

서비스 기반 소프트웨어 개발을 지원하는 4D 프로세스

(A 4D Process for Service Oriented Software Development)

박준석[†] 문미경^{**} 남태우[†] 염근혁^{***}
(JoonSeok Park) (Mikyeong Moon) (TaeWoo Nam) (Keunhyuk Yeom)

요약 최근에 제시된 서비스 기반 컴퓨팅 패러다임에서의 서비스 기반 소프트웨어 개발을 지원하기 위한 체계적인 프로세스 모델에 대한 요구가 대두되고 있다. 기존에 제시된 Unified Software Development Process, UML Components 등의 방법론은 컴포넌트에 초점을 두고 만들어진 방법으로, 서비스의 정의, 바인딩, BPEL(Business Process Execution Language)을 이용한 서비스의 조합등과 같은 서비스 기반의 개발이 가지고 있는 특성을 반영하지 못하고 있다. 또한 서비스 기반의 개발 방법론으로 제시된 SOUP(Service Oriented Unified Process), SOMA(Service Oriented Modeling and Architecture) 등은 개념적인 절차가 위주로 되어 있지, 구체적인 가이드라인, 산출물 및 접근 방법을 명확히 제시하지 않는다. 그러므로 본 논문에서는 서비스 기반의 개발 패러다임을 지원하기 위한 체계화되고, 간소화된 프로세스 모델을 제시한다. 또한 OMG에서 제시한 SPEM(Software Process Engineering MetaModel)을 적용하여 프로세스 모델을 명시적으로 표현한다. 본 논문에서 제안된 프로세스 모델을 실제 서비스 기반의 개발에 적용함으로써 서비스 기반 소프트웨어 개발의 체계성 및 효율성을 증대시킬 수 있게 된다.

키워드 : 서비스 기반 컴퓨팅, 서비스 기반 소프트웨어 개발, 서비스 기반 프로세스

Abstract Recently, Service-oriented computing is the emerging computing paradigm. In this paradigm, we require the practical process model to support service-oriented software development. The well-known development methods e.g., Unified Software Development Process, UML components have been proposed focused on component. So, these methods cannot support service-oriented computing concepts such as service definition, binding and composition concepts using Business Process Execution Language (BPEL). Also, a few proposed service-oriented approach, for example Service Oriented Unified process (SOUP), and Service Oriented Modeling and Architecture (SOMA) have appeared. However, these approaches do not explicitly represent detailed guideline, artifacts and approach. Therefore, in this paper we propose a practical and simple process model to support service oriented software development. Also, we explicitly represent process model and artifact using Software Process Engineering Metamodel (SPEM) which is proposed by OMG. By using our approach, it can enhance systematization and effectiveness for service-oriented software development.

Key words : Service Oriented Computing, Service Oriented Software Development, Service Oriented Process

· 이 논문 또는 저서는 2008년 교육과학기술부로부터 지원받아 수행된 연구임(지역거점연구단육성사업/차세대물류IT기술연구사업단)

논문접수 : 2007년 11월 1일

심사완료 : 2008년 10월 6일

† 학생회원 : 부산대학교 컴퓨터공학과
pis50@pusan.ac.kr
kaluas@pusan.ac.kr

** 정회원 : 동서대학교 컴퓨터정보공학부 전임강사
mkmoon@pusan.ac.kr

*** 종신회원 : 부산대학교 컴퓨터공학과 교수
yeom@pusan.ac.kr
(Corresponding author임)

Copyright © 2008 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제35권 제11호(2008.11)

1. 서론

컴포넌트 기반의 개발 패러다임이 서비스를 설계 및 조합하여 소프트웨어를 개발하는 서비스 지향 컴퓨팅[1]이라는 패러다임으로 변화하고 있다. 현재 UML Components[2], Unified Software Development Process [3]등 제시된 방법론들은 컴포넌트를 단위로 제시된 방법론으로 서비스 기반의 개발에 적용하기 위해서는 서비스 기반의 산출 요소들을 기반으로 재 정의가 필요하다. 또한 SOMA(Service Oriented Modeling and Architecture)[4], SODA(Service Oriented Development of Applications)[5], SOAD(Service Oriented Analysis and Design)[6] 등의 서비스 기반 개발 방법론들이 제시되었지만, 개념적인 절차나 원칙들이 제시되어있지, 구체적인 하부 행위 및 가이드라인, 산출물의 형태 및 접근 방법을 명시적으로는 제시하지는 않고 있다. 또한 서비스 기반의 패러다임은 약결합(loosely coupled)을 가지는 자치적인 재사용 가능한 서비스를 기본 단위로 소프트웨어를 개발한다는 점 및 WSDL(Web Service Description Language)[7]이라는 표준적인 인터페이스를 가지는 서비스를 조합하여 소프트웨어를 구성하게 된다는 특징이 존재한다. 따라서 본 논문은 이러한 서비스 기반의 개발이 가지는 특징을 반영하고, 프로세스와 산출물들을 명시적으로 표현한 간소화된 체계화된 프로세스 모델을 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 기반 연구를 제시하며, 3장에서는 본 논문에서 제시하는 4D 프로세스를 개관하고, 4장에서 7장은 4D 프로세스의 각 단계를 소개한다. 8장에서는 비교평가를 제시하고, 9장에서는 결론 및 향후 연구 방향을 제시한다.

