

서비스 지향 프로덕트 라인 아키텍처 기반 비즈니스 프로세스 자동화를 위한 프레임워크

예은숙* · 문미경** · 홍민우*** · 염근혁****

1. 서 론

오늘날 증시되고 있는 비즈니스 환경은 ‘신속성’이다[1]. 신속성을 향상시키기 위해 기업은 기업의 비즈니스 전략과 비즈니스 목표에 맞추어 운영을 재조정할 수 있는 유연성을 필요로 한다. 신속성과 유연성의 조합을 통해 기업은 제품과 서비스의 타임 투 마켓, 고객 대응력, 고객 만족도 면에서 경쟁력을 확보할 수 있다.

이를 위해 각 기업들은 내부의 데이터, 애플리케이션 및 프로세스의 통합뿐 아니라 공급자와 파트너 등 다수 기업과의 협업 관계에 의한 프로세스 통합을 진지하게 고려하고 있다. 이러한 비즈니스 요구를 효율적으로 대응하기 위한 IT 아키텍처 전략으로서 주목되고 있는 것이 서비스 지향 아키텍처(Service Oriented Architecture, SOA)이다.

SOA는 재사용성이 있고 상호운용이 가능한 서비스들이 느슨하게 연결되어 조합된 애플리케이션

이선 개발을 가능하게 하는 아키텍처이다[2]. 기존의 방대한 코드로 이루어진 애플리케이션들을 개발하기 위한 것이 아니라 비즈니스 기능을 수행하는 서비스를 구성하고 이들 서비스를 조합함으로써 비즈니스 프로세스를 구현할 수 있게 하는 정보 시스템 구축을 위한 아키텍처이다.

이러한 SOA는 전통적인 소프트웨어 중심의 분석, 설계, 개발 방식에서 비즈니스 프로세스 관점에서 재활용 가능한 단위로 서비스를 분석, 설계, 개발함으로써 특정 프로세스나 서비스의 변경 또는 내외부 시스템과의 비즈니스 통합시 효율적이고 빠른 대응이 가능하다는 점에서 그 의미가 크다.

그러나, 시시각각 변하는 시장에 적응할 수 있는 프로세스를 필요 시마다 구축하기 위해서 드는 시간과 비용도 상당하다. 따라서 기업에서는 프로세스를 변경하는데 드는 시간과 비용이 최대한 적게 들면서 프로세스의 빠른 설계와 구축을 요구하게 되는데, 소프트웨어 프로덕트 라인 공학(Software Product Line Engineering, SPLE)은 이러한 요구를 만족시킬 수 있다.

소프트웨어 프로덕트 라인(Software Product Line, SPL)이란, 일련의 비슷한 소프트웨어들의 집합이다[3]. SPL은 소프트웨어들 간의 공통적인 요소와 서로 다른 요구를 만족하는 요소들을 공유하고, 이러한 요소들은 미리 만들어진 공통적인

※ 교신저자(Corresponding Author) : 염근혁, 주소 : 부산시 금정구 장전동 산30번지(609-735), 전화 : 051)510-2475, FAX : 051)517-2431, E-mail : yeom@pusan.ac.kr

* 부산대학교 컴퓨터공학과 석사과정 (E-mail : yeeunsuk@naver.com)

** 부산대학교 정보컴퓨터공학부 연구교수 (E-mail : mkmoon@pusan.ac.kr)

*** 부산 대학교 컴퓨터공학과 석사과정 (E-mail : hmw2525@naver.com)

**** 부산대학교 정보컴퓨터공학부 부교수

핵심 자산 집합에 의해 제공된다. SPL에서 중요한 것은, 재사용될 수 있는 공통성의 부분과 요구에 따라 변경이 필요한 가변성 부분을 분석하는 것이다. 공통성과 가변성이 식별되어 명시적으로 표현된 프로덕트 라인 아키텍처(Product Line Architecture, PLA)는 재사용될 수 있는 핵심자산으로 관리된다. 그리고, PLA에 나타난 가변성을 구체화하여 도메인 내의 애플리케이션을 개발한다.

본 연구에서는 SPLE 기술을 SOA에 적용하였다. 도메인 공학 단계에서 서비스 지향 프로덕트 라인 아키텍처 (Service Oriented Product Line Architecture, SOPLA)를 핵심자산으로 관리하고 이를 이용하여 비즈니스 프로세스 패밀리 (Business Process Family, BPF)를 개발한다. 여기서 BPF는 유사한 비즈니스 프로세스의 집합을 의미한다. SOPLA는 BPF의 모든 비즈니스 프로세스들이 갖는 공통성뿐 아니라 비즈니스 프로세스마다 다른 목표를 이루기 위해 달라지게 되는 가변성을 포함한다. SOPLA는 BPF의 개발에 재사용할 수 있도록 미리 계획되어 만들어졌기 때문에, SOPLA로부터 특정한 비즈니스 프로세스를 개발하는 시간이 단일의 비즈니스 프로세스를 구축하는 시간보다 단축될 수 있다. 이는 더욱 기업 프로세스의 신속성과 유연성을 지원해주는 기술임이 분명하며 프로세스를 변경하는데 드는 시간과 비용을 줄일 수 있다.

