

# 비즈니스 프로세스 패밀리 모델을 이용한 비즈니스 프로세스 모델 자동 생성 도구

## (An Automatic Business Process Model Generation Tool Using Business Process Family Models)

홍민우<sup>†</sup>      문미경<sup>\*\*</sup>      염근혁<sup>\*\*\*</sup>  
(Minwoo Hong)      (Mikyeong Moon)      (Keunhyuk Yeom)

**요약** 오늘날 소비자의 요구사항이 빠르고 다양한 형태로 변화함에 따라 기업의 자산인 비즈니스 프로세스도 이러한 변화에 민첩하게 대처할 수 있는 능력이 요구되고 있다. 이러한 요구를 만족시키기 위해서는 비즈니스 프로세스 모델의 생성 단계에서부터 발생할 수 있는 변화에 대하여 분석 및 표현이 가능한 핵심 자산이 존재해야 하며 이러한 핵심자산의 재사용을 통해 특정 변화를 만족하는 비즈니스 프로세스 모델의 자동 생성이 가능해야 한다. 본 논문에서는 동일한 목적을 가지는 여러 비즈니스 프로세스 모델에 대하여 공통적인 특징을 공통성으로, 차별적인 특징을 가변성으로 분석하고 표현할 수 있는 비즈니스 프로세스 패밀리 모델을 설명한다. 그리고 비즈니스 프로세스 패밀리 모델의 공통성을 재사용하고 가변성에 대한 의사결정 정보를 반영하여 특정한 변화를 만족하는 비즈니스 프로세스 모델의 자동 생성을 지원하는 도구에 대하여 설명한다.

**키워드** : 프로덕트 라인 공학, 서비스 기반 아키텍처, 비즈니스 프로세스 모델, 핵심 자산 재사용

**Abstract** Nowadays, lots of requirements are changed quickly and variously. So, we need the ability to support the change of business process which is the asset of a company. To support this ability, the core asset that supports analyzing and describing about changes of business process should be exist from generating phase of a business process model. And a business process model that is satisfied the changed requirements should be automatically generated using core assets. In this paper, we present a business process family model (BPFM) which represents the commonalities and the variabilities of a set of business processes. we propose the automated tool using variabilities of BPFM and decision-making information for generation of business process model that is satisfied the changed requirements.

**Key words** : Product Line Engineering, Service Oriented Architecture, Business Process Model, Reusability of Core Assets

· 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음

- † 학생회원 : 부산대학교 컴퓨터공학과  
masterking@pusan.ac.kr
  - \*\* 정회원 : 동서대학교 컴퓨터정보공학부 전임강사  
mkmoon@dongseo.ac.kr
  - \*\*\* 종신회원 : 부산대학교 컴퓨터공학과 교수  
yeom@pusan.ac.kr  
(Corresponding author)
- 논문접수 : 2008년 4월 14일  
심사완료 : 2008년 7월 21일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저술물의 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 제8호(2008.8)

### 1. 서론

비즈니스 프로세스는 특정 시장 또는 고객들을 위한 명시적인 결과물의 생산을 최종 목적으로 하는 일련의 활동들이다[1]. 비즈니스 프로세스에 따라 기업은 보다 더 많은 이윤을 획득할 수 있고 적은 비용의 소비를 유지할 수 있으므로 비즈니스 프로세스가 기업의 중요한 자산임을 의미하기도 한다. 오늘날 시장 또는 고객의 요구사항이 다양하고 빠르게 변화하게 됨으로써 기업의 자산인 비즈니스 프로세스는 이러한 변화에 민첩하게 대응할 수 있는 능력을 제공할 필요가 있다.

비즈니스 프로세스의 실현을 지원하는 소프트웨어 분야에서 이러한 요구는 기존의 시스템 및 컴포넌트를 재

사용하고 통합하여 비즈니스 프로세스의 구현을 쉽게 접근 할 수 있게 하는 프레임워크인 서비스 기반 아키텍처에 대한 관심으로 나타나게 된다. 그러나 서비스 기반 아키텍처는 비즈니스 프로세스를 실현하는 소프트웨어의 재사용에 대해 초점을 맞추고 있을 뿐 실제적인 비즈니스 프로세스의 변화에 대해서는 고려하고 있지 않다.

