

프로덕트 라인 기반 서비스 명세 생성 방법

박 준 석[†] · 문 미 경^{**} · 염 근 혁^{***}

요 약

소프트웨어 프로덕트 라인은 일련의 관련된 시스템들, 즉 도메인 내에서 재사용될 가능성이 높은 공통된 부분들을 식별하고, 시스템마다 상이하게 나타나는 가변적 요소들의 분석을 통해 핵심 자산을 생성하는 방법이다. 이러한 프로덕트 라인 개념은 객체 및 컴포넌트 기반의 소프트웨어 개발 패러다임에 적용되어 소프트웨어 개발의 생산성을 증대시켰다. 최근 들어 소프트웨어 개발의 패러다임은 서비스라는 핵심 자산을 이용하여 소프트웨어를 개발하는 서비스 지향 패러다임으로 변화하고 있다. 따라서 이러한 패러다임하에서 비즈니스 변화에 민첩하게 대응하고, 체계적이고 유연한 서비스 지향 소프트웨어 개발을 할 수 있도록 핵심 자산인 서비스 생성 방법은 프로덕트 라인 기법을 적용시킬 필요가 있다. 본 논문에서는 서비스 지향 패러다임과 프로덕트 라인의 결합을 통해 소프트웨어 개발의 핵심 자산인 도메인 서비스를 개발하고 기술하는 방법에 대해 제시한다. 도메인 서비스는 도메인 서비스 공통적/선택적 속성 가변성, 오퍼레이션 타입의 가변성, 메시지 타입의 가변성을 통해 공통성과 가변성이 명확히 기술되도록 한다. 본 논문에서 제시하는 방법을 통해 유사한 기능을 가진 도메인 서비스를 목적에 맞게 커스터마이징하여 다양한 서비스를 생성하게 함으로써 서비스 지향 개발을 유연하고 효율적으로 수행할 수 있다.

키워드 : 서비스 지향 컴퓨팅, 프로덕트 라인, 서비스 지향 소프트웨어 개발, 서비스 가변성, 도메인 서비스

An Approach to Developing Service Descriptions Based on Product Line

JoonSeok Park[†] · Mikyeong Moon^{**} · Keunhyuk Yeom^{***}

ABSTRACT

Software product line engineering is an approach to creating core assets by systematically exploits common that can be highly reusable and different characteristics in related system families that is domain. By adapting this approach in object and component paradigm, software productivity is extended. Recently, Software development paradigm is changed to service oriented paradigm which is using services that are core assets. Therefore, to develop service oriented software more flexible, systematic and to enhance business agility about business changes, we need adaptation of product line to creation of services that are core assets. In this paper, we propose an approach to create and specify domain service through combine of product line and service oriented paradigm. Domain service explicitly specifies commonality and variability by domain service common/optional property variability, operation type variability, message type variability. By using this approach, it can support efficient and flexible service oriented development by producing various services through customization of domain service for the purpose.

Keywords : Service Oriented Computing, Product Line, Service Oriented Software Development, Service Variability, Domain Service

1. 서 론

소프트웨어 프로덕트 라인[1, 2]은 제품 군의 공통적인 특징과 차별적인 특징을 체계적으로 분석하여 핵심자산을 개발 및 이를 활용하여 소프트웨어를 개발하는 방법이다. 이러한 프로덕트 라인은 객체 및 컴포넌트 기반의 IT 중심적

인 소프트웨어 개발에 적용되어 생산성과 재사용성을 증대시키는 효과를 가져왔다. 최근 들어 비즈니스 변화에 민첩하고 유연하게 대처하기 위해 비즈니스 관점이 반영된 상위 수준의 개념인 서비스를 핵심 자산으로 생성하고 활용하는 서비스 지향 컴퓨팅(Service Oriented Computing)[3]이 새로운 개념으로 제시되고 있다. 이러한 서비스 지향 컴퓨팅과 연관된 목적을 현실화 하는 것을 지원하기 위한 기본적인 수단으로 서비스를 위치시킴으로써 효율성, 민첩성 그리고 기업의 생산성을 추구하는 아키텍처 모델[4]로서 서비스지향 아키텍처(Service Oriented Architecture:SOA)가 제시되었다. 서비스 지향 컴퓨팅 및 SOA 에서의 핵심 자산은 서비

* 이 논문은 2009년 교육과학기술부로부터 지원받아 수행된 연구임(지역적
접연구단육성사업/차세대융합기술연구사업단).
† 준 회 원 : 부산대학교 컴퓨터공학과 박사과정
** 정 회 원 : 동서대학교 컴퓨터정보공학부 전임강사
*** 정 회 원 : 부산대학교 정보컴퓨터공학부 교수(교신저자)
논문접수: 2009년 5월 27일
수정일: 1차 2009년 6월 29일
심사완료: 2009년 7월 4일

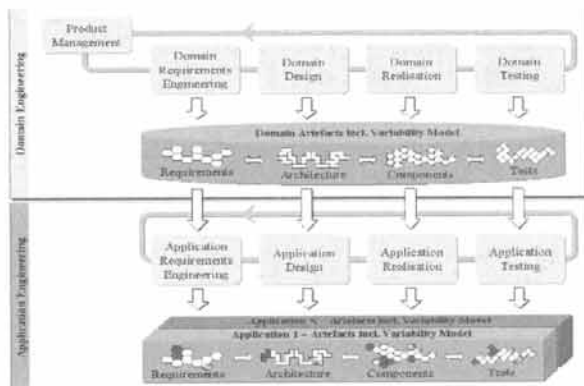
스이다. 현재 서비스에 대해서 다양한 정의가 내려지고 있다. [5]에서는 서비스를 표준화된 정의 언어로 기술되고, 공식된 인터페이스를 가지고 있고, 공통의 비즈니스 작업 또는 프로세스를 협업적으로 지원하기 위해 서로의 오퍼레이션의 실행을 요구하면서 의사소통 하는 것으로 정의하고 있다. 또한 [6]에서는 잘 정의되고, 표준적인 비즈니스 기능성을 제공하는 자기 기술적인 모듈이며, 다른 서비스의 문맥 또는 상태와는 독립적인 것이라고 기술하고 있다.