따라서, SOPLA를 개발하고 이를 재사용하여 BPF를 개발하는 과정인 아키텍처 중심의 서비스 지향 프로덕트 라인 공학 프로세스와 SOPLA의 설계 지원도구를 개발하는 것이 본 논문의 목표이다.

본 논문의 구성은 2장에서 연구의 배경지식인 SPLE와 SOA에 대해 설명하고, 3장에서는 SO-

PLA에 대해 소개하고, 4장에서 SOPLA를 기반으로 BPF를 개발하기 위한 프레임워크 구조를 설명한다. 그리고 5장에서 SOPLA의 설계를 지원하는 도구를 소개하고 6장에서 결론 및 향후 연구를 마지막으로 마무리한다.

2. 연구 배경

2.1 프로덕트 라인 공학

프로덕트 라인 공학(Product Line Engineering, PLE)[3]은 유사한 여러 애플리케이션들을 프로덕트 라인으로 모으고, 이들이 공유할 수 있는 핵심자산의 재사용을 통해 애플리케이션을 개발하는 새로운 소프트웨어 개발 방법이다. 핵심자산은 프로덕트 라인 멤버들이 공통적으로 이용하기 때문에, 공통성과 가변성을 잘 정의하여 높은 재사용성을 가진 핵심자산을 개발하는 것은 고품질의 애플리케이션을 보다 적은 예산을 들여서 빠른 시간에 개발할 수 있게 한다. 이를 위해 PLE은 핵심자산을 개발하기 위한 도메인 공학 단계(Domain Engineering)와 핵심자산을 재사용하여 애플리케이션을 개발하는 단계인 애플리케이션 공학 단계(Application Engineering)로 구성된다. 그림 1은 PLE의 두 단계와 핵심자산의 관계를 보이고 있다.

도메인 공학 단계에서는 프로덕트 라인 멤버들

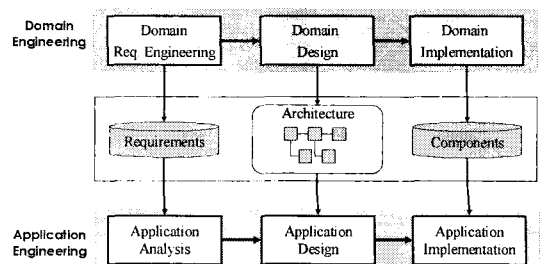


그림 1. 프로덕트 라인 공학

의 공통성과 가변성을 분석하여 재사용할 수 있는 핵심자산을 개발한다. 이를 위해 도메인 분석, 아키텍처 개발과 재사용 가능한 자산 개발 단계로 이루어지며 이 과정에서 요구사항, 분석모델, 아키텍처, 재사용 컴포넌트 등의 산출물이 생성된다.

프로덕트 라인 아키텍처(Product Line Architecture, PLA)는 핵심자산의 주요한 산출물 중 하나로서, 일반적인 소프트웨어 아키텍처와는 달리 프로덕트 라인에 속한 여러 애플리케이션 아키텍처에 대한 공통성과 가변성 요소를 가진 아키텍처이다. 공통성 요소는 모든 프로덕트 라인 멤버 아키텍처에 포함되는 것이고, 가변성 요소는 각 프로덕트 라인 멤버의 고유한 요구사항에 따라 달라지게 되는 요소이다.

애플리케이션 공학 단계에서는 미리 정의된 핵심자산을 요구에 맞게 재사용한 애플리케이션을 개발한다. 이 단계에서는 핵심자산을 바탕으로 애플리케이션의 목표를 위한 개발 산출물이 생성된다.

2.2 서비스 지향 아키텍처(SOA)

SOA (Service Oriented Architecture)[1,2]는 1996년 가트너(Gartner)그룹에 의해 최초 소개되어 현재 소프트웨어 분야의 최신 경향 중 하나이다. 가트너는 SOA를 '잘 정의된 인터페이스들을 가진, 재사용이 가능한 일련의 컴포넌트들로 구축되는 기술구조 방식이다' 라고 정의하고 있다.

SOA가 추구하는 목표는, 플랫폼간의 상호 운영성(Interoperability), 느슨한 결합(Loose Coupling), 기존의 존재하는 시스템의 재사용성(Reusability)과 기존의 서비스를 이용하여 새로운 서비스를 생성할 수 있는 능력(Composability)을 달성하는 애플리케이션 아키텍처이다.