2. 기반연구

2.1 SPEM (Software Engineering Process Meta Model)

OMG(Object Management Group)에서 제시한 SPEM [8] 명세는 프로세스를 표현할 때의 다양한 용어, 개념, 관계를 기술한 메타모델이 정의되어 있다. SPEM은 기존에 제시된 RUP(Rational Unified Process) 등 다양한 프로세스를 명시적으로 표현 및 기술하는 모델로 제시되었다. 따라서 본 논문에서는 4D프로세스 모델을 이러한 SPEM에서 제시된 요소들을 기반으로 하여 명시적으로 표현할 수 있도록 정의하였다.

2.2 Business Process

연관된, 비즈니스 행위들로 구성되고, 특정 입력에 대해 연속적인 비즈니스 행위들의 수행을 통해 결과를 도출하는 과정을 의미하는 비즈니스 프로세스[9]는 비즈니스

스와 IT의 간극을 줄이고, 좀 더 기민하고 유연한 시스템 개발을 가능하게 하는 구조인 서비스 지향에서는 중요한 요소로서 자리하고 있다. 그러나 기존의 애플리케이션 개발 방식은 하위 레벨인 IT 관점에 초점이 된 유즈케이스에 기반한 접근법이 주가 되었다. 따라서 제안한 프로세스 모델은 유즈케이스 중심적인 접근법이 아닌 상위단계의 비즈니스 프로세스를 시작점으로 하는 비즈니스 프로세스 중심적인 접근법을 기반으로 하였다.

2.3 서비스 기반 개발 방법론

서비스 기반 개발 방법론으로 SOUP[10], SOAD, SODA, SOMA, 플라스틱[11] 등의 방법론이 존재한다. SOUP(Service Oriented Unified Process)는 RUP와 XP의 요소들을 이용한 소프트웨어 개발 방법으로, 시작, 정의, 설계, 구성, 배치, 지원 이라는 6단계를 나누고, 단계별 활동을 정의하고 있다. 플라스틱은 UML을 기반으로 한 모델의 코드로의 변환에 초점을 맞춘 방식으로 자바라는 특정 환경에 맞춘 서비스를 개발하는 방법이다. SOUP나 플라스틱은 비즈니스 프로세스, 서비스 조합 이라는 서비스 기반 개발이 가질 수 있는 특징을 프로세스에서 보여주지 못하고 있다. SODA는 BPM (Business Process Modeling), EA(Enterprise Architecture), OOAD(Object-Oriented Analysis and Design) 등의 기존의 모델링 개념들을 SOA에 적용한 방법론으로 BPM 개념을 적용하여 비즈니스 프로세스를 기반으로 한 접근법을 가지고 있지만 서비스 단위나 세부절차에 대해 구체적으로 명시하지 않고 있다[12]. SOMA는 서비스 식별, 서비스 명세, 서비스 구체화의 세 단계로 제시되어 있다. 이 방법은 서비스를 식별하고 분류, 그리고 서비스를 조합하는 구체적인 활동들에 대해서는 제시하지 못하고 있다. Gartner에서 제시된 SOAD는 SOA를 구현하기 위한 서비스 설계, 개발에 대한 원칙들만을 제시한 방법론이다.

3. 4D 프로세스 개관

본 논문에서는 정의(Definition), 설계(Design), 개발(Development), 배치(Deploy)라는 4단계로 정의된 4D 프로세스 모델을 제시한다. 제안한 4D프로세스 메타모델을 정의하는데 있어서 SPEM 에서 정의된 요소들을 활용하여 그림 1에 제시되어 있는 것과 같이 단계(Phase), 목적(Goal), 작업산출물(WorkProduct), 활동(Activity), 단위활동(Step)으로 구성이 된 4D 메타 모델을 정의하였다. 4D프로세스의 각 단계는 하나의 목적을 가지고 있으며, 단계는 활동으로 구성이 되며, 활동은 단위활동으로 구성이 된다. 하나의 활동은 작업산출물을 이용 또는 산출한다.