비즈니스 변화에 민첩하게 대응하고, 체계적이고 유연한 서비스 지향 소프트웨어 개발을 할 수 있도록 핵심 자산인 서비스 생성 방법으로 프로덕트 라인 기법을 적용시킬 필요가 있다. 그러므로 본 논문에서는 SOA의 핵심 자산인 서비스에 초점을 두고, 이를 개발하는 데 있어서 프로덕트 라인을 적용한 접근법을 제시한다. 도메인에서 사용할 수 있는 가변성을 가진 상위 수준의 도메인 서비스를 기술하기 위해 도메인 서비스의 공통적/선택적 속성 가변성, 오퍼레이션 타입 가변성, 메시지 타입 가변성을 제시하고 이를 명확히 기술하는 방법을 제시한다. 본 논문은 2장에서는 관련 연구를 기술하고, 3장에서는 가변성이 포함된 도메인 서비스 구성 요소를 제시한다. 4장에서는 도메인 서비스에 포함된 가변성 타입에 대해 설명하고, 5장은 본 논문에서 제시한 도메인 서비스 분석 방법을 통해 도메인 서비스 명세서가 자동 생성되는 지원도구 및 평가 내역을 기술한다. 6장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 프로덕트 라인

프로덕트 라인은 특정한 고객 또는 시장 분야에 대한 가변성을 관리하면서, 연관된 시스템의 집합에서 체계적으로 공통성을 획득하고, 개발하는 기법이다[7]. 프로덕트 라인은 핵심자산을 개발하기 위한 도메인 공학 단계(Domain Engineering)와 핵심자산을 재사용하여 애플리케이션을 개발하는 단계인 애플리케이션 공학 단계(Application Engineering)로 구성된다.



(그림 1) 프로덕트 라인 프로세스[8]

핵심자산들은 소프트웨어 개발의 각기 다른 단계에서 생산되기 때문에 표현 요소들의 추상화 수준이 다르며, 이로 인해 각 핵심자산이 가지고 있는 가변성 또한 각기 다른 수준에서 각기 다른 유형으로 나타나게 된다.

2.2 Web Service Description Language

W3C에서는 배치되어 실행될 수 있는 서비스의 표준으로 Web Service Description Language(WSDL)를 제시하였다 [9]. WSDL은 설명(description), 타입(type), 인터페이스(interface), 바인딩(binding), 서비스(service), 문서(documentation) 부분으로 구성이 된다. 설명 부분에서는 WSDL 전체에서 사용되는 네임스페이스를 선언한다. 타입 부분에서는 서비스가 사용되는 메시지를 정의한다. 인터페이스 부분에서는 인터페이스의 이름과, 오퍼레이션, 메시지의 입력, 출력 등에 대해 기술한다. 바인딩 부분에서는 구체화된 메시지 형태와 전송 프로토콜 등 메시지가 어떻게 교환될 것인가를 기술한다. 서비스 부분에서는 서비스의 이름, 인터페이스, 서비스가 접근할 수 있는 endpoint 등의 항목을 명시한다. 문서 부분에서는 서비스에 대한 부분적인 설명, 서비스가 상호작용하기 위한 기본적인 메커니즘 등이 기술된다.

2.3 SOA 기반의 가변성 분석 관련 연구

SOA 기반에서 가변성을 고려한 관련연구로 [10-12]가 제시되었다. [10]은 웹 서비스의 개발에 있어서의 재사용성에 초점을 두고 서비스의 엔드 포인트의 가변성, WSDL 기술의 가변성과 비즈니스 로직의 가변성이라는 세가지 분류를 제시하고, 패턴에 기반하여 가변성을 관리하는 기법을 제시하였다. 그러나 가변성 분류만을 제시하고 이를 명시적으로 표현하고 기술하는 것에 대해서는 다루고 있지 않다. 이 방법은 실제 수행되는 서비스에 국한하여 가변성을 찾고 있는 반면, 본 논문에서 제안한 방법은 이보다 상위의 도메인 서비스라는 개념을 정립하고 가변성을 찾아냄으로써 이를 기반으로 다양한 서비스를 파생시켜 나가는 방식을 채택하고 있다. [11]은 소프트웨어의 적용에 초점을 두고 워크플로우 가변성, 조합 가변성, 인터페이스 가변성, 로직의 가변성이라는 분류를 제시하였다. 이 접근법은 서비스가 조합되고, 처리될 수 있는 측면에서 가변성을 다룬 연구이다. 이 접근법 또한 가변성 표현에 대한 명시적이고 구체적인 방법을 제시하지는 않고 있다. [12]는 웹 서비스의 플로우에 초점을 두고 바인딩 타입 가변성, 파트너 선택 영역 가변성, 메시지교환 가변성, 프로토콜 가변성이라는 분류를 제시하였다. 여기서 제시한 방법들은 하나의 서비스가 실행되고, 호출되는 측면에서의 가변성을 다룬 연구이다. 기존의 관련 연구들은 서비스가 '실행 시' 나타나는 가변성에 초점을 두고 있다. 그러나 본 논문에서는 프로덕트 라인이 가지고 있는 개념에 기반하여 서비스 '개발 시' 도메인 단계에서의 서비스가 가질 수 있는 가변성에 초점을 두고 있다.

