품아 적용 ●SOA를 구현하기 위한 서비스 설계, 개발, 조합에 관한 원칙을 적용하고 있음 	<ul style="list-style-type: none"> ●프로세스 위주의 서비스 식별로 비즈니스 변화 시 재사용의 어려움 발생 ●세부적인 절차나 활동이 미흡함
L사	<ul style="list-style-type: none"> ●비즈니스 프로세스와 비즈니스 도메인 모델을 분석한 서비스 식별방법 ●하향식과 상향식을 연결하여 기존 시스템의 통합을 위한 아이디어 제시 	<ul style="list-style-type: none"> ●사용자의 측면을 고려하지 않은 서비스 식별방법
1사의 SOMA	<ul style="list-style-type: none"> ●기존 IBM의 SOAD를 참조하여 확장된 방법론 ●서비스 식별, 명세, 실현의 세 단계로 구성 ●하향식과 상향식을 통합하여 정의 	<ul style="list-style-type: none"> ●서비스 식별에 대한 구체적인 활동이 제시가 미흡함

< 표 1> 각사의 방법론 현황

3. SOA 서비스 모델링 방법론

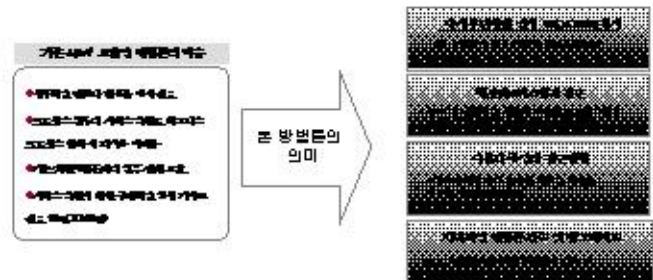
본 연구에서는 서비스에 대한 정의를 <그림1>과 같이 정의한다.



<그림 1> 서비스의 정의

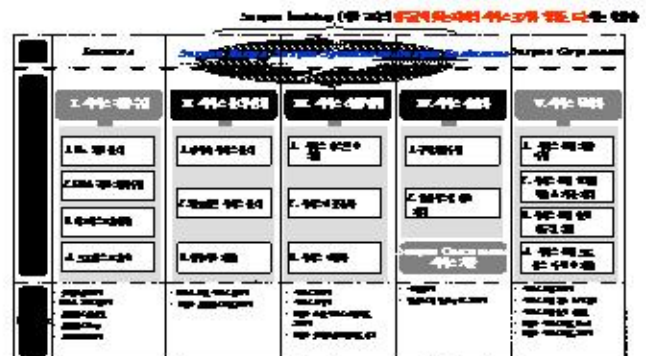
SOA에서 서비스의 범위는 비즈니스 서비스와 순수 어플리케이션 서비스를 포괄한다. 하지만 후자는 기존의 모델링 방법론으로도 쉽게 적용이 가능하므로 비즈니스 서비스에 대한 사항만을 전제로 하였다.

K(가칭)방법론은 국내 방법론의 이슈 사항을 방법론에 대한 의미로 부여 하였으며 차세대 프로젝트의 경험을 통하여 검증과 업그레이드 과정을 거쳐 실제 적용이 가능하도록 개발 되었다.



<그림 2> 방법론의 의미

방법론은 크게 Initiation, Service Design, Service Specification, Service Realization, Service Construction, Service Governance 의 단계로 나뉜다. SOA 생명주기와 서비스 생명주기의 분석[6-10,13]에 따라 서비스의 전략을 수립하고 모델링하기 위한 일련의 과정을 상세화 하여 비즈니스 관점의 SOA 프로젝트 수행을 가능하게 한다. <그림3>은 K-방법론의 전체 프레임워크이다.