4D 프로세스 메타모델이 가지는 의미 및 프로세스에

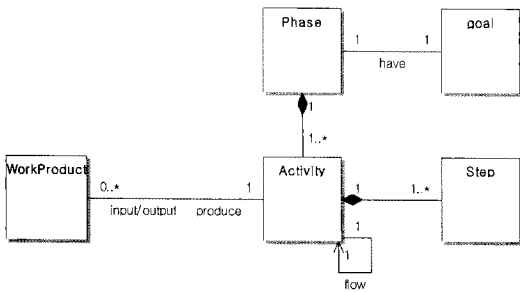


그림 1 4D프로세스 메타 모델

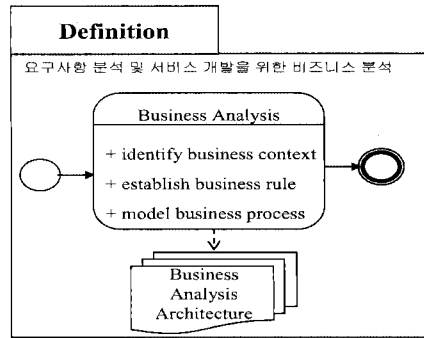


그림 2 정의 단계

표 1 4D 프로세스 구성 요소 및 표기법

구성 요소	의미	표기법
단계 (Phase)	프로세스 단계	Phase name goal
활동 (Activity)	프로세스 단계의 주 활동	Activity +step
목적 (Goal)	단계의 목적	
단위 활동 (Step)	활동을 수행하는데 사용되는 단위 활동	
작업 (workProduct)	산출물 활동에 생산되는 물리적인 산출물	활동되거나 정보 혹은 WorkProduct

사용되는 표기법은 표 1에 정의되어 있다. 제안된 프로세스 모델에 따라 4장부터 7장까지 각 단계별 설명 및 사례 적용으로 LIT(Logistic Information Technology) 연구센터[13]에서 개발한 RFID 기술기반으로 상품의 배치, 상태 등을 체크하는 스마트 선반(Smart-shelf) 시스템의 예를 사용한다.

4. 정의 단계

4D 프로세스의 정의 단계는 요구사항 분석 및 서비스 개발을 위한 비즈니스 분석을 목적으로 한다.

비즈니스 분석 활동은 비즈니스 문맥 식별, 비즈니스 규칙 정의, 비즈니스 프로세스 모델 생성이라는 하위의 단위 활동을 가진다. 이러한 단위 활동을 통하여 4W 분석서, 비즈니스 를 정의서, 비즈니스 프로세스 모델로 구성되는 비즈니스 분석 아키텍처를 생성한다. 비즈니스 문맥 식별 단계에서는 요구사항 문서와 비즈니스에 대

한 지식을 바탕으로 비즈니스 개체(Who), 비즈니스 개체의 행위(What), 비즈니스 개체의 행위가 일어나는 장소(Where), 비즈니스 개체의 행위가 발생하는 시간 정보 및 스케줄 정보(When) 측면에서 비즈니스를 분석하여 도출되는 결과물인 4W 분석서를 생성한다. 비즈니스 규칙 정의 단계에서는 4W 분석서의 When 항목에서 비즈니스 규칙으로 도출할 수 있는 것을 식별하고, 비즈니스에 대한 지식 및 시스템 구성에 참여하는 Stakeholder 들과의 협의를 통해 비즈니스 규칙을 정의한다. 비즈니스 프로세스 모델링 단계에서는 UML의 표준 다이어그램 중 하나인 액티비티 다이어그램을 기반으로 정의된 비즈니스 규칙을 적용하여 비즈니스 프로세스를 모델링한다.

스마트 선반 시스템의 비즈니스 분석 활동을 통해 산출된 스마트 선반 4W 분석서, 스마트 선반 비즈니스를 정의서의 예시는 표 2와 같다.

5. 설계 단계

설계 단계는 서비스 분석 및 설계를 목적으로 한다. 설계 단계는 서비스 추출, 논리적 서비스 모델링, 물리적 서비스 모델링, 서비스 조합 모델링이라는 활동으로 구성된다. 프로세스 메타모델에 따라 설계 단계를 도식화하면 그림 3과 같다.