3. 도메인 서비스 구성 요소

(그림 2)는 도메인 서비스의 구성 요소를 나타내는 도메인 서비스 아키텍처 메타 모델이다.

3.1 기본 개념

본 논문에서 제안하는 프로덕트 라인 기반의 서비스 명세 생성 방법에 적용된 개념에 대해 아래와 같이 정의한다.

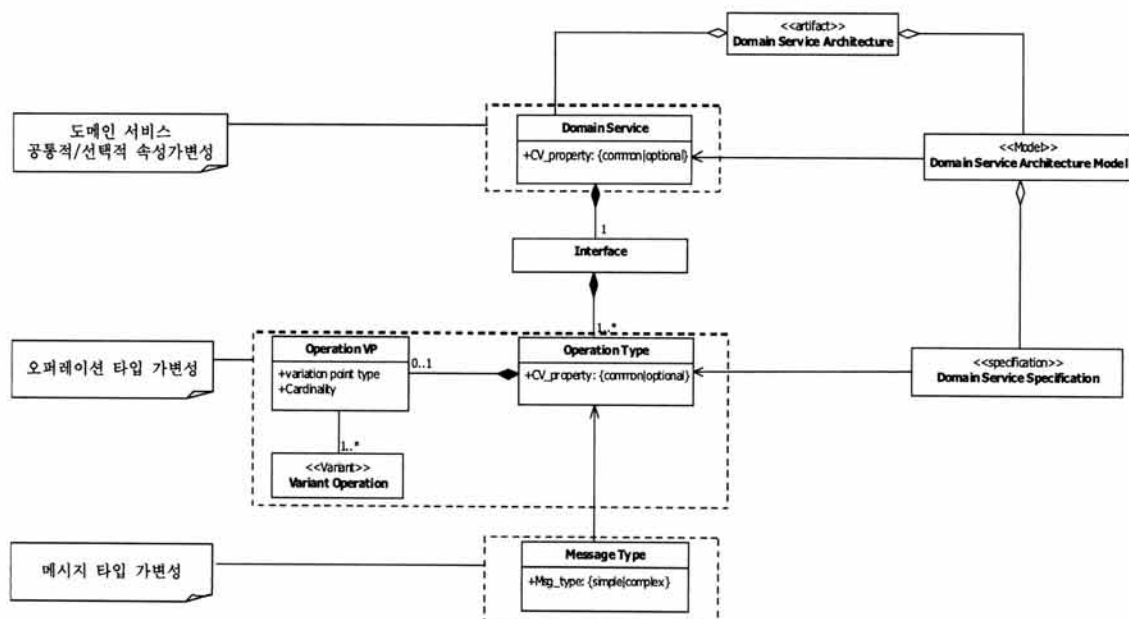
- 도메인 서비스 아키텍처: 도메인 서비스 아키텍처는 서비스 지향 환경에서 설계자간에 기록되고 전달될 수 있는 가변성을 가진 추상화된 아키텍처이다.
- 도메인 서비스: 도메인 서비스는 프로덕트 라인 개념상의 도메인 공학 단계에서 생성될 수 있는 가변성을 가진 재사용 가능한 서비스를 의미한다. 도메인 서비스는 인터페이스, 오퍼레이션 타입, 메시지 타입의 세가지 요소로 구성된다.
- CV_property: 공통적(common) 또는 선택적(optional)인 속성을 나타낸다.
- 인터페이스: 도메인 서비스는 하나의 인터페이스를 가진다. 도메인 서비스의 인터페이스를 통해 도메인 서비스를 연결 및 도메인 서비스의 기능을 사용한다. 인터페이스는 WSDL로 전이될 때 WSDL 상의 인터페이스와 매핑된다.
- 오퍼레이션 타입: 도메인 서비스가 제공하는 오퍼레이션을 일반화시킨 개념이다. 오퍼레이션 타입은 하나의 오퍼레이션에 해당되거나, 오퍼레이션들 중 유사한 기능을 가진 대체될 수 있는 오퍼레이션인 경우 이를 하나의 오퍼레이션 타입으로 일반화 시킨다.
- 오퍼레이션 VP: 도메인 서비스의 가변치 오퍼레이션이

연결 될 수 있는 가변점과 가변치 오퍼레이션의 선택 가능 수를 나타내는 바인딩 카디널리티를 오퍼레이션 VP로 정의한다.

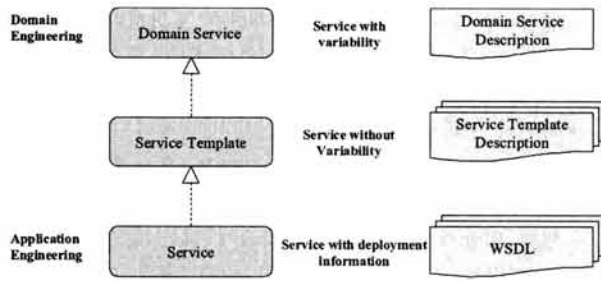
- 가변치 오퍼레이션: 공통적, 선택적 속성이 표현된 도메인 서비스의 오퍼레이션들을 가변치 오퍼레이션이라고 정의한다. 또한 가변치 오퍼레이션이 명기될 때 오퍼레이션 이름과 오퍼레이션이 가지게 되는 입력 메시지는 in 이라는 키워드를 붙여 나타내며, 출력 메시지는 out 이라는 키워드를 붙여 나타낸다.
- 메시지 타입: 도메인 서비스의 입력 및 출력으로 가변치 오퍼레이션에 사용되는 메시지가 일반화된 형태를 메시지 타입이라고 정의한다. 도메인 서비스에서 사용되는 메시지 타입은 단순(simple)과 복잡(complex)으로 구분한다. 단순 타입은 정수형, 숫자형, 불리언 형 등 일반적인 프로그램에서 활용되는 데이터 타입을 가진 메시지 형태를 정의한다. 복잡은 이러한 단순 형태가 두개 이상 합쳐진 형태의 메시지를 지칭한다.