현재, 변화를 고려한 소프트웨어 개발 방법론에는 소프트웨어 프로덕트 라인 공학이 존재한다. 소프트웨어 프로덕트 라인 공학에서는 소프트웨어 제품 군에 대한 공통적인 특징을 공통성으로 정의하고 차별적인 특징을 가변성으로 정의할 수 있도록 한다. 분석된 공통성과 가변성은 소프트웨어 개발 시 재사용되어 동일한 목적을 가지나 차별적인 특징을 보유한 소프트웨어의 생산을 가능하게 하여 짧은 시간 안에 적은 비용으로 품질이 높은 소프트웨어의 개발을 가능하게 한다. 하지만 소프트웨어 프로덕트 라인 공학은 컴포넌트 단위로 구성된 소프트웨어에 대한 공통성과 가변성을 다루므로 비즈니스 프로세스의 변화를 그대로 표현하기에는 불가능하다.

본 논문에서는 비즈니스 프로세스 군에 대한 공통성과 가변성을 정의할 수 있는 비즈니스 프로세스 패밀리 모델을 재사용하여 변화된 요구사항을 만족하는 비즈니스 프로세스 모델을 민첩하게 개발할 수 있는 자동 생성 도구에 대해 설명한다.

본 논문의 구성은 다음과 같다. 2절에서 관련연구로 서비스 기반 아키텍처와 소프트웨어 프로덕트 라인 공학에 대해 설명하며 3절에서는 비즈니스 프로세스 모델에 소프트웨어 프로덕트 라인 공학의 공통성과 가변성을 표현할 수 있는 비즈니스 프로세스 패밀리 모델에 대해 설명한다. 4절에서는 개발된 비즈니스 프로세스 패밀리 모델의 공통성을 재사용하고 가변성 정보를 특정 요구사항에 적합한 가변치로 결정하여 비즈니스 프로세

스 모델을 자동 생성하는 도구에 대해 설명하고 5절에서는 지능형 선반 시스템의 비즈니스 프로세스 패밀리 모델을 제시하고 이것을 이용하여 특정 요구사항에 맞는 비즈니스 프로세스 모델을 자동 생성한 사례연구를 제시한다. 마지막으로 6절에서는 결론 및 향후 연구 방향에 대해 설명한다.

## 2. 관련연구

### 2.1 서비스 기반 아키텍처

비즈니스 프로세스는 “비즈니스 행위들로 구성되고 특정 입력에 대해 연속적인 비즈니스 행위들의 수행을 통해 최종적인 결과를 도출하게 하는 과정”[2]을 의미한다. 이러한 비즈니스 프로세스의 실제화를 고려한 프레임워크(framework)에는 서비스 기반 아키텍처(Service Oriented Architecture: SOA)가 존재한다.

SOA는 분리된 비즈니스 프로세스와 서비스들을 쉽게 결합하고 재사용 할 수 있게 하는 어플리케이션 프레임워크이다[3]. 1996년 가트너(Gartner)그룹에 의해 최초 소개 되었고 현재는 소프트웨어 분야의 최신 경향 중 하나로 자리잡고 있다. 가트너그룹은 SOA에 대한 정의를 “잘 정의된 인터페이스들을 가진, 재사용이 가능한 일련의 컴포넌트들로 구축되는 기술구조 방식이다”라고 나타내고 있다[4]. 이것은 SOA가 느슨한 결합(loose coupling), 이 기중 플랫폼(platform)의 상호 운영성, 기존 시스템에 존재하는 기능의 재사용성과 새로운 기능을 기존 시스템에 존재하는 기능들을 서로 조합하여 생성할 수 있는 능력(composability)을 제공할 수 있다는 것이다.

결국 SOA는 특정 기술과 플랫폼에 독립적인 어플리케이션 또는 시스템의 개발을 지원하는 어플리케이션 아키텍처라고 말할 수 있다.