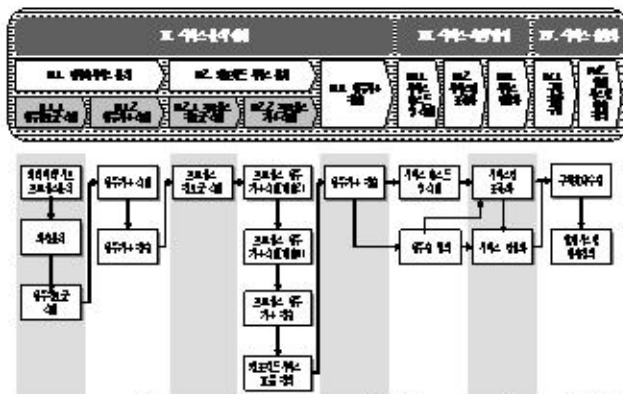


<그림 3> K 방법론 프레임워크

각 Stage 별 수행 단계는 서비스 전략 수립, 서비스 분석/설계, 서비스 실체화, 서비스 통제화이며 반복의 단계를 거치며 적용된다. 각 단계별 상세 설명은 다음과 같다.

- 서비스 전략수립: SOA 프로젝트를 시작하기 위한 도입단계로서, 현재 비즈니스 환경 분석을 통해 목표모델을 수립하고 비즈니스 프로세스를 정립함으로써 서비스 전략을 수립한다.
- 서비스 분석/설계: 수립된 전략에 따라 서비스를 식별하고, 기능을 설계하며, 설계된 기능은 구현/운영의 용이성, 기능의 유관성을 바탕으로 한 정렬의 과정을 거친다.
- 서비스 식별/명세: 식별된 서비스의 테스트 방법을 정의하고, 서비스명을 표준화하고 서비스를 명세화 한다.
- 서비스 실체화: 정의된 서비스에 대한 구현전략 수립 및 서비스를 시스템 컴포넌트로 정의한다.
- 서비스 개발: 서비스의 실체화에 따라 정렬된 서비스를 개발한다.
- 서비스 통제화: Initiation 부터 Service Construction까지의 방법론 전 단계를 통해 서비스를 통제한다.

4. SOA 서비스 모델링 프로세스 상세



<그림 4> 서비스 모델링 프로세스 상세

4.1 업무 정보군 식별

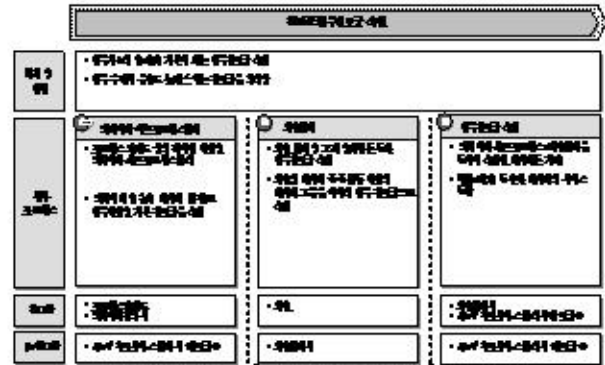
업무 정보군 식별을 위해 프로세스체계도 상의 최하위 단위인 액티비티를 분석한다. 액티비티의 입출력 데이터 분할의 과정을 거쳐 업무 정보군으

로 정의된다.

단, 액티비티중 비전산화 대상은 제외한다.

다음 화면상의 정보그룹을 업무 정보군으로 정의하며, 각 정보들은 서비스의 입출력 데이터로 식별된다.

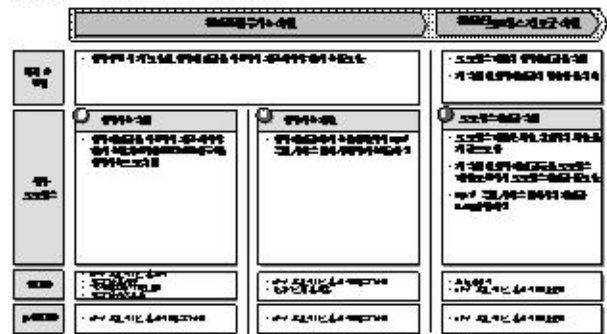
프로세스 또는 화면을 통하여 식별된 서비스 후보들 중 입출력 메시지가 전체, 또는 부분적으로 동일한 서비스에 대하여 통합을 수행한다.