3.2 프로덕트 라인 기법이 적용된 서비스 구체화 과정

본 논문에서 제시하는 핵심 자산인 도메인 서비스를 실제 구체화하고, 개발하는 프로덕트 라인 개념을 적용한 과정은 (그림 3)과 같다. 도메인 공학 단계에서는 도메인에서 재사용될 수 있는 가변성 정보를 포함한 도메인 서비스가 기술된다. 이러한 도메인 서비스의 가변 내역에 대한 커스터마이징을 통해 가변적인 항목들이 선택되고, 제거되어 다양한 서비스 템플릿 기술서가 만들어지게 된다. 애플리케이션 공학 단계에서는 이러한 다양한 서비스 템플릿 기술서에 서비스가 실제 실행되기 위한 정보를 추가함으로써, 실제 배치되어 실행할 수 있는 서비스인 WSDL을 생성한다.



(그림 2) 도메인 서비스 아키텍처 메타모델



(그림 3) 프로덕트 라인의 기법이 적용된 도메인 서비스 구체화

4. 도메인 서비스 가변성 분석

본 장에서는 도메인 서비스의 공통적/선택적 속성 가변성, 오퍼레이션 타입 가변성, 메시지 타입 가변성을 명시적으로 기술하는 방법을 제시한다. 이러한 가변성을 설명하기 위해 서비스 지향의 유사한 호텔 예약 시스템(도메인)의 예를 사용한다. 본 연구에서는 도메인 서비스, 도메인 인터페이스, 도메인 오퍼레이션, 메시지를 표현하기 위해 <표 1>의 표기법을 사용한다.

도메인 서비스는 이중선 을 가진 사각형 형태로 표현하며, 도메인 서비스 이름과 가변적속성(공통적 또는 선택적 속성)을 표기한다. 도메인 인터페이스는 이중 라인을 가진 막대사탕 형태로 표현한다. 도메인 오퍼레이션은 이중선 을 가진 둥근 라인 사각형 형태로 표현하며, 공통적 선택적 속성(<<clp>>), 도메인 오퍼레이션의 바인딩 카디널리티(vpCardinality), 오퍼레이션명, 오퍼레이션의 입출력 메시지(in message, out message)를 표기한다. 메시지는 단순 메시지 또는 복합 메시지로 구분되는 메시지 타입과 메시지 노트로 표현된다. 메시지 노트(Message note)를 통해 메시지의 요소에 대한 정보를 구체적으로 표시한다.

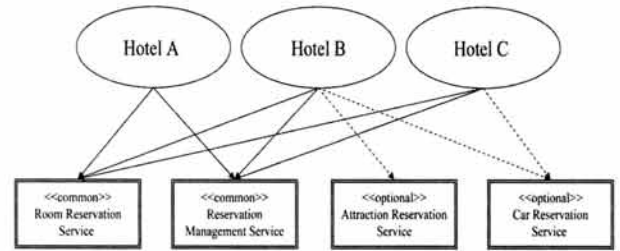
<표 1> 도메인 서비스, 인터페이스, 오퍼레이션, 메시지 표기법

도메인 서비스	
도메인 인터페이스	
도메인 오퍼레이션	
메시지	

4.1 도메인 서비스 공통적/선택적 (Common/Optional) 속성 가변성

유사한 기능을 가진 서비스 지향의 소프트웨어를 개발하는데 있어서 도메인 서비스 자체가 공통적인 요소 또는 선택적인 요소로 구분되어 활용되게 된다. 이러한 도메인 서비스 자체가 가지게 되는 가변성을 도메인 서비스의 공통적

/선택적(common/optional property) 속성 가변성으로 정의한다. 예를 들어 호텔 예약 도메인에서 추출된 도메인 서비스 중 방 예약(Room Reservation) 서비스와 예약 관리(Reservation Management) 서비스는 유사한 호텔 예약 시스템에 있어서 공통적(<<common>>)으로 사용되는 서비스이다. 그러나 차량 예약(Car Reservation) 서비스 또는 관심지 예약(Attraction Reservation) 서비스는 특정 호텔 예약 시스템에 선택적(<<optional>>)으로 사용될 수 있다.



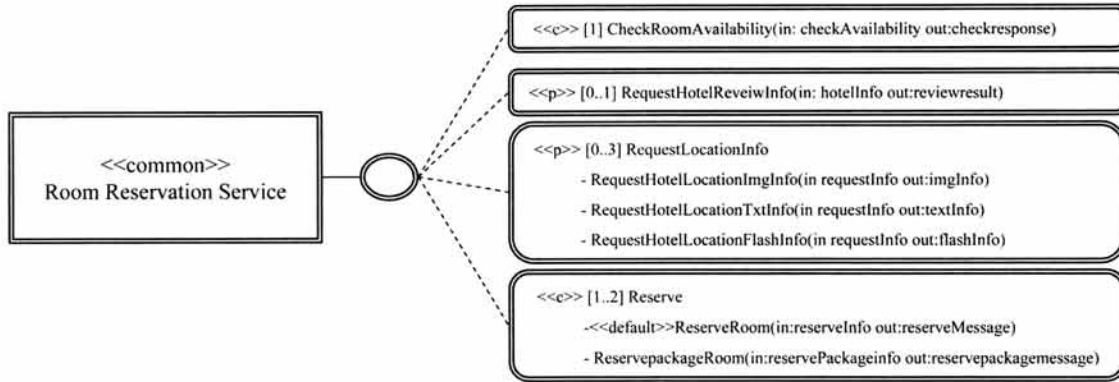
(그림 4) 공통적/선택적 서비스 선택에 따른 다양한 호텔예약 시스템 구성