5.1 서비스 추출

서비스 추출 활동의 단위 활동은 후보 서비스의 도출, 후보 서비스의 그룹화 및 우선순위로 구성이 된다. 후

표 2 스마트 선반 4W 분석서 및 비즈니스 규칙 정의서

Who	What	Where	When
스마트 선반	상품의 수량을 점검	매장	주기적 [매30분]
	상품의 배치를 점검		주기적 [매 30분]
	상품 대체 이벤트를 발생		[스마트 선반 설정 수량 != 현재의 상품 수량]
스마트 선반 비즈니스 규칙 정의서			
BR01	스마트 선반은 30분 단위로 수량과 배치를 체크한다.		
BR02	스마트 선반은 24시간 단위로 상품의 유효기간을 체크한다.		

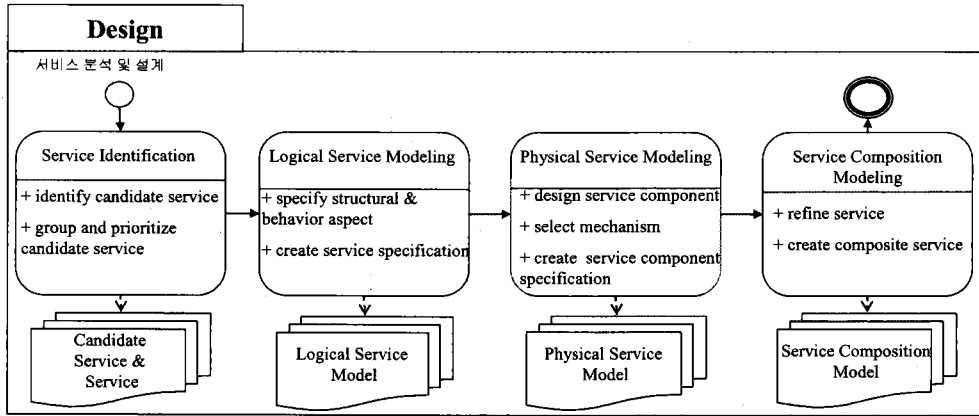


그림 3 설계 단계

보 서비스의 도출 단위 활동에서는 정의 단계에서 나온 비즈니스 프로세스로부터 1차적인 후보 서비스를 식별한다. 비즈니스 프로세스에서 독립적이거나 외부에서 활용되는 비즈니스 단위는 후보 서비스로 도출된다. 비즈니스 프로세스 상에서 분할(fork) 노드 또는 병합(Join) 노드는 서비스를 분리하는 기준점으로 활용하여 단위 활동을 병합하거나 나누어 후보 서비스를 도출한다. 또한 비즈니스 프로세스 상에서 비즈니스 단위활동간에 자동적인 전이가 가능한 부분들을 병합하여 후보 서비스로 식별한다. 후보 서비스의 그룹화 및 우선 순위화 단위 활동에서는 1차적인 후보 서비스간의 연관 관계를 바탕으로 후보 서비스에서 결합될 것은 결합하고, 서비스의 우선순위(필수적/중요도)를 고려하여 최종적인 서비스를 도출한다.

5.2 논리적 서비스 모델링

논리적 서비스 모델링 활동의 단위 활동은 정적인 구조적 측면 및 동적인 행위적 측면 명세, 서비스 명세 생성으로 구성된다. 논리적 서비스 모델링 활동에서는 서비스 추출 활동에서 나온 작업산출물인 서비스를 플랫폼 독립적인 모델로 설계한다. 구조적인 측면 및 동적인 행위적 측면 명세에서는 추출된 서비스의 인터페이스를 정의하고, 추출된 개별 서비스들의 내부 흐름을 모델링하고, 서비스간의 연결 관계를 도식화한다. 서비스 명세 생성 단위활동에서는 각 서비스의 상위 관점에 대해 서비스 이름, 역할, 입력, 출력, 전제조건, 사후 조건, 예외 사항의 항목으로 명세서를 생성한다. 그림 4는 스마트 선반 시스템에서 추출된 서비스 간의 구조적 및 동적인 행위적 측면 명세 및 *Receive error event* 서비스 명세예를 보여준다.

5.3 물리적 서비스 모델링

물리적 서비스 모델링 활동의 단위 활동은 개발 메커

니즘 선택, 서비스 컴포넌트 설계, 서비스 컴포넌트 명세로 구성된다. 물리적 서비스 모델링 활동에서는 논리적 서비스 모델링 단위활동의 작업산출물인 논리적 서비스 모델 정보를 이용하여, 플랫폼 종속적인 서비스 컴포넌트 및 서비스 컴포넌트 명세로 구성된 물리적 모델을 생성한다. 개발 메커니즘 선택 단위 활동에서는 서비스를 개발할 플랫폼 종속적인 메커니즘을 선택한다.