<그림 5> 업무정보군 식별 태스크

4.2 업무기능의 식별

이미 식별된 서비스 후보들에 대하여 제공되어야 하는 업무기능 (조회, 수정, 생성, 삭제)을 식별한다. 이러한 기능간의 선/후행 관계와 호출의 전제조건을 정의한다. 이러한 기능들은 서비스 오퍼레이션 목록으로 정의된다.



<그림 6> 업무기능의 식별 태스크

4.3 프로세스 정보군 식별

식별된 업무 정보군을 화면 또는 프로세스 내 배치하고 이들의 선/후행 관계, 실행에 대한 전제조건을 정의한다. 화면 또는 프로세스를 레벨3의 프로세스 서비스를 정의한다. 레벨3의 프로세스 서비스는 SOA의 레이어 구성 시 최상위의 컴포지트 서비스이며 BPM이 도입된 경우 BPM과의 인터페이스를 담당한다.

프로세스 정보군 식별		프로세스 정보군 식별	
목적 범위	프로세스 정보군 식별을 위한 정보군 식별		프로세스 정보군 식별을 위한 정보군 식별
주요 프로세스	프로세스 정보군 식별	프로세스 정보군 식별	프로세스 정보군 식별
	프로세스 정보군 식별	프로세스 정보군 식별	프로세스 정보군 식별
출력	프로세스 정보군 식별	프로세스 정보군 식별	프로세스 정보군 식별
비고	프로세스 정보군 식별	프로세스 정보군 식별	프로세스 정보군 식별

<그림 7> 프로세스 정보군 식별 태스크

4.4 프로세스 기능 식별

프로세스 서비스에 배치된 업무 정보군들의 기능들을 바탕으로 프로세스 업무기능을 식별한다. 이때 서비스내의 업무 기능들 중 다른 프로세스 서비스에도 재사용이 되는 기능들을 묶어서 레벨2의 프로세스 서비스로 정의한다. 레벨2의 서비스는 레벨3의 서비스와 베이직 서비스 사이 중간 계층의 컴포지트 서비스이다.

프로세스 기능 식별				
목적 범위	프로세스 기능 식별을 위한 정보군 식별			
주요 프로세스	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별
	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별
	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별
	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별
출력	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별
비고	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별	프로세스 기능 식별

<그림 8> 프로세스 기능식별 태스크

4.5 업무기능의 정렬

서비스 통합 기준에 근거한 업무기능을 중심으로 분석된 후보서비스의 기능들을 정렬하는 과정을 거친다.

(1) 기능의 유관성[19]

동일한 혹은 유사한 기능을 제공하는 서비스 간 통합(또는 분리)을 고려한다.

- 서비스가 제공하는 비즈니스 기능이 같거나 유사한 경우 서비스들을 통합한다.
- 서비스를 요청, 제공을 위해 주고받는 메시지의 내용이 같거나 유사한 경우 서비스들을 통합한다.
- 서비스가 제공하는 비즈니스 기능을 사용자가

명시적으로 식별하기 어려운 경우, 서비스들은 통합(또는 분리)한다.

(2) 구현의 용이성

서비스 구현을 위한 기술적 용이성과 제약사항을 고려 서비스 간 통합(또는 분리)을 검토한다.

- 동일 시스템의 동일 어플리케이션, 동일 데이터베이스 인스턴스로 구현 가능한 경우 통합한다.
- 동일 프로그램 모듈에서 제공하는 오퍼레이션 또는 메소드 레벨로 구현 가능한 경우 서비스들은 통합을 고려한다(예 : 동일 세션빈(Session Bean)이 제공하는 메소드는 하나의 서비스로 통합).
- 동일 데이터 모델 또는 테이블에 접근하여 생성/삭제/변경하는 서비스들을 통합한다.

(3) 운영의 용이성

서비스의 운영/관리 용이성을 고려하여 서비스 간 통합(또는 분리)을 검토한다.