SOA는 그림 1과 같이 비즈니스 프로세스 영역, 비즈

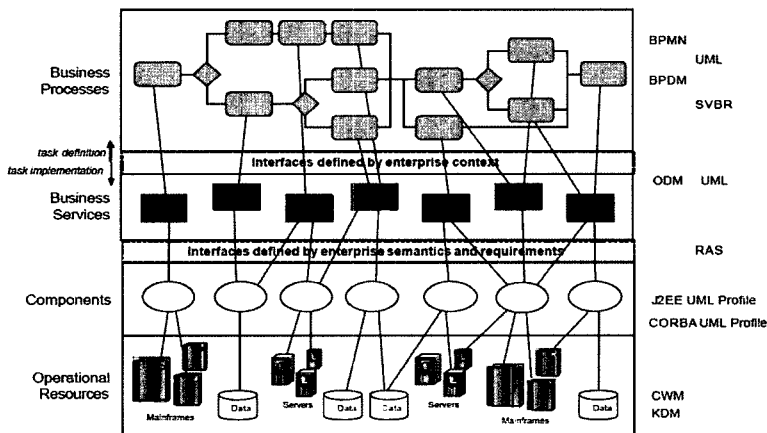


그림 1 서비스 기반 아키텍처[3]

니스 서비스 영역, 컴포넌트 영역과 동작 자원 영역으로 구성된다. 가장 상위의 비즈니스 프로세스영역은 기업이 가지는 비즈니스 프로세스를 가시적인 요소들로 표현한 비즈니스 프로세스 모델들로 구성되며 이러한 비즈니스 프로세스 모델은 Business Process Modeling Notation (BPMN)[5], Unified Modeling Language(UML)[6], Petri-net등을 사용하여 모델링 될 수 있다[7]. 비즈니스 서비스영역은 비즈니스 프로세스 모델에 존재하는 행위들을 수행하는 비즈니스 서비스로 구성된다. 비즈니스 서비스는 비즈니스 프로세스를 수행하는 최소 행위인 작업, 비즈니스 프로세스의 특정 흐름, 비즈니스 프로세스 내의 논리적인 의미로 추상화된 부 프로세스로 나타낼 수 있으며 현재는 웹 서비스(Web-Service) 기술을 사용하여 구현 가능한 단위이다. 이러한 비즈니스 서비스는 비즈니스 프로세스의 여러 영역에서 사용되는 부분이거나 독립적인 기능을 수행하는 부분으로 나타낼 수 있다. 비즈니스 서비스 영역을 구성하는 비즈니스 서비스는 컴포넌트 영역에 존재하는 컴포넌트들의 조합을 통해 실현 가능하며 컴포넌트들은 동작 자원 영역에 존재하는 서버, 메인프레임(mainframe), 데이터를 사용하여 특정한 요구사항을 만족시키는 동작을 수행한다. 결국 SOA는 기업의 비즈니스 프로세스에 대해 이를 실현하기 위해 기존에 존재하는 서비스의 재사용 또는 기존 컴포넌트의 재 조합을 통해 새로운 서비스 생성을 지원하여 기업의 비즈니스 프로세스의 실제화를 목표로 한다.

그러나 SOA는 비즈니스 프로세스를 실제화하기 위한 기존 자원의 재활용에 초점을 맞추고 있으므로 SOA의 구현기술인 웹 서비스 기술 영역의 변화에 대한 연구들은 많이 진행되고 있을 뿐[8,9], 비즈니스 프로세스 영역에서 발생할 수 있는 변화에 대한 대응방법에 대해서는 고려하고 있지 않다.

**2.2 소프트웨어 프로덕트 라인 공학**

소프트웨어 개발의 중요한 성공요소는 낮은 비용과 높은 품질을 가지는 소프트웨어를 시장에 빠른 시간에 공급하는 것(short time-to-market)이다[10]. 이러한 요구를 위해 제안된 소프트웨어 개발 방법론에는 소프트웨어 프로덕트 라인 공학(Software Product Line Engineering: SPLE)이 존재한다.

SPLE는 그림 2와 같이 크게 도메인 공학과 어플리케이션 공학이라는 두 가지의 생명주기로 구성된다. SPLE의 도메인공학에서는 특정 소프트웨어 제품 군에 존재하는 특징(feature)들을 분석하여 공통적으로 존재하는 특징은 공통성(commonality)으로 표현하고 차별적인 특징은 가변성(variability)으로 표현하여 이를 핵심 자산(core assets)으로 관리한다. 어플리케이션 공학에서는 특정 소프트웨어를 개발하기 위해 존재하는 핵심자산의

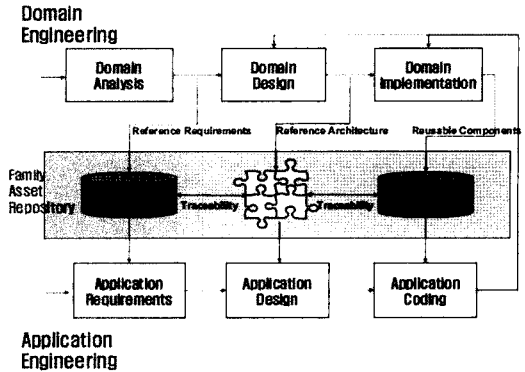


그림 2 소프트웨어 프로덕트 라인 공학[12]