SOA는 특정 기술이나 플랫폼에 독립적이다. 기존의 기술 중심의 관점에서 개발된 애플리케이션은 특정 기술이나 플랫폼에 수행능력이 제한되어 비즈니스까지 제한될 수 있었지만, SOA는 비즈니스 모델을 지원할 수 있도록 서비스를 구성한다. 즉, 비즈니스 모델 중심으로 사용 가능한 독립적인 기술(서비스)을 느슨하게 연결하여 비즈니스를 수행할 수 있게 한다. 서비스들이 느슨하게 연결되어 있다는 의미는 한 서비스의 수행은 다른 서비스에 의존하는 것 없이 독립적으로 이루어진다는 것이다.

따라서, 서비스는 애플리케이션보다 비즈니스에 더 밀접하게 관련을 맺고 있고 비즈니스 모델을 보다 정확하게 구현한다[4]. 서비스는 비즈니스 프로세스를 지원하기 위해 정의되고 프로세스 전반에 걸쳐 프로세스의 진행을 위해 다양한 서비스들이 조합된다. SOA모델들은 비즈니스 모델을 구성하는 프로세스와 서비스들로 구축된다. 즉, SOA를 설계하는 것은 비즈니스 모델을 지원하는 유용한 일련의 서비스 집단을 만드는 것이다.

3. 서비스 지향 프로덕트 라인 아키텍처(SOPLA)

서비스 지향 프로덕트 라인은 유사한 비즈니스 프로세스의 집합을 의미하며 프로덕트 라인 공학 프로세스를 포함하고 있다. 서비스 지향 프로덕트 라인 공학 프로세스는 서비스 지향 도메인 공학 프로세스와 서비스 지향 애플리케이션 공학 프로세스로 구성된다. 이들은 일반적인 소프트웨어 보다는 비즈니스 프로세스를 개발하기 위한 것이다.

서비스 지향 프로덕트 라인 아키텍처(SOPLA)는 서비스 지향 도메인 공학 프로세스에서 생성되는 산출물이다. SOPLA는 일반적인 프로덕트 라인 아키텍처와 상통한다. 그렇기 때문에 SOPLA

는 서비스 지향 프로덕트 라인의 재사용 할 수 있는 핵심 아키텍처가 되고 비즈니스 프로세스 구성 요소들의 공통적인 규칙을 정의하고 있다. 공통적인 규칙은 프로덕트 라인에 포함되는 비즈니스 프로세스들 사이에 공통적으로 사용할 수 있는 요소와 요구 사항에 따라 달라지는 요소를 정의하고 기술한 것이다. 중요한 것은, 달라지는 요소의 정의 즉 가변성을 구현할 수 있도록 가변점과 가변종의 표현과 함께 가변성 구현 메커니즘을 정의하는 것이다.

SOPLA를 기술하기 위하여 UML 액티비티 다이어그램을 사용하였다. 액티비티 다이어그램은 UML의 행위모델의 한 종류로서 제어흐름을 기술할 수 있기 때문에 비즈니스 프로세스를 모델링할 때 유용하게 사용된다. UML 액티비티 다이어그램 외에도 BPMN(Business Process Modeling Notation)[5], BPML(Business Process Modeling Language)[6] 등의 언어를 이용하여 비즈니스 프로세스를 모델링 할 수 있지만 소프트웨어 개발자 중심의 프로덕트 라인 개발이라고 가정하면 새로운 표기법을 익히는 것은 번거로울 수 있기 때문에 두루 통용되고 있는 모델링 언어인 UML을 사용한다.

그러나 액티비티 다이어그램의 요소로는 SOPLA가 나타내는 중요한 부분인 공통성과 가변성을 표현하는데 한계가 있다. 특히 가변성을 나타내기 위해 가변점과 가변종, 구현 메커니즘의 표현 능력이 부족하다. 다른 연구에서 UML의 확장 메커니즘인 스테레오 타입과 태그드 밸류를 이용함으로써 액티비티 다이어그램에서 가변성을 표현하기도 하나, 본 연구에서는 보다 쉽게 분별할 수 있도록 다른 모양의 요소를 첨가하여 SOPLA를 모델링 한다. 추가되는 액티비티 다이어그램의 요소에 대해서는 5장에서 액티비티 다

이어그램의 메타 모델을 보이고 설명한다. 그림 2에서 공통성과 가변성을 정의하고 있는 SOPLA 모델과 이에서 유사한 비즈니스 프로세스 모델들을 생성함을 보이고 있다. 공통적으로 사용할 수 있는 요소는 실선으로, 가변적인 요소는 회색으로 채워진 점선의 액션 기호로 나타내었다.

또한, SOPLA는 다이어그램과 더불어 XML[7]로 기술된다.

4. SOPLA기반 비즈니스 프로세스 자동화 프레임워크

SOPLA는 유사한 비즈니스 프로세스의 집합을 프로덕트 라인 공학 기술로 관리하는 아키텍처이다. 이것을 재사용하여 BPF를 개발할 수 있다. SOPLA로부터 비즈니스 프로세스 모델을 생성하고 비즈니스를 실행하기 위해 본 연구는 SOPLA 기반 비즈니스 프로세스 개발 프레임워크를 제시한다[8].