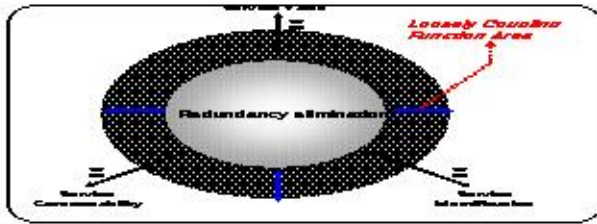
- 서로 다른 보안 사항을 적용하는 사용자에 대해서는 서비스 분리를 검토한다.
- 서비스 사용자가 요구하는 비 기능적 요구사항 성능, 장애 허용 정도, 서비스 유효 시간이 다른 경우, 서비스를 분리한다.

업무기능의 정렬	
목적 범위	서비스 간의 유사성을 평가하기 위하여 업무기능을 정렬한다
주요 프로세스	업무기능 정렬
	업무기능 정렬
출력	업무기능 정렬
비고	업무기능 정렬

<그림 9> 업무기능의 정렬 태스크

4.6 서비스의 테스트 및 식별

식별된 서비스 후보가 서비스 지향적 특성[17]을 달성하는가에 대한 검증의 과정을 거친다. <그림 10>와 같이 서비스 테스트 기준[27]를 참조하여 서비스 후보들에 대한 평가질문과 각 질문에 대한 가중치, 기준 점수를 정의한다. 현업 사용자의 서비스 후보에 대한 평가를 통하여, 기준점수를 통과한 후보서비스는 정식 서비스로 정의되며 기준 점수를 통과하지 못한 후보 서비스들은 사용자의 검증 과정을 통하여 최종 분류된다.



구분	서비스 명세서
Service Name	서비스 이름
Service Version	서비스 버전
Service Description	서비스 설명
Service Category	서비스 카테고리
Service Provider	서비스 제공자
Service Consumer	서비스 소비자
Service Interface	서비스 인터페이스
Service Data	서비스 데이터
Service Error	서비스 오류
Service Security	서비스 보안
Service Performance	서비스 성능
Service Availability	서비스 가용성
Service Reliability	서비스 신뢰성
Service Maintainability	서비스 유지보수
Service Portability	서비스 이식성
Service Interoperability	서비스 상호운용성
Service Reusability	서비스 재사용성
Service Scalability	서비스 확장성
Service Flexibility	서비스 유연성
Service Configurability	서비스 구성가능성
Service Auditability	서비스 감사가능성
Service Testability	서비스 테스트가능성
Service Deployability	서비스 배포가능성
Service Installability	서비스 설치가능성
Service Uninstallability	서비스 제거가능성
Service Upgradeability	서비스 업그레이드가능성
Service Downgradeability	서비스 다운그레이드가능성
Service Rollbackability	서비스 롤백가능성
Service Backupability	서비스 백업가능성
Service Restoreability	서비스 복구가능성
Service Migrationability	서비스 마이그레이션가능성
Service Compatibility	서비스 호환성
Service Incompatibility	서비스 호환되지 않음
Service Supportability	서비스 지원가능성
Service Trainingability	서비스 교육가능성
Service Documentationability	서비스 문서화가능성
Service Localizationability	서비스 현지화가능성
Service Internationalizationability	서비스 국제화가능성
Service Accessibility	서비스 접근가능성
Service Usability	서비스 사용가능성
Service Learnability	서비스 학습가능성
Service Interactability	서비스 상호작용가능성
Service Recoverability	서비스 복구가능성
Service Resiliency	서비스 탄력성
Service Fault Tolerance	서비스 장애내성
Service Availability	서비스 가용성
Service Reliability	서비스 신뢰성
Service Maintainability	서비스 유지보수
Service Portability	서비스 이식성
Service Interoperability	서비스 상호운용성
Service Reusability	서비스 재사용성
Service Scalability	서비스 확장성
Service Flexibility	서비스 유연성
Service Configurability	서비스 구성가능성
Service Auditability	서비스 감사가능성
Service Testability	서비스 테스트가능성
Service Deployability	서비스 배포가능성
Service Installability	서비스 설치가능성
Service Uninstallability	서비스 제거가능성
Service Upgradeability	서비스 업그레이드가능성
Service Downgradeability	서비스 다운그레이드가능성
Service Rollbackability	서비스 롤백가능성
Service Backupability	서비스 백업가능성
Service Restoreability	서비스 복구가능성
Service Migrationability	서비스 마이그레이션가능성
Service Compatibility	서비스 호환성
Service Incompatibility	서비스 호환되지 않음
Service Supportability	서비스 지원가능성
Service Trainingability	서비스 교육가능성
Service Documentationability	서비스 문서화가능성
Service Localizationability	서비스 현지화가능성
Service Internationalizationability	서비스 국제화가능성
Service Accessibility	서비스 접근가능성
Service Usability	서비스 사용가능성
Service Learnability	서비스 학습가능성
Service Interactability	서비스 상호작용가능성
Service Recoverability	서비스 복구가능성
Service Resiliency	서비스 탄력성
Service Fault Tolerance	서비스 장애내성