4.2 오퍼레이션 타입 가변성

도메인 서비스가 제공하는 오퍼레이션들 중 서로 유사한 기능을 가진 대체될 수 있는 오퍼레이션인 경우 이를 하나의 오퍼레이션 타입으로 일반화 시킬 수 있다. 즉, 일반화된 오퍼레이션 타입은 다수의 실행 가능한 공통적 또는 선택적 오퍼레이션으로 구체화된다. 이를 오퍼레이션 타입 가변성으로 정의한다. 도메인 오퍼레이션 타입 가변성은 다음과 같은 절차로 식별된다.

- 1) 도메인 서비스에 나타나 있는 후보 오퍼레이션 중에서 대체 관계를 가질 수 있는 것을 식별한다.
- 2) 식별된 오퍼레이션들을 하나의 오퍼레이션 타입으로 일반화시킨다.
- 3) 후보 오퍼레이션들은 가변치 오퍼레이션(variant operation)으로 나타낸다.
- 4) 오퍼레이션 타입의 가변성 속성(공통적 또는 선택적)을 결정한다. 공통적으로 사용될 수 있는 오퍼레이션은 <<c>>라는 표기법으로 나타내고, 선택적으로 사용될 수 있는 오퍼레이션은 <<p>>로 표현한다.
- 5) 오퍼레이션 타입과 가변치 오퍼레이션에 대한 바인딩 카디널리티를 결정한다.

바인딩 카디널리티는 [1, 0..1, 0..N, 1..N]의 관계를 가질 수 있다. 바인딩 카디널리티 [1]로 표현한 것은 가변치 오퍼레이션 중 하나가 반드시 구체화 될 수 있음을 의미한다. 바인딩 카디널리티 [0..1]로 표현한 것은 가변치 오퍼레이션들이 최대 하나가 구체화 될 수 있음을 나타낸다. 바인딩 카디널리티 [0..N]으로 표현된 것은 가변치 오퍼레이션들이 최대 N 개가 구체화 될 수 있음을 나타낸다. 바인딩 카디널리티 [1..N]으로 표현된 것은 가변치 오퍼레이션들이 반드시 하나 이상 N개까지 구체화 될 수 있음을 나타낸다. 가변치 오퍼레이션 중 반드시 구체화되는 오퍼레이션은 default라는



(그림 5) Room Reservation 도메인 서비스의 오퍼레이션 타입 가변성

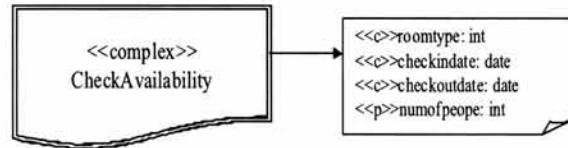
키워드를 사용하여 구분한다.

예를 들어 호텔 예약 시스템의 방 예약 (Room Reservation) 도메인 서비스는 (그림 5)와 같이 오퍼레이션 타입 가변성이 식별되어 표현된다. 예로 *CheckRoomAvailability*는 호텔 예약이 가능한가를 체크하는 오퍼레이션을 의미하는 것이고, 이는 공통적인 오퍼레이션이 되므로 <<c>>로 표현 및 바인딩 카디널리티 [1]로 기술이 된다. 또한 *ReserveRoom*은 방을 예약하는 오퍼레이션이고, *ReservepackageRoom*은 부대 서비스를 포함한 방 예약 오퍼레이션이다. 이 둘은 서로 대체 가능한 후보 오퍼레이션이 되므로 *Reserve*라는 오퍼레이션 타입으로 일반화 된다. 이 기능은 모든 호텔 예약 시스템에서 공통적이므로 <<c>>로 표현하고, 가변치 오퍼레이션들이 중복적으로 구체화될 수 있기 때문에 바인딩 카디널리티 [1..2]로 표현한다. 이때 *ReserveRoom*은 반드시 구체화 되는 오퍼레이션에 포함되는 오퍼레이션이므로 <<default>> 스테레오 타입을 추가적으로 표현한다.

4.3 메시지 타입의 가변성

도메인 서비스의 입출력 메시지가 가지는 가변성을 메시지 타입 가변성이라고 정의한다. 입출력 메시지가 기본적인 요소 하나만을 가진 메시지를 단순 타입으로 정의한다. 또한 두 개 이상의 요소로 표현되는 입출력 메시지는 하나의 복합 메시지 타입으로 일반화시키고, 복합 메시지를 구성하는 개별적인 요소들에 대해 공통적인 요소는 <<c>>, 선택적인 요소는 <<p>>로 구분하여 메시지가 가지는 가변성을 표현한다. 또한 도메인 서비스의 메시지가 가지는 가변성은 오픈 바운더리로 설정해 놓는다. 즉 도메인 서비스의 메시지 요소는 단지 공통적 또는 선택적 속성만 정의하고, 이러한 복합 메시지를 구성하는 요소에 대해서는 실제 서비스가 구체화될 때 결정할 수 있도록 한다. (그림 6)은 방 예약 (Room Reservation) 도메인 서비스의 *CheckAvailability*라는 메시지의 예를 보여준다.