서비스 컴포넌트 설계 단위 활동에서는 서비스를 UML의 클래스 다이어그램을 이용하여 선택된 개발 메커니즘에 따라 설계한다. 서비스 컴포넌트 명세화 단위 활동에서 서비스를 구성하는 개별 컴포넌트에 대해서 서비스 이름, 서비스 컴포넌트 이름, 제공 및 요청하는 인터페이스, 속성, 구현메커니즘의 항목으로 기술한다. 물리적 모델링 활동에서는 실제 구현 환경에 맞추어 서비스를 설계한다. 표 3은 *Display Error Product List* 서비스를 구성하는 컴포넌트 중의 *Display Error Product List Bean* 컴포넌트 명세 및 EJB로 설계된 예를 보여준다.

5.4 서비스 조합 모델링

서비스 조합 모델링 활동의 단위 활동은 서비스 정제와 조합 모델 생성으로 구성된다. 서비스 정제 및 조합 모델 생성 활동에서는 서비스를 기반(Infrastructure): 미들웨어, 운영체에, 컨테이너 등에서 추출되는 서비스와 외부에서 지원되는 서비스를 의미, 공통(Common): 서비스 수행을 위해 공통적으로 사용되는 서비스를 의미, 비즈니스(Business): 기반 서비스나 공통 서비스를 제외한 서비스로 특정 비즈니스 지원을 위해 생성 또는 사용되는 서비스의 세 가지 형태로 구분하여 서비스간의 연결 관계를 정의하고, 나누어진 계층에 따라 서비스들을 배치하고, 서비스간의 의존성 및 연결 관계를 구성한다.