<그림 10> 서비스 테스트 기준

4.7 서비스명의 표준화 및 명세화

확정된 서비스들의 서비스명에 대한 표준화 과정을 거친다. 통합된 서비스가 제공하는 기능을 대변하는 이름으로 명명한다. 서비스의 명은 다음과 같은 기준을 만족해야 한다.

- 해당 도메인의 사용자가 기술적인 지식 없이도 서비스의 이름만 보고도 그 기능을 이해할 수 있는 수준이어야 한다.[17].
- 용어표준 산출에 의한 명사 및 오퍼레이션을 나타내는 동사를 조합하여 기능의 구성을 표현한다.[17].

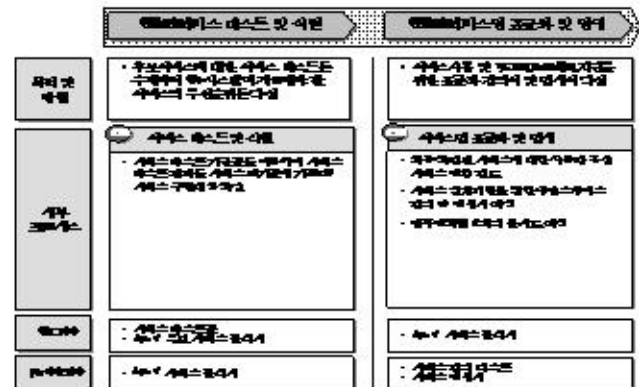
표준화된 서비스명을 기준으로 서비스정의 리스트를 작성한다. 이때 서비스의 운영을 감안하여 서비스리스트 구조를 정의한다. 이는 서비스 리포지터리로 구축 시 서비스 검색 및 이용이 쉬워진다. 서비스 명세서에는 <그림 11>과 같은 형식으로 작성한다.

- 서비스의 입출력 메시지를 정의한다.
- 서비스 품질에 대한 요구사항 즉 가용성, 접근가능성, 무결성, 성능, 신뢰성, 표준, 보안에 대한 사항을 정의한다.

구분	서비스 명세서
Service Name	서비스 이름
Service Version	서비스 버전
Service Description	서비스 설명
Service Category	서비스 카테고리
Service Provider	서비스 제공자
Service Consumer	서비스 소비자
Service Interface	서비스 인터페이스
Service Data	서비스 데이터
Service Error	서비스 오류
Service Security	서비스 보안
Service Performance	서비스 성능
Service Availability	서비스 가용성
Service Reliability	서비스 신뢰성
Service Maintainability	서비스 유지보수
Service Portability	서비스 이식성
Service Interoperability	서비스 상호운용성
Service Reusability	서비스 재사용성
Service Scalability	서비스 확장성
Service Flexibility	서비스 유연성
Service Configurability	서비스 구성가능성
Service Auditability	서비스 감사가능성
Service Testability	서비스 테스트가능성
Service Deployability	서비스 배포가능성
Service Installability	서비스 설치가능성
Service Uninstallability	서비스 제거가능성
Service Upgradeability	서비스 업그레이드가능성
Service Downgradeability	서비스 다운그레이드가능성
Service Rollbackability	서비스 롤백가능성
Service Backupability	서비스 백업가능성
Service Restoreability	서비스 복구가능성
Service Migrationability	서비스 마이그레이션가능성
Service Compatibility	서비스 호환성
Service Incompatibility	서비스 호환되지 않음
Service Supportability	서비스 지원가능성
Service Trainingability	서비스 교육가능성
Service Documentationability	서비스 문서화가능성
Service Localizationability	서비스 현지화가능성
Service Internationalizationability	서비스 국제화가능성
Service Accessibility	서비스 접근가능성
Service Usability	서비스 사용가능성
Service Learnability	서비스 학습가능성
Service Interactability	서비스 상호작용가능성
Service Recoverability	서비스 복구가능성
Service Resiliency	서비스 탄력성
Service Fault Tolerance	서비스 장애내성