이것은 프로덕트 라인 공학 프로세스와 유사하다. 핵심자산(core asset)으로 관리되는 비즈니스 프로세스 아키텍처를 개발하는 도메인 공학 단계와 비즈니스 목표와 전략을 반영한 다양한 비즈니스 프로세스 모델을 생성하는 변환 단계와 비즈니스 프로세스 공학 단계로 구성된다(그림 3).

도메인 공학 단계에서는 한 도메인 내의 비즈

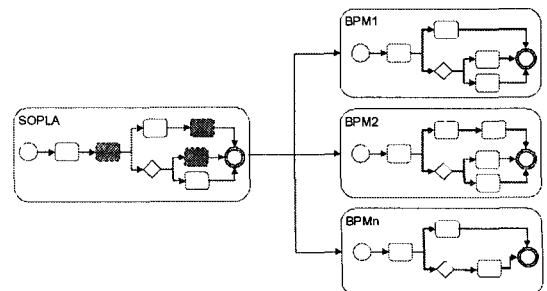


그림 2. 서비스 지향 프로덕트 라인 아키텍처(SOPLA)

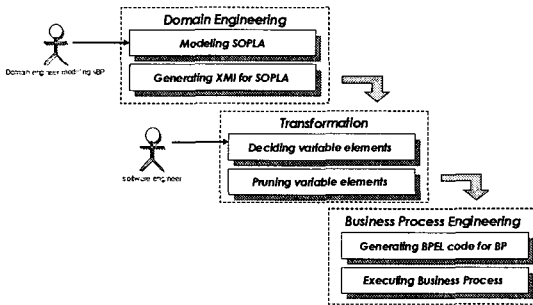


그림 3. BPF개발을 위한 프레임워크

니스 프로세스들 사이의 공통성과 가변성을 식별한다. 그 다음 식별된 공통성과 가변성을 명시적으로 나타낸 비즈니스 프로세스 아키텍처를 개발한다. 이것은 프로덕트 라인의 핵심자산으로 관리되고 이후 비즈니스 프로세스를 생성할 때 재사용된다. 즉, 비즈니스 전략에 따라 비즈니스 프로세스를 설계할 때 그것은 유용한 기반 정보로 활용하게 된다.

변환 단계에서는, 기업의 비즈니스 목표와 전략에 따른 비즈니스 프로세스 요구사항을 분석하고 그에 맞춰 비즈니스 프로세스 아키텍처를 커스터마이징한다. 비즈니스 프로세스 공학 단계에서는, 구체화된 비즈니스 프로세스 모델을 BPEL (Business Process Execution Language) [9]로 변환하고 기술하여 비즈니스 프로세스를 운용하게 된다.

4.1 도메인 공학 단계(Domain Engineering)

도메인 공학 단계에서는 SOPLA를 모델링하고 이 모델을 코드로 생성하는 과정으로 구성된다.

SOPLA를 모델링하기 위해서는 먼저 유사한 BP들을 분석함으로써 비즈니스 서비스에 대하여 공통성과 가변성을 식별하여야 한다. 모든 비즈니스 프로세스에서 항상 포함하는 비즈니스 서비스는 공통성으로 나타내고 비즈니스 프로세스마다

변화가 있는 것은 가변성으로 식별할 수 있다. 예를 들면 주문하기 프로세스에서 결제 서비스는 공통성으로 뽑을 수 있고 멤버십 포인트로 결제하기 서비스는 가변적인 요소로 뽑을 수 있다.

비즈니스 프로세스에서 액티비티의 존재 여부에 따른 가변성 타입은 다음과 같이 분류한다.

- 공통 비즈니스 액티비티 - 공통적인 속성을 가지므로, 모든 비즈니스 프로세스에 포함된다.
- 가변성을 가진 공통 비즈니스 액티비티 - 공통적으로 포함되는 비즈니스 액티비티이지만 이것은 주로 추상적 액티비티를 나타내고 구체적인 액티비티 요소가 가변된다.
- 선택적 비즈니스 액티비티 - 비즈니스 프로세스에 포함될 수도 안될 수도 있는 액티비티이다.
- 가변성을 가진 선택적 비즈니스 액티비티 - 추상적인 선택적 비즈니스 액티비티가 비즈니스 프로세스에 포함될 때 구체적인 액티비티 요소로의 결정이 필요한 가변요소를 가진다.

그리고, 비즈니스 액티비티 내에 존재할 수 있는 가변성 타입을 다음과 같이 분류한다.

- 데이터 가변성 - 비즈니스 액티비티의 파라미터에서 나타나는 가변성
- 게이트웨이 가변성 - 비즈니스 액티비티들의 조건적인 수행 흐름에서 나타나는 가변성
- 이벤트 가변성 - 비즈니스 액티비티들의 수행을 위한 이벤트에서 나타나는 가변성

이러한 비즈니스 프로세스의 공통적, 가변적인 속성을 분석한 것을 바탕으로, 가변적인 비즈니스 프로세스 모델은 핵심자산으로 관리될 수 있도록 표현되고 기술되어야 한다. 그러므로 앞에서 언급

하였듯이 UML 액티비티 다이어그램에 가변성을 위한 추가 요소를 더하여 SOPLA를 표현한다.