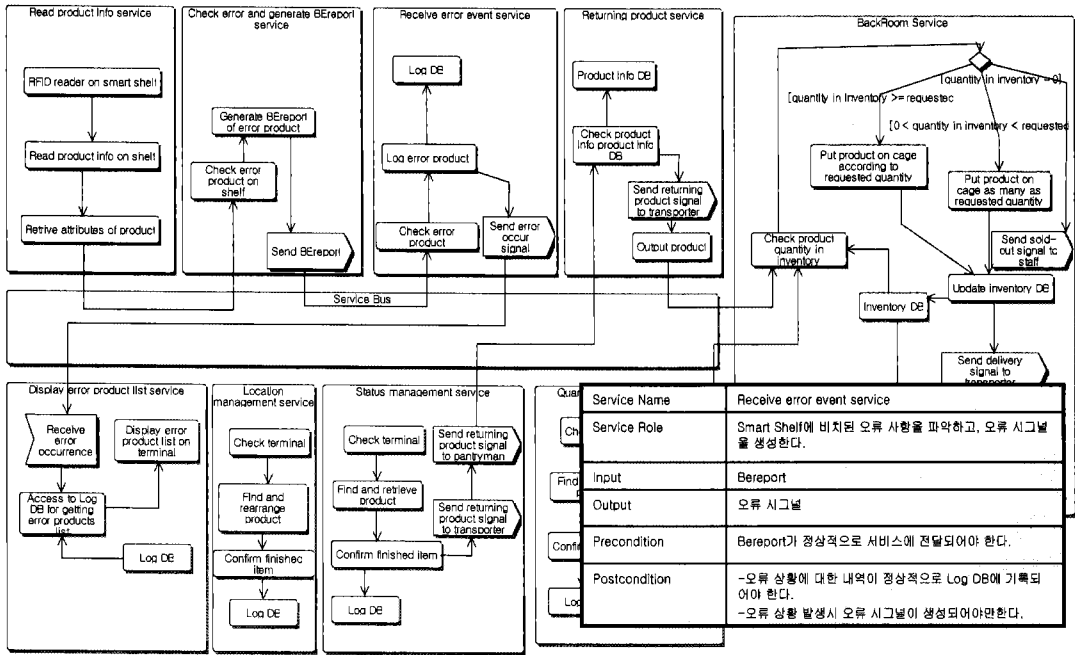


그림 4 스마트 선반 시스템 구조적 및 동적 행위 측면 명세 및 서비스 명세

표 3 Display Error Product List Service

서비스명	DisplayErrorProductListService
서비스 컴포넌트 명	DisplayErrorProductListBean
제공되는 인터페이스	LogInfo[] retrieveLogInfo(String errorType) -Input parameter : errorType(오류 타입) 1: 위치오류2: 수량오류3: 상태오류 -return type : LogInfo[]
속성	LogInfo[] 속성, int oid: 고유식별값, String errorType: 오류타입, String productName: 상품명, String productec: 상품 전자상품코드, String locshelfid: 있어야할 선반 ID, String curshelfid: 현재 위치한 선반 ID, int appQty: 적정 수량, int curQty: 현재 수량, String expDate: 유통기한, String occurTime: 오류 발생 시간
구현 메커니즘 (EJB Session Bean)	<pre> classDiagram class DisplayErrorProductListSession { +EJBContext() +EJBPostCreate() +EJBPreDestroy() +EJBSessionContext() +EJBSessionContext() +EJBRemove() } class DisplayErrorProductListBean { <<create>> +DisplayErrorProductListBean() +EJBCreate() +EJBPostCreate() +EJBPreDestroy() +EJBRemove() +EJBSessionContext() +retrieveLogInfo(String errorType, String) LogInfo } class LogInfoHandler { +LogInfoHandler() +updateInventoryInfo(ComList, String, QtyUnit, String) String +inventoryQtyUpdate(String) int } class LogInfo { +oid int +errorType String +productName String +productec String +locshelfid String +curshelfid String +appQty int +curQty int +expDate String +occurTime String +date String } DisplayErrorProductListSession --> DisplayErrorProductListBean DisplayErrorProductListBean --> LogInfoHandler DisplayErrorProductListBean --> LogInfo </pre>

6. 개발 단계

본 논문에서 제안하는 4D 모델은 서비스를 개발하고 구체화하는 기술로 웹 서비스를 기반 모델로 삼아 구현 모델로 끌어 가도록 정의하였다. 이에 따라 명세화된 개발 단계는 서비스 및 컴포넌트 구현을 목적으로 서비스 구체화와 BPEL(Business Process Execution Language)[14] 코딩이라는 활동으로 구성된다(그림 5).

서비스 구체화(Service Realization)는 서비스 컴포넌트를 개발하는 단위 활동과 WSDL을 구성하는 단위 활동으로 구성된다. 서비스 컴포넌트를 개발하는 단위 활동에서는 실제 단계의 물리적인 모델링 활동에서 산출된 서비스 컴포넌트 명세서를 기반으로 실제 서비스 컴포넌트를 개발한다. WSDL을 구성하는 단위 활동에서

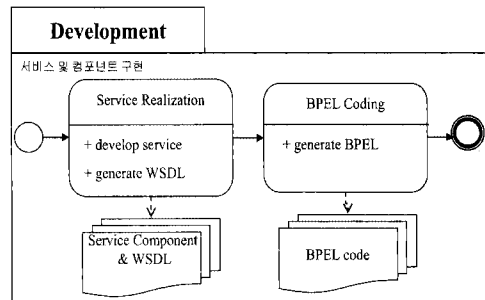


그림 5 개발 단계

는 개발된 서비스 컴포넌트를 WSDL로 명세화한다. BPEL 코딩 활동에서는 개발 단계의 서비스 조합 모델링 결과물을 바탕으로 조합된 서비스를 개발하는 단계를 진행한다.

7. 배치 단계

배치 단계는 개발 단계에서 개발된 서비스를 배치 및 운영시키는 목적을 가진다. 그림 6과 같이 배치 단계는 서비스 배치와 BPEL 배치 활동으로 구성된다. 서비스 배치 활동은 개발된 서비스 컴포넌트를 배치시키는 단위 활동과 WSDL을 배치시키는 단위 활동으로 구성된다. 이렇게 개발된 서비스 컴포넌트 및 WSDL이 배치가 됨으로써 사용자는 서비스를 호출하여 사용할 수

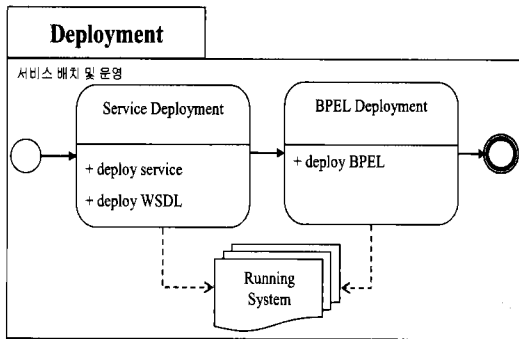


그림 6 배치 단계

있다. BPEL 배치 활동은 BPEL을 BPEL 엔진에 배치하는 단위 활동을 진행한다.

그림 7은 이러한 서비스의 배치 및 BPEL 배치 후 웹을 클라이언트로 하여 최종 개발된 웹 기반의 스마트 선반 시스템을 보여준다.