구분	서비스 명세서
Service Name	서비스 이름
Service Version	서비스 버전
Service Description	서비스 설명
Service Category	서비스 카테고리
Service Provider	서비스 제공자
Service Consumer	서비스 소비자
Service Interface	서비스 인터페이스
Service Data	서비스 데이터
Service Error	서비스 오류
Service Security	서비스 보안
Service Performance	서비스 성능
Service Availability	서비스 가용성
Service Reliability	서비스 신뢰성
Service Maintainability	서비스 유지보수
Service Portability	서비스 이식성
Service Interoperability	서비스 상호운용성
Service Reusability	서비스 재사용성
Service Scalability	서비스 확장성
Service Flexibility	서비스 유연성
Service Configurability	서비스 구성가능성
Service Auditability	서비스 감사가능성
Service Testability	서비스 테스트가능성
Service Deployability	서비스 배포가능성
Service Installability	서비스 설치가능성
Service Uninstallability	서비스 제거가능성
Service Upgradeability	서비스 업그레이드가능성
Service Downgradeability	서비스 다운그레이드가능성
Service Rollbackability	서비스 롤백가능성
Service Backupability	서비스 백업가능성
Service Restoreability	서비스 복구가능성
Service Migrationability	서비스 마이그레이션가능성
Service Compatibility	서비스 호환성
Service Incompatibility	서비스 호환되지 않음
Service Supportability	서비스 지원가능성
Service Trainingability	서비스 교육가능성
Service Documentationability	서비스 문서화가능성
Service Localizationability	서비스 현지화가능성
Service Internationalizationability	서비스 국제화가능성
Service Accessibility	서비스 접근가능성
Service Usability	서비스 사용가능성
Service Learnability	서비스 학습가능성
Service Interactability	서비스 상호작용가능성
Service Recoverability	서비스 복구가능성
Service Resiliency	서비스 탄력성
Service Fault Tolerance	서비스 장애내성

<그림 11> 서비스 명세서

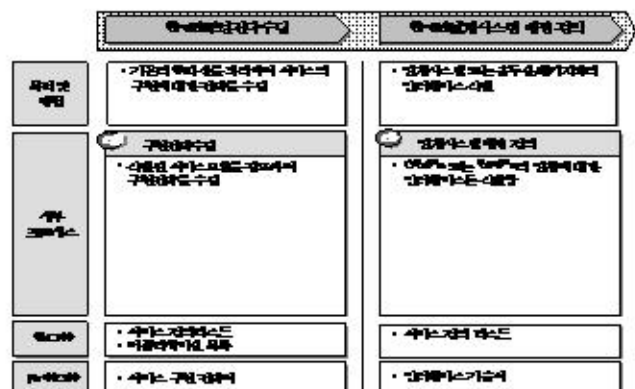


<그림 12> 서비스 식별 및 명세 태스크

4.8 서비스의 실체화

서비스정의 리스트를 참조로 구현전략을 수립한다. 서비스의 구현방법은 서비스 구현 환경에 어떤 아키텍처를 선택했는지와 기존의 어플리케이션 자산 현황을 고려하여 선정된다. 구현전략으로는 신규개발, 기존시스템 이용, 기존시스템의 변형, 솔루션의 구입 가운데서 선택할 수 있다.

서비스들은 UML 다이어그램으로 실체화 과정을 거쳐 최종 서비스 컴포넌트로 도출된다. 또한 서비스와 연계되어야 하는 시스템에 대한 인터페이스 사양을 정의한다.