정보를 이용하는데 핵심 자산의 공통성은 재사용하고 개발 소프트웨어가 가지는 차별적인 특징에 해당하는 가변치(Variants)를 핵심 자산의 가변성 선택 및 결정을 통해 획득한다. SPLE의 공통성과 가변성은 특정 소프트웨어 군에 해당하는 소프트웨어의 개발 시 공통성의 재사용을 통해 품질이 높은 소프트웨어 개발을 지원하며 가변성의 선택 및 결정을 통해 다양한 요구사항을 만족하는 소프트웨어를 빠른 시간 안에 개발할 수 있도록 지원한다[11,12]. 결국 비즈니스 프로세스가 다양한 변화에 민첩하게 대응하기 위해서는 특정 비즈니스 프로세스 군에 존재하는 공통성과 가변성의 분석 및 표현, 그리고 이를 이용한 가변성 선택 및 결정이 필요한 것이다.

하지만 일반적으로 SPLE는 컴포넌트 단위로 구성된 소프트웨어의 개발에 대해 초점을 두고 있어[13,14] 컴포넌트 보다 더 높은 추상적인 의미를 표현하는 비즈니스 프로세스에 SPLE의 개념을 그대로 적용하는 것은 불가능하다.

**3. 비즈니스 프로세스 패밀리 모델**

비즈니스 프로세스 패밀리 모델(Business Process Family Model: BPFM)은 UML2.0 Activity Diagram을 확장하여 특정한 목적을 가지는 비즈니스 프로세스 군의 특징을 공통성과 가변성으로 표현하는 기능을 지원한다[15]. 즉 BPFM을 통해 동일한 목적을 가진 비즈니스 프로세스 모델들에서 공통적으로 존재하는 프로세스 또는 흐름들을 공통성으로, 각각의 비즈니스 프로세스에 차별적으로 존재하는 기능 또는 흐름은 가변성으로 표현할 수 있다.

BPFM의 공통성과 가변성은 비즈니스 프로세스 모델의 개발에 재사용되며 이를 통해 빠르고 다양한 형태로 변화하는 비즈니스 프로세스에 대해 짧은 시간에 품질이 높은 비즈니스 프로세스 모델의 생성을 가능하게 한다. 그림 3은 특정 BPFM과 이를 사용하여 개발된 비즈니스

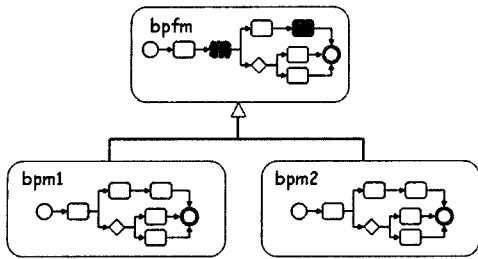


그림 3 BPFM과 비즈니스 프로세스 모델 간의 관계[10]

스 프로세스 모델간의 관계를 상속관계로 표현하고 있다. 즉 BPFM의 공통성은 이를 사용하여 개발된 비즈니스 프로세스 모델에 그대로 재사용되어 나타나며 BPFM의 가변성은 특정한 요구사항에 맞도록 더욱 구체적인 형태로 나타나게 됨을 의미한다.

BPFM에서 공통성은 UML2.0 Activity Diagram의 요소를 사용하여 표현하고 가변성은 2차원적 접근방법 [16]을 기반으로 1st-Level, 2nd -Level의 가변성으로 표현한다. 1st-Level 가변성은 비즈니스 프로세스 모델을 구성하는 부 프로세스와 작업에 대한 존재 유무의 가변성을 표현한다. 2nd -Level의 가변성은 부 프로세스와 작업에 대한 변화 지점을 가변점(VariationPoint)으로 표현하고 가변점의 정보에 따라 가변되는 형태를 지정할 수 있도록 한다. 제한적으로 부 프로세스와 작업은 하나의 가변점만을 가지고 있다. 가변점은 변화될 수 있는 가변치들과 연결관계를 통해 변화 가능 형태를 표현할 수 있다. 가변점을 가지는 프로세스는 가변점에 연결된 가변치로 변화 가능하거나 가변치들의 조합을 통해 새로운 비즈니스 프로세스 흐름으로 변화 가능하다. BPFM 모델 구성요소와 가변점 정보, 가변형태 및 속성에 대한 상세한 설명은 이후 4.1과 4.2절에서 제시하도록 하겠다.