8. 비교 평가

이 장에서는 4D 프로세스 모델을 서비스 기반으로 제시된 SODA, SOAD, SOMA, SOUP, PLASTIC과 비교를 한다. 1)프로세스의 목적: 각 프로세스마다 특정 목적을 달성하기 위해 제시된다. 프로세스의 목적을 비교한다. 2)프로세스의 특징: 프로세스가 가지고 있는 특징적인 부분을 비교한다. 3)비즈니스 관점의 지원: 상위 수준의 비즈니스의 개념부터 IT의 개념을 이끌어 내는 측면을 가지고 있는지를 비교한다. 4)서비스의 속성 반영: 서비스는 자치적인 재사용 단위, 조합 가능한 단위라는 개념을 반영하는 측면을 제시하는지를 비교한다. 5)프로세스가 지원하는 생명주기: 일반적인 요구사항 분석 설계, 개발 등의 프로세스가 지원하는 생명주기 측면을 비교한다. 6)표현요소 및 산출물: 프로세스의 표현요소 및 산출물들이 명확하게 제시되어 있는지 비교한다. 표 4는 이러한 기준에 따라 비교된 내역이다.

9. 결론 및 향후 연구

서비스 기반 패러다임에서 소프트웨어 개발은 서비스

그림 7 배치된 서비스, BPEL, 구현된 스마트 선반 시스템

표 4 프로세스 비교

평가항목	서비스 기반 개발 방법론					
	SODA	SOAD	SOMA	SOUP	PLASTIC	4D
목적	애플리케이션의 서비스 기반 개발	SOA 프로젝트의 분석과 설계 단계를 지원	SOA의 근간을 형성하는 모델링/분석/설계기술 제공	조직의 SOA 프로젝트를 지원	자기적용적인 서비스를 개발	서비스 기반 시스템의 개발
특징	컴포넌트/분산시스템, SOA의 공통 분모를 기반으로 정의	BPM, EA, OOAD의 개념을 접목	하향식과 상향식 개발방법 통합	RUP 및 XP 기법을 기반	MDD의 개념 적용, 모델→코드로의 변환에 초점, 단일 서비스의 개발 과정에 초점	비즈니스 프로세스 기반의 접근법 및 서비스 모델링과 명확한 명세 기법 제시
비즈니스 관점	구체적인 세부 지원 기법을 제시하지는 않음	지원(강) -비즈니스 프로세스 모델링	명시적 모델 제시하지 않음	미지원-유즈케이스 중심	지원(약)-개념모델	지원(강) -비즈니스 분석
서비스 속성	자치적인 재사용 단위	세부적인 절차, 모델 구체적 제시 안됨	지원(약) -서비스 식별단계의 구체적 내역 제시 안함	미지원	지원(약) -서비스 기능 명세	지원(강) -논리적인 서비스 모델
	서비스 조합	세부적인 절차나 모델 구체적 제시 안됨	지원(약) -서비스분류/구분 단계를 제시하나 구체적인 하부활동 제시안함	미지원	미지원	지원(강) -서비스조합 모델
생명주기	구체적인 라이프사이클 제시 안함	분석, 설계, 개발 단계로 제시 - *개발 단계에 대한 내용 없음	서비스 식별, 명세, 서비스 구체화	시작, 정의, 설계, 구성, 배치, 지원	명세화, 변환, 코드 생성 단계	서비스 정의, 설계, 개발, 배치 단계로 구성
표현 요소 및 산출물	제시안됨	BPM, EA, OOAD 기존 제시 기법을 활용한다고만 명시	명세화단계에서 다양한 명세산출물을 제시, 식별, 구체화 단계의 산출물 형태를 명확히 제시 못함	각 단계에서 나올 산출물들만 리스트 해놓음 표현 기법 및 구체적 구성 내역 제시 안함	UML 기반	UML 요소와 템플릿 요소 제공 및 SPEM을 활용한 모델 제시