<그림 13> 서비스의 실체화 태스크

5. 결론

서비스의 재사용은 자산을 구축하는 사람보다 자산을 사용하는 사람의 입장에서 생각해야 한다. 즉 개발자가 아닌 사용자의 시각으로 서비스가 모델링 되어야 한다는 의미이다.

K방법론은 (1)유저인터페이스를 기반으로 한 서비스의 식별이라는 특징과 (2)상세한 가이드와 템플릿을 제공 (3)사용자 관점의 용어 사용이라는 특징으로 사용자측면의 SOA 서비스 모델링 방법론이다.

K방법론을 전사 규모의 차세대 SOA 프로젝트에 적용한 결과 현업의 IT에 대한 이해를 도왔으며, 이에 적극적인 참여까지 유도할 수 있었다. 이는 아직 진행 중인 프로젝트의 성공의 요인으로 작용될 것이다. K방법론은 국내 SOA의 진정한 가치 실현을 통한 비즈니스 민첩성 확보에 촉매제의 역할을 할 것으로 기대한다.

[참고문헌]

- [1] IBM, "Redbook-Implementation Self-Servie in an S OA environment"
- [2] IBM, "Redbook-Integrating Enterprise Service Bus in a Service-Oriented Architecture"
- [3] IBM, "Redbook-Extended Enterprise SOA and Web Services"
- [4] IBM, "Redbooks paper- Using Business Service Choreography in Conjunction With An Enterprise Service Bus"
- [5] IBM, "Redbooks-Implementing a SOA Using an Nt erprise Servie Bus"
- [6] Bearing Point, "SOA Practitioners' Guide Part 1: Why Services-Oriented Architecture?"
- [7] Bearing Point, "SOA Practitioners' Guide Part 2: S OA Reference Architecture"
- [8] Bearing Point, "SOA Practitioners' Guide Part 3: In troduction to Services Lifecycle"
- [9] IBM, "Enabling SOA using WebShpere Messaging"
- [10] OASIS, "A methodology for service architectures"
- [11] OASIS, "Reference Model for Service Oriented Ar chitecture 1.0", <http://www.oasis-open.org/>
- [12] IBM, "Redbook-Patterns: SOA Foundation-Busin ess Process Management Scenario"
- [13] http://en.wikipedia.org/wiki/SOA_Lifecycle
- [14] [한국생산성본부교육]SOA기반의_기업IT혁신전략(SOA의_적용_모델5.1)
- [15] IBM, <http://www.ibm.com/developerworks/webser vices/library/ws-soa-term3>
- [16] <http://ruudsteeghs.blogspot.com/2007/04/rup-for-soa.html>
- [17] Thomas Erl, Service-Oriented Architecture: Concept, Technology, and Design, Prentice Hall PTR, August 2,2005
- [18] Dirk Krafsig, Karl Banke, and Dirk Slama, Enterprise SOA: Service-Oriented Architecture Best Practices, Prentice Hall PTR, November 9, 2004
- [19] Eric A. Marks and Michael Bell, Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology, Wiley, 1 edition, April 28, 2006
- [20] Jason Bloomberg and Ronald Schmelzer, Service Oriented or Be Doomed! How Service Orientation Will Change Your Business, Wiley, March 10, 2006
- [21] 한국정보과학회지 2006년11월호 제24권 제 11호 한상우, etc "Service Oriented "Architecture 격용을 위한 서비스 식별 기 법"
- [22] "Component-Based Development Helps Enable SODA", Gartner Research, 16 December 2002
- [23] <http://www-128.ibm.com/developerworks/webser vices/library/ws-soa-design1/>
- [24] Sandy Carter, The New Language of Business: SOA & Web 2.0, IBM Press, 1 edition, February 20, 2007
- [25] Paul C. Brown, Succeeding with SOA: Realizing Business Value Through Total Architecture, Addison-Wesley Professional, April 24, 2007
- [26] Steve, Jones, SEnterprise SOA Adoption Strategies, Lulu.com, November 9, 2006
- [27] http://www.ibm.com/developerworks/websphere/techjournal/0706_col_simmons/0706_col_simmons.html