이 메시지는 방 예약을 체크하는 메시지 타입으로 투숙일, 방 형태 등의 두 가지 이상의 요소가 필요한 구조이기 때문에 *CheckAvailability*라는 복합 메시지로 일반화되었다. 또한 방의 타입(*roomtype*), 투숙일(*checkindate*), 퇴실일



(그림 6) CheckAvailability 메시지

(*checkoutdate*)는 예약 체크를 위해서는 공통적인 내용이 되며, 투숙할 사람의 수(*numofpeople*)는 선택적인 사항으로 분류할 수 있다. 따라서 이러한 가변성을 가진 메시지 타입을 정의해 놓음으로써 메시지 타입의 결정을 통해 구체화된 서비스에 맞게 메시지가 결정된다.

4.4 도메인 서비스 명세서

이러한 도메인 서비스의 가변성 정보를 기술하는 도메인 서비스 명세서 형식은 <표 2>와 같다.

<표 2>에 제시되어 있는 것과 같이 Domain service name 항목에서는 도메인 서비스의 이름이 기술된다. 도메인 서비스 자체가 가지는 공통적/선택적 가변성은 common/optional property 항목에 기술을 한다. Description은 도메인 서비스의 역할 등 도메인 서비스의 부수적인 정보를 기록한다. 도메인 서비스 인터페이스는 도메인 서비스를 접근할 수 있는 인터페이스 명을 기술한다. 오퍼레이션 타입 가변성은 제시된 개념인 일반화된 오퍼레이션, 바인딩 카디널리티, 오퍼레이션으로 나누어 기술을 한다. 일반화된 오퍼레이션은 상호 유사 기능 또는 대체 관계를 가질 수 있는 가변치 오퍼레이션을 하나의 일반화된 오퍼레이션타입으로 정의한 이름을 기술한다. 바인딩 카디널리티는 [1], [0..1], [0..N], [1..N]의 타입을 가진다. [1]은 공통적인 경우를 나타내는 것으로 가변치 오퍼레이션 중 하나가 반드시 구체화됨을 나타낸다. [0..1]은 선택적인 경우를 나타내는 것으로 가변치 오퍼레이션 중 최대 하나가 구체화됨을 나타낸다. [0..N]은 선택적인 경우의 다중 선택을 나타내는 것으로 가변치 오퍼레이션이 최대 N개가 구체화 됨을 나타낸다. [1..N]은 공통적인 경우의 다중 선택을 나타내는 것으로 가변치 오퍼레이션이 반드시 하나 이상 N개까지 구체화 됨을 나타낸다. 오퍼

〈표 2〉 도메인 서비스 명세서 형식

Domain service name	도메인 서비스명 기술	common/optional property	도메인 서비스의 공통적/선택적 속성 표시
Description	도메인 서비스 역할 기술		
Domain service interface	도메인 서비스 인터페이스 기술		
Operation Type	Generalized Operation	Binding Cardinality	Operation
	일반화된 도메인 오퍼레이션 기술	[1]: 가변치 선택 하나	operation_name (in: MessageType out: MessageType) 형태로 기술
		[0..1]: 가변치 선택 최대 하나	
		[0..N]: 가변치 선택 최대 N개	
[1..N]: 가변치 선택 하나에서 최대 N개			
Message Type	Message Structure		Message Type Information
	[Simple complex]typename		type 정보 기술

레이션의 기술 형태는 *오퍼레이션명(in: 메시지 타입, out: 메시지 타입)*의 형태로 기술이 된다. 메시지 타입 가변성 기술에 있어서 메시지 타입을 메시지 구조(Message Structure)와 메시지 타입 정보(Message Type Information)로 구분한다. 따라서 메시지 구조는 단순형 및 복합형의 메시지 이름을 기술한다. 메시지 타입 정보에서 단순형은 타입 정보만, 복합 요소는 복합 요소를 구성하는 메시지 요소 이름과 타입형을 기술하고 대략적인 내용을 기술한다. 이러한 타입 이름과 타입형을 기술해 줄 때 타입이 공통적으로 활용되는 정보인지, 선택적으로 활용되는 정보인지를 구분하기 위해 선택적인 메시지 요소인 경우에는 <<option>>으로 기술해 준다. 제시된 도메인 서비스 명세서 내역에 따라 4장의 (그

림 5)에 제시된 Room Reservation 도메인 서비스를 명세하면 <표 3>과 같다.