가변성을 나타내는 요소는 다음과 같은 가변성 정보를 가진다.

- 가변이 발생하는 위치(가변점)
- 가변점에서 바인딩될 수 있는 가변종의 정의
- 바인딩을 위한 속성- 가변성 타입, 바인딩되는 가변종의 개수, 바인딩되는 가변종의 관계, 제약사항 등

그러나 도형으로 표현한 다이어그램은 비즈니스 프로세스의 가변성의 속성을 표현하는데 한계가 있기 때문에 비즈니스 프로세스를 문자로 작성하는 것이 필요하다. 즉, SOPLA 모델의 다이어그램 뿐 아니라 가변성의 속성 정보를 코드언어로 표현하여 비즈니스 프로세스 모델을 상세화해야 한다.

위에서 제시한 가변성을 나타내는 정보를 표현할 수 있는 SOPLA를 기술하기 위해 본 논문에서는 XML기반의 언어인 XMI로 기술한다. 이로서 SOPLA 모델을 도메인 공학 단계의 핵심자산으로 공유되거나 재사용이 용이하도록 하는 것이 가능하다[10].

그림 4는 SOPLA가 다이어그램과 XMI로 생성되고 이를 재사용하여 가변성 바인딩을 통하여 비즈니스 프로세스를 생성하는 단계를 보이고 있다.

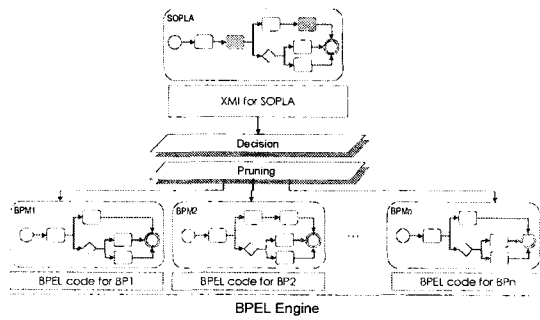


그림 4. 비즈니스 프로세스를 위한 PLA

4.2 변환 단계 (Transformation)

변환 단계는 SOPLA의 바인딩될 가변적 요소를 결정하고(Decision), 결정 이후 선택되지 않은 가변 요소를 제거하는 과정(Pruning)으로 구성된다. 새로운 비즈니스 프로세스를 구현하기 위해서는 비즈니스 목표와 전략에 따른 프로세스의 분석이 수행되어야 한다. 기업의 비즈니스 목표와 전략은 변화할 수 있다. 비즈니스 목표와 전략이 바뀌면 비즈니스 프로세스를 설계하기 위한 요구사항도 변경된다. 그러므로 비즈니스 목표를 달성하기 위한 비즈니스 요구사항이 분석되고 명세되어야 한다. 비즈니스 요구사항은 결정의 근거가 된다. 결정은 엔지니어에 의해 행해지므로 변환 단계에서 SOPLA는 가변점을 강조하여 표현할 수 있다. 이 지점은 가변성을 가진 요소가 존재하기 때문에 결정이 수행되어야 함을 나타낸다. 또, 결정은 가변점들 간에서나 가변종들 간에서 서로 영향을 줄 수 있다. 그래서 그들의 의존 관계를 나타내어 엔지니어가 올바른 결정을 수행할 수 있도록 지원할 수 있다.

SOPLA에서 가변성 타입에 따라 결정을 수행하는 방법을 분류할 수 있다.

- 선택하기(Selection) - 선택적인 가변성 타입이면 비즈니스 액티비티를 선택하거나 선택하지 않음으로 결정이 수행된다.
- 구성하기(Configuration) - 비즈니스 액티비티 내에 존재하는 데이터 가변성, 게이트웨이 가변성, 이벤트 가변성 타입에서는 파라미터 값을 설정함으로써 결정을 수행한다.

만약 가변 요소들 중 디폴트로 정의된 것이 존재하면, 엔지니어의 결정이 없이 디폴트 요소로 시스템이 가변성을 바인딩 할 수 있다.

새로운 비즈니스 프로세스를 구현하기 위해 도

메인 공학 단계에서 분석되지 못한 비즈니스 액티비티가 필요할 수도 있다. 이때에는 SOPLA에서 가변점으로 나타나지 않기 때문에 필요한 비즈니스 액티비티를 구현하거나 서비스 레퍼지토리아에서 찾아 추가할 수 있다.

결정을 수행한 후에는 가지치기 과정을 수행한다. 이 단계에서는 가변점에서 바인딩 되지 않은 가변요소는 제거된다. 즉, 바인딩 되지 않은 선택적 비즈니스 액티비티가 비즈니스 프로세스에서 삭제되고 바인딩 되지 않은 가변 파라미터나 가변적 제어 흐름은 삭제되어 프로세스가 수행된다. SOPLA에서 공통성의 요소와 바인딩된 가변성 요소를 가져와서 비즈니스 프로세스 모델을 재구성하는 것이다.