를 기본 단위로 하여 유연한 시스템을 개발할 수 있는 이점을 가진다. 본 논문에서는 이러한 개발 패러다임을 지원하는 방법에 대한 연구로 서비스의 식별, 조합의 개념등을 명시적으로 반영한 체계화되고, 간소화된 4D 프로세스 모델을 제안하였다. 먼저, 비교평가에 제시한 바와 같이 다양한 서비스 기반의 개발 접근법에 대하여 분석하였다. 또한 이러한 여러 개발 접근법들이 서비스 기반의 개발을 구체적으로 가이드하는 요소를 보여주지 못하는 부분을 보완하기 위하여 프로세스 모델에서 논리적 모델, 물리적 모델, 서비스 조합 모델 등의 다양한 템플릿을 제시하였다. 그리고 프로세스가 가지는 산출물에 대해서도 명시적으로 표현하지 못하는 점을 보완하기 위하여 SPEM을 활용하여 프로세스의 산출물을 정의하여 명시적으로 표현할 수 있도록 구조화 및 제안한 프로세스 모델을 적용하기 위한 구체적인 접근법을 제시하였다. 마지막으로 서비스 기반 스마트 선반 시스템의 구현을 통해 제안된 방법을 적용하는 사례 및 방법의 타당성을 제시하였다. 따라서 본 논문에서 제안한 체계화되고 간소화된 4D 프로세스 모델을 서비스 기반

개발에 적용함으로써 비즈니스 분석부터 시작하여 시스템 개발에 있어서 체계성 및 효율성을 증대시킬 수 있으며, 4D 프로세스의 각 모델을 통해 시스템 개발에서 산출되는 요소들을 체계적으로 표현함으로써 향후 서비스 기반의 개발 시스템에 대한 전체적인 구조파악을 손쉽게 할 수 있는 이점을 제공한다.

향후 연구방향으로는 4D 프로세스를 지원하는 지원 도구 개발 및 CMMI 등의 기법 적용 등 프로세스 확장 및 개선 연구를 진행 해 나갈 것이다.

참고 문헌

[1] M. Bichler and K. J. Lin, "Service-Oriented Computing," Computer, Vol. 39, Issue 3, pp. 99-101, Mar. 2006.
 [2] J. Cheesman, and J. Daniels, *UML Components A Simple Process for Specifying Component Based Software*, Addison-Wesley, 2000.
 [3] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*, Addison Wesley, 1998.

- [4] A. Arsanjani, "Service-Oriented Modeling and Architecture: How to identify, specify and realize services for your SOA," IBM developerWorks, Nov., 2004.
- [5] Gartner, "Service-Oriented Development: SODA and the Web Service Producer Platform," Gartner Symposium ITXPO, 2002.
- [6] O. Zimmermann, P. Krogdahl, and C. Gee, "Elements of Service-Oriented Analysis and Design," IBM developerWorks, Jun., 2004.
- [7] WEB Service Description Language, <http://www.w3.org/TR>.
- [8] OMG, "Software Process Engineering Metamodel," Version 1.1, format/05-01-06, Jan. 2005.
- [9] Business Process, <http://www.bizopt.co.uk/Methods/glossary.htm>
- [10] K. Mittal, "Build your SOA: Maturity and Methodology, Part III," <http://www.soainstitute.org/articles/article/article/build-your-soa-maturity-and-methodology-part-iii/news-browse/2.html>, Jun, 2006.
- [11] M. Autili, L. Berardinelli, V. Cortellessa, A. Di Marco, D. Di Ruscio, P. Inverardi, and M. Tivoli, "A Development Process for Self-Adapting Service Oriented Applications," Proceedings of International Conference on Service Oriented Computing, Springer, LNCS 4749, pp. 442-448, 2007.
- [12] 이현주, 최병주, 최정원, "서비스 지향 아키텍처를 위한 컴포넌트기반 시스템의 서비스 식별", 정보과학회논문지, 제35권, 제2호, pp. 70-80, 2008.
- [13] 부산대학교 차세대물류 IT기술연구사업단 제1 총괄과제, <http://www.rclit.com>
- [14] BPEL, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel



남 태 우

2007년 부산대학교 정보컴퓨터공학부(학사). 2007년 3월~현재 부산대학교 컴퓨터공학과 석사과정. 관심분야는 소프트웨어 아키텍처, 서비스 기반 소프트웨어 개발, RFID 미들웨어, 상황인식기법 등

염 근 혁

정보과학회논문지 : 소프트웨어 및 응용 제 35 권 제 8 호 참조



박 준 석

1999년 부경대학교 컴퓨터공학과(학사)
 2002년 부산대학교 컴퓨터공학과(석사)
 2002년 11월~2007년 6월 (주)사이버텔월드 부설연구소 연구원. 2002년~현재 부산대학교 컴퓨터공학과 박사과정. 관심분야는 서비스 지향 아키텍처, 소프트웨어 프로덕트 라인, 소프트웨어 아키텍처, 유비쿼터스 컴퓨팅 등

문 미 경

정보과학회논문지 : 소프트웨어 및 응용 제 35 권 제 8 호 참조