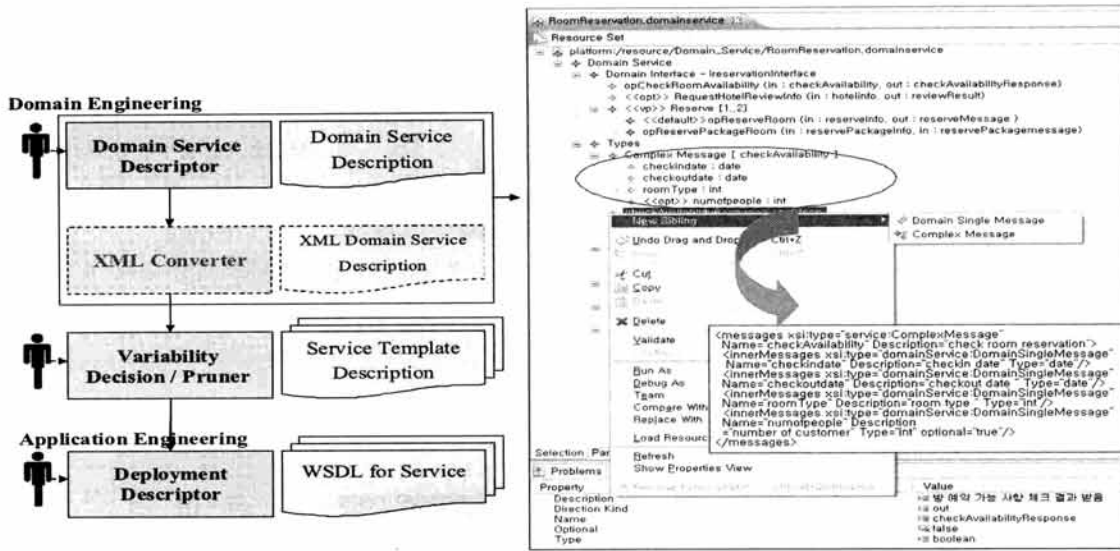
5. 도메인 서비스 명세 생성 도구 및 평가

5.1 도메인 서비스 명세 생성 도구

본 논문에서 제안한 도메인 서비스상의 가변성을 지원하고, 도메인 서비스 명세서를 자동 생성하기 위한 도구를 개발하였다. 이 도구는 이클립스 플러그인 GMF(Graphic Modeling Framework)[13]를 기반으로 개발되었다. 본 도구의 주요 기능은 (그림 7)의 왼쪽 편에 제시되어 있다.

〈표 3〉 도메인 서비스 명세 예제

Domain service name	Room Reservation	common/optional property	common
Description	This is used for hotel web service system.		
Domain service interface	IRoomReservationInterface		
Operation Type	Generalized Operation	Binding Cardinality	Operation
	-	[1]	opCheckRoomAvailabiety(in:checkAvailability, out:checkAvailabilityResponse)
	-	[0..1]	opRequestHotelReviewInfo(in:hotelinfo, out:reviewresult)
	Reserve	[1..2]	<<default>>opReserveRoom(in:reserveInfo out: reserveMessage) opReservePackageRoom(in:reservePackageInfo, out:reservePackageMessage)
RequestLocationInfo	[0..3]	opRequestHotellLocationImgInfo(in:requestInfo out: imgInfo) opRequestHotellLocationTxtInfo(in:requestInfo, out:textInfo) opRequestHotellLocationFlashInfo(in:requestInfo, out:flashInfo)	
Message Type	Message Structure		Message Type Information
	checkAvailabilityResponse	boolean->{true: 가능, false: 불가능}	
	checkAvailability	{checkindate:date, checkoutdate:date, roomtype: int, <<option>>numofpeople: int} ->{투숙일,퇴실일,방형태,투숙객수}	
	



(그림 7) 도메인 서비스 가변성 분석 지원 도구

Domain Service Descriptor는 본 논문에서 제안한 도메인 서비스 가변성 유형에 따라 도메인 서비스 가변성을 분석하여 도메인 서비스 명세서를 자동 생성한다. 그림 7의 오른쪽 부분이 Domain Service Descriptor의 동작 모습을 보여준다. XML Converter는 내부에 존재하는 것으로서 Domain Service Descriptor로 기술된 도메인 서비스를 XML 형태로 변환한다. Variability Decision/Pruner는 도메인 서비스가 가지고 있는 가변성을 선택하고 커스터마이징하여 목적에 맞는 서비스로 활용될 수 있도록 도메인 서비스의 가변성이 제거된 다수의 서비스 템플릿 명세서를 생성한다. 애플리케이션 공학단계의 Deployment Descriptor는 실제 실행 가능한 정보 및 배치 정보를 포함하여 배치되는 WSDL 문서를 생성한다.

5.2 평가

상기에 기술된 WSDL은 Room Reservation 도메인 서비스에서 가변성의 결정 후 생성될 수 있는 서비스 중 하나를 나타내고 있다. 4장의 (그림 5)와 <표 3>에 제시되어 있는 것과 같이 Room Reservation 도메인 서비스는 하나의 공통적인 오퍼레이션과 하나의 선택적인 오퍼레이션, 0개에서 3개까지의 가변치 오퍼레이션을 가진 선택적인 오퍼레이션 타입 하나, 1개에서 2개까지 선택이 가능한 오퍼레이션 타입으로 구성이 된다. 따라서 공통적인 서비스로 CheckRoomAvailibility와 ReserveRoom을 가진 Room Reservation 서비스는 반드시 생성이 되고, 나머지 5개의 선택적인 오퍼레이션의 선택적인 조합($5C_1+5C_2+5C_3+5C_4+5C_5=32$)에 따라 총 33가지 경우의 다른 형태의 서비스가 생성이 된다.

따라서 제시한 프로덕트 라인 기반의 서비스 명세 생성 방법은 도메인 서비스를 목적에 맞게 커스터마이징하여 다양한 서비스를 생성하게 함으로써 훨씬 서비스 지향 개발을 유연하고 효율적으로 수행할 수 있음을 확인 할 수 있다.

<표 4> Room Reservation 도메인 서비스 조합 예

공통적인 오퍼레이션	선택적인 오퍼레이션	생성되는 서비스 수
CheckRoomAvailability ReserveRoom	ReservePackageRoom RequestHotelReviewInfo RequestLocationImgInfo RequestHotelLocationTxInfo RequestHotelLocationFlashInfo	33가지

6. 결론 및 향후 연구

본 논문에서는 프로덕트 라인 기반으로 서비스를 기술하는 방법을 제시하였다. 즉 SOA에서의 서비스 지향 시스템들의 개발에 있어서 민첩성, 재사용성, 효율성을 증대시키기 위해 프로덕트 라인의 개념을 접목하여 도메인에서 사용될 수 있는 가변성을 가진 도메인 서비스를 기술하는 방법을 제안하였다. 우선, 도메인 서비스 아키텍처 메타 모델을 제안하였으며, 도메인 서비스가 가지는 가변성 유형 타입을 분류하여 제시하였다. 제안한 도메인 서비스 가변성 타입의 유형은 도메인 서비스 자체가 가질 수 있는 공통적/선택적 가변성, 오퍼레이션 타입이 가변치 오퍼레이션으로 구체화될 수 있는 오퍼레이션 타입 가변성, 도메인 서비스의 입력 출력 메시지가 가변될 수 있는 메시지 타입 가변성이다. 이러한 분석된 가변성 유형을 명확하게 기술할 수 있는 도메인 서비스 명세서를 제시하였으며, 이를 자동 생성하기 위한 지원 도구를 개발하였다.