그림 4에서 SOPLA로부터 결정과 가지치기를 수행하여 다양한 여러 개의 비즈니스 프로세스 모델을 생성함을 보이고 있다.

4.3 비즈니스 프로세스 공학 단계 (Business Process Engineering)

비즈니스 프로세스 공학 단계에서 각각 비즈니스 프로세스 모델은 자동적 실행을 지원하는 BPEL로 기술된다. BPEL은 웹 서비스를 기반으로 이들을 조합하여 비즈니스 프로세스를 만드는 표준으로 제안된 스펙이다. 이것은 비즈니스 프로세스 흐름에 관련된 웹 서비스들을 조합하는데 필요한 컨트롤 로직을 XML로 표현한다. UML 액티비티 다이어그램이나 BPMN으로 표현된 비즈니스 프로세스 모델을 BPEL로 변환하기 위한 연구가 상당히 진행되었고, BPMN을 BPEL로 변환은 상용 툴에서 제공하고 있다.

BPEL로 변환된 비즈니스 프로세스 모델은 BPEL엔진에 의해 운용하게 된다. 이 BPEL엔진은 웹서비스 환경에서 비즈니스 프로세스에 포함된

서비스를 수행하며, 이기종 시스템과의 바인딩, 예외 발생시 보정 트랜잭션 등을 고려한 서비스 통합에 대한 기술적 문제를 해결해 주기도 한다.

5. 지원도구

4장에서 제시한 프레임워크의 도메인 공학 단계 과정인 SOPLA를 개발하도록 지원하는 도구를 개발하였다. SOPLA 모델의 다이어그램을 작성하는 기능과 다이어그램의 XMI 생성 기능을 제공하기 위해 이클립스 플랫폼 기반의 플러그인으로 개발하였다. 이클립스는 모델링 기능을 쉽게 개발할 수 있도록 그래픽 편집 플러그인 개발에 유용한 프레임워크를 지원하기 때문에 그래픽 에디터에서 다이어그램을 작성하는 도구를 편하게 개발할 수 있도록 해준다. 이 장에서는 지원도구의 기능에 대한 요구사항과 SOPLA모델의 메타 모델을 설명한다.

5.1 기능 분석

본 논문에서는 제시된 프레임워크를 따라 SOPLA를 모델링하고 관리할 뿐 아니라 비즈니스 프로세스를 모델링하고 구현하는 프로세스의 자동화를 지원하는 도구를 개발하고 있다.

지원 도구 개발을 위한 분석 사항은 다음과 같다. 표 1은 지원도구의 기능적 요구사항의 일부를

표 1. 기능 요구사항

SP100	비즈니스 프로세스 다이어그램 작성
SP101	새 모델 작성
SP200	비즈니스 프로세스 모델 관리 기능
SP201	비즈니스 프로세스 모델 파일 열기
SP202	비즈니스 프로세스 모델 저장
SP300	비즈니스 프로세스 다이어그램의 XMI파일 생성
...	...

보이고 있고, 표 2는 요구사항 중 새 모델 작성에 대한 요구 명세서의 일부이다.

표 2. 새 모델 작성 요구사항 명세서

유스케이스 번호	SP101
유스케이스 이름	새 모델 작성
개요	새로운 다이어그램 만들기 위해 각 뷰들을 준비한다.
관련 액터	Domain modeler, Application modeler
선행조건	없음
후행조건	프로그램내의 트리뷰, 팔레트, 속성뷰 등이 활성화되고, 새로운 에디터가 열리고 modeler의 작업을 기다리고 있다.
이벤트 흐름	기본흐름 1. Modeler는 프로그램의 'new BPF'를 선택한다. 2. 프로그램은 작성할 모델의 이름을 입력받는다. 3. 프로그램은 받은 이름으로 empty diagram을 생성하고 트리 뷰, 팔레트, 속성 뷰를 리로드하고 새로운 에디터를 연다.
	대안흐름 1. Modeler는 프로그램의 'new BP'를 선택한다. 2. 프로그램은 작성할 BP의 이름을 입력받는다. 3. 프로그램은 비즈니스 프로세스 작성 방법 선택 다이얼로그를 띄운다. * 비즈니스 프로세스 작성 방법 - modeler가 직접 비즈니스 프로세스 모델을 작성 (use Domain Model) - 비즈니스 프로세스 패밀리로부터 비즈니스 프로세스 모델을 작성 (create Model directly) 4. 프로그램은 받은 이름으로 empty diagram을 생성하고 트리 뷰, 팔레트, 속성 뷰를 리로드하고 새로운 에디터를 연다.

그림 5는 지원도구의 기능을 나타낸 유스케이스 다이어그램이다. 지원도구의 사용자는 도메인 공학 단계에서 모델을 작성하는 모델러와 변환단계에서 작업을 하게 되는 애플리케이션 모델러가 존재하고, 각 액터와 연관된 기능이 나타나 있다.

5.2 SOPLA 모델의 메타모델

비즈니스 프로세스와 SOPLA모델은 UML 액티비티 다이어그램으로 표현한다. 그러나 SOPLA는 유사한 비즈니스 프로세스들에 대한 공통성과 가변성 요소를 가지기 때문에 액티비티 다이어그램의 표준 요소에 가변성을 명확히 표현하기 위한 요소가 추가되어야 한다. 그림6은 확장된

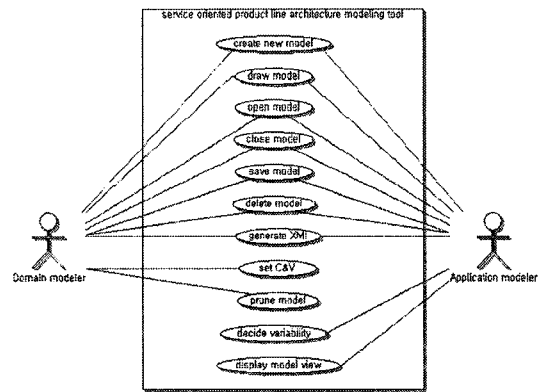


그림 5. 유스케이스 다이어그램

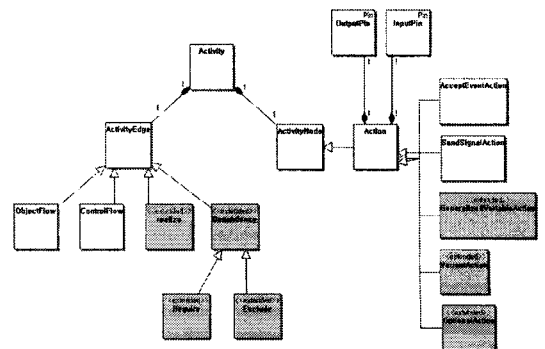


그림 6. 확장된 액티비티 다이어그램 메타모델

UML 액티비티 다이어그램의 메타모델이다. 추가된 요소는 <<extended>>스테레오 타입을 표기하고 회색으로 구분된다.

가변성을 가진 액션으로 OptionalAction, GeneralizedVariableAction, VariantAction이 있다.

OptionalAction은 선택적인 액션이다. 수행하거나 하지 않음의 선택을 하여 가변성을 바인딩 할 수 있다. GeneralizedVariableAction은 하위에 포함되는 VariantAction으로 바인딩 되는 추상적 액션이다. 바인딩 속성에 지정된 대로 VariantAction이 바인딩 될 수 있다. VariantAction은 GeneralizedVariableAction의 하위에 존재하고 변환단계에서 결정된 VariantAction은 GeneralizedVariableAction을 구체화하게 된다. GeneralizedVariableAction에서 VariantAction로 구체화됨을 realize관계로 표현한다. Dependency 관계는 Require와 Exclude관계가 존재할 수 있다. 한 가변성 액션에서 가변성 바인딩이 될 때 그 선택된 요소가 다른 가변성 요소의 결정이나 특정 바인딩 요소로의 결정을 필수로 하는 관계를 Require로 나타낼 수 있다. 반대로, 한 가변성 요소의 결정이 다른 가변성 요소에서 특정 바인딩 요소로의 결정이나 가변성 요소의 존재에 대해 충돌이 있을 경우 Exclude관계를 설정하여 나타낼 수 있다.

5.3 지원도구 유저인터페이스

지원도구의 기능 요구사항을 도출하고 SOPLA의 메타모델을 적용하여 이클립스 플랫폼 기반 플러그인 형태의 도구를 개발하였다. 그림7에서는 이를 실행한 도구의 유저인터페이스를 보이고 있다.

6. 결론 및 향후 연구

비즈니스의 신속성과 유연성을 위해 유연한 비

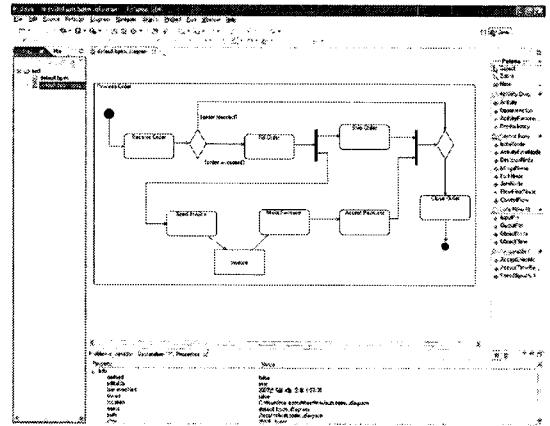


그림 7. 이클립스 기반 SOPLA모델링 도구의 인터페이스

즈니스 프로세스가 필수적이다. 본 연구에서는 유연한 비즈니스 프로세스 즉, 비즈니스 목표와 전략에 맞춘 비즈니스 프로세스를 프로덕트 라인 아키텍처의 재사용을 통해 신속하게 개발할 수 있도록 체계적으로 계획하여 정의된 SOPLA를 개발하였다.