본 논문에서 제안한 서비스 기반의 핵심 자산인 서비스를 생성 및 기술하는데 있어서 프로덕트 라인 기반 접근법은 가변성이 포함되어 있는 도메인 서비스의 선택을 통한 커스터마이징을 수행한다. 따라서 다양한 실행 수준 서비스(공통적인 기능만 가진 서비스, 공통적인 기능과 선택적인 기능

을 가진 서비스)의 유형 생성을 체계적으로 지원하므로 서비스 지향의 핵심 자산인 서비스 생성 측면의 유연성과 확장성을 증대시킨다. 향후 연구로는 도메인 서비스를 기반으로 다양한 서비스 조합시 분석될 수 있는 가변성에 대하여 연구를 진행할 것이며, 도메인 서비스 가변성 분석 지원도구를 확장해 나갈 것이다.

참 고 문 헌

[1] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*, Addison Wesley, 2001.

[2] M. Moon, K. Yeom and H.S. Chae, "An Approach to Developing Domain Requirements as a Core Asset Based on Commonality and Variability in a Product Line", *IEEE Transactions on Software Engineering*, Vol.31, No.7, pp.551-569, Jul., 2005.

[3] M. Bichler and K. J. Lin, "Service-Oriented Computing", *Computer*, Vol.39, No.3, pp.99-101, Mar, 2006.

[4] T. Erl, *SOA Principles of Service Design*, Prentice-Hall, 2008.

[5] P. Fremantle, S. Weerawarana and R. Khalaf, "Enterprise services", *Communications of the ACM*, Vol.45, No.10, pp.77-82, Oct., 2002.

[6] M.P.Papazoglou and W.J.V.D.Heuvel, "Service-oriented architectures: approaches, technologies and research issues", *International Journal on Very Large Data Bases*, Vol.16, No.3, pp.389-415, Jul., 2007.

[7] S. Cohen and R. Krut, Chapter1 in the *First Workshop on Service-Oriented Architectures and Software Product Lines*, May, 2008.

[8] K. Pohl, G. Böckle and F.V. Linden, *Software Product Line Engineering Foundations, Principles, and Techniques*, Springer, Berlin Heidelberg New York, 2005.

[9] W3C, "Web Service Description Language (WSDL) Version 2.0 Part 0: Primer", <http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/>

[10] J. Jiang, A. Ruokonen and T. Systa, "Pattern-based variability management in Web service development", *Third IEEE European Conference on Web Services*, pp.12-23, Nov., 2005.

[11] S. Chang, H. La and S. Kim, "A Comprehensive Approach to Service adaptation", *IEEE International Conference on Service Oriented Computing and Applications*, pp.191-198, Jun., 2007.

[12] S. Segura, D. Benavide, A. Ruiz-Cortes and P. Trinidad, "A Taxonomy of Variability in Web Services Flows", *First Workshop on Service-Oriented Architectures and Software Product Lines*, May, 2008.

[13] *Graphical Modeling Framework Home*, <http://www.eclipse.org/gmf/index.php>



박 준 석

e-mail : pjs50@pusan.ac.kr

1999년 부경대학교 컴퓨터공학과(학사)
2002년 부산대학교 컴퓨터공학과(공학석사)
2002년~현 재 부산대학교 컴퓨터공학과 박사과정

관심분야: 소프트웨어 프로덕트 라인 공학, 소프트웨어 아키텍처, SOA, 유비쿼터스 컴퓨팅 등



문 미 경

e-mail : mkmoon@dongseo.ac.kr

1990년 이화여자대학교 전자계산학과(학사)
1992년 이화여자대학교 전자계산학과 (이학석사)
2005년 부산대학교 컴퓨터공학과(공학박사)
2005년~2005년 부산대학교 차세대물류IT 기술사업단 박사후 연구원

2005년~2006년 부산대학교 컴퓨터 및 정보통신 연구소 기금교수
2006년~2008년 부산대학교 정보컴퓨터공학부 연구교수
2008년 3월~현 재 동서대학교 컴퓨터정보공학부 전임강사
관심분야: 소프트웨어 프로덕트 라인 공학, 적응형 소프트웨어 개발, RFID 미들웨어 개발 및 RFID 기반 애플리케이션 개발 등



염 근 혁

e-mail : yeom@pusan.ac.kr

1985년 서울대학교 계산통계학과(학사)
1992년 Univ. of Florida 컴퓨터공학과 (석사)
1995년 Univ. of Florida 컴퓨터공학과 (박사)

1985년~1988년 금성반도체 컴퓨터연구실 연구원
1988년~1990년 금성사 정보기기연구소 주임연구원
1995년~1996년 삼성SDS 정보기술연구소 책임연구원
1996년~현 재 부산대학교 정보컴퓨터공학부 교수
관심분야: 소프트웨어 재사용, 프로덕트 라인 공학, 소프트웨어 아키텍처, 센서 네트워크기반 상황인식 미들웨어, 적응형 소프트웨어 개발 등