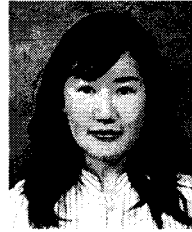
또한, SOPLA를 기반으로 비즈니스 프로세스 모델을 개발하는 프레임워크를 제시하였다. 프레임워크는 핵심자산인 SOPLA를 개발하는 도메인 공학 단계와 이를 재사용하여 비즈니스 요구에 맞춰 가변성을 바인딩하는 변환단계와 생성된 비즈니스 프로세스 모델을 운용하는 비즈니스 프로세스 공학 단계로 구성된다.

그리고, SOPLA를 설계하고 모델 작성을 지원하는 이클립스 플러그인 기반 도구를 개발하였다.

향후 연구 과제는 아키텍처 범위의 프레임워크를 확장하여 서비스 지향 프로덕트 라인 공학 프로세스를 개발하는 것이다. 이것은 핵심자산의 설계와 개발에서 비즈니스 프로세스의 운용 및 유지의 프로세스를 제시할 것이다. 아울러 SOPLA로부터 가변성을 바인딩하고 비즈니스 프로세스의 개발을 지원하도록 도구를 확장하는 것이다.

참 고 문 헌

- [1] 이상민, “비즈니스 신속성, SOA가 책임진다,” www.oracle.com/global/kr/magazine/archive/25summer/Trend_5.pdf.
- [2] Thomas Erl, Service-Oriented Architecture, concepts, technology, and design, Prentice Hall, 2005.
- [3] Software Engineering Institute(SEI), Carnegie Mellon University, Software Product Lines, <http://sei.cmu.edu/productlines/index.html>.
- [4] Brian, “A service oriented architecture allows better alignment with business processes,” http://builder.com.com/5100-6386_14-5069752.html.
- [5] Wite, S.A. Business Process Modeling Notation. BPMN 1.0, Business Process Modeling Initiative (2004)
- [6] BPML.org. Business Process Modeling Language (BPML). Accessed November 2002 from www.bpml.org, 2002.
- [7] OMG, XML Metadata Interchange, <http://www.omg.org/technology/documents/formal/xmi.htm>.
- [8] Eunsuk ye, Mikyeong Moon, Youngbong Kim, Keunhyuk Yeom, “An Approach to Designing Service-Oriented Product-Line Architecture for Business Process Families,” Proceedings of the 9th International conference on Advanced Communication Technology 2007, pp. 999-1002 Feb. 2007.
- [9] Curbera, F. et al. Business Process Execution Language for Web Services version 1.1 (2003)
- [10] Martin Becker, “XML-Enhanced Product Family Engineering,” Proceedings of the Sixth Biennial World Conference on Integrated Design and Process Technology (IDPT2002), June 2002, Pasadena, USA.



예 은 속

- 2003년 부경대학교 전자계산학과(학사)
- 2006년~현재 부산대학교 컴퓨터공학과 석사과정
- 관심분야 : 프로덕트 라인 공학, SOA, 비즈니스 프로세스 모델링 등



문 미 경

- 1990년 이화여자대학교 전자계산학과(학사)
- 1992년 이화여자대학교 전자계산학과(석사)
- 2005년 2월 부산대학교 컴퓨터공학과(박사)
- 2005년 3월~2005년 8월 부산대학교 차세대물류IT기술사업단 박사후 연구원
- 2005년 9월~2006년 8월 부산대학교 컴퓨터 및 정보통신연구소 기금교수
- 2006년 9월~현재 부산대학교 정보컴퓨터공학부 연구교수
- 관심분야 : 소프트웨어 프로덕트 라인, RFID기반 애플리케이션 개발모델, 적응형 소프트웨어 개발 등



홍 민 우

- 2007년 부산대학교 컴퓨터공학과(학사)
- 2007년~현재 부산 대학교 컴퓨터공학과 석사과정
- 관심분야 : 프로덕트 라인 공학, SOA, MDA 등



염 근 혁

- 1985년 서울대학교 계산통계학과(학사)
- 1992년 Univ. of Florida 컴퓨터공학과(공학석사)
- 1995년 Univ. of Florida 컴퓨터공학과(공학박사)
- 1985년~1988년 금성반도체 연구원
- 1988년~1990년 금성사 주임연구원
- 1995년~1996년 삼성SDS 책임연구원
- 1996년~현재 부산대학교 정보컴퓨터공학부 부교수
- 관심분야 : 소프트웨어 재사용, 프로덕트라인 공학, 서비스기반 비즈니스 프로세스, 상황인식 미들웨어, 적응형 소프트웨어 개발 방법 등