#### 4. 비즈니스 프로세스 모델 자동 생성 도구

비즈니스 프로세스 모델 자동 생성 도구는 BPFM의 모델링 기능을 지원하고 개발된 BPFM을 재사용하여 비즈니스 프로세스 모델의 자동생성을 지원한다. 또한 개발된 비즈니스 프로세스 모델의 편집 및 생성 기능도 지원한다.

비즈니스 프로세스 모델 자동 생성 도구는 그림 4와 같은 구조를 가지며 이클립스(Eclipse)에서 구동된다. 비즈니스 프로세스 모델 자동 생성 도구에서 Business Process Family Modeling Tool(BPFMT)는 UML2.0 Activity Diagram을 확장하여 BPFM 생성 및 편집 기능을 지원한다. Decision Making Tool of Business Process Family Model(DMT-BPFM)은 개발된 BPFM

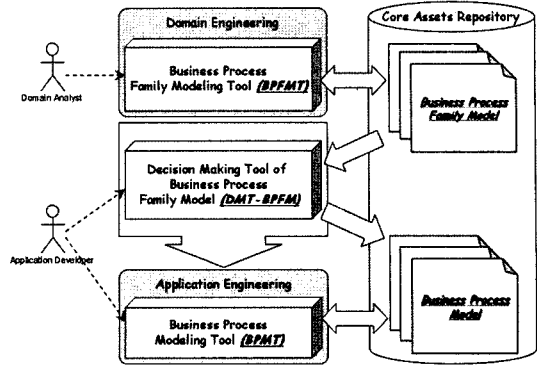


그림 4 비즈니스 프로세스 모델 자동 생성 도구 구성

의 공통성을 재사용하고 존재하는 가변성들에 대해 개발 대상인 비즈니스 프로세스의 요구사항에 적합한 형태로 선택하고 결정하여 비즈니스 프로세스 모델의 자동생성 기능을 지원한다. 최종적으로 생산된 비즈니스 프로세스 모델은 UML2.0 Activity Diagram을 사용하여 비즈니스 프로세스 모델을 표현 할 수 있는 Business Process Modeling Tool(BPMT)에 의해 편집 가능하다. 물론 DMT-BPFM을 사용하지 않고 BPMT를 통한 직접적인 비즈니스 프로세스 모델의 생성도 가능하다.

##### 4.1 Business Process Family Modeling Tool

BPFMT는 이클립스 Graphical Modeling Framework (GMF)[17]를 사용하여 개발 되었다. GMF는 구조화된 모델링 도구의 설계와 구현을 지원하므로 개발자들이 손쉽게 모델링 도구 개발에 접근할 수 있으며 GMF를 통해 개발된 산출물들은 GMF로 다른 모델링 도구의 개발 시 재사용이 용이하므로 빠른 시간에 품질이 높은 모델링 도구의 개발을 가능하도록 지원한다. 이러한 GMF의 장점을 통해 BPFMT에서 개발된 개발 산출물을 4.2절에서 설명할 BPMT의 개발에 재사용해 BPMT의 개발 시간과 비용을 절감할 수 있다.

BPFM의 구성요소들을 구조적으로 정의한 메타 클래스 다이어그램은 GMF에서 지원하는 산출물 형태인 Ecore모델을 통해 표현하였다. Ecore 모델은 모델링 도구에 사용될 모델 요소의 의미적인 관계를 표현한 것이며 BPFM의메타 클래스 다이어그램은 그림 5와 같이 Ecore모델로 정의된다. 단 그림 5에서는 제한적으로 전체적인 BPFM의 메타 클래스 다이어그램이 아닌 가변성 요소의 확장을 위해 사용된 클래스들의 구조를 강조하기 위해 축약된 메타 클래스 다이어그램이다. BPFM에서 공통성은 UML2.0 Activity Diagram의 의미와 기호를 그대로 사용하므로 그림 5와 같이 흰색으로 표현된 메타 클래스들로 정의한다. 가변성은 가변성의 의미를 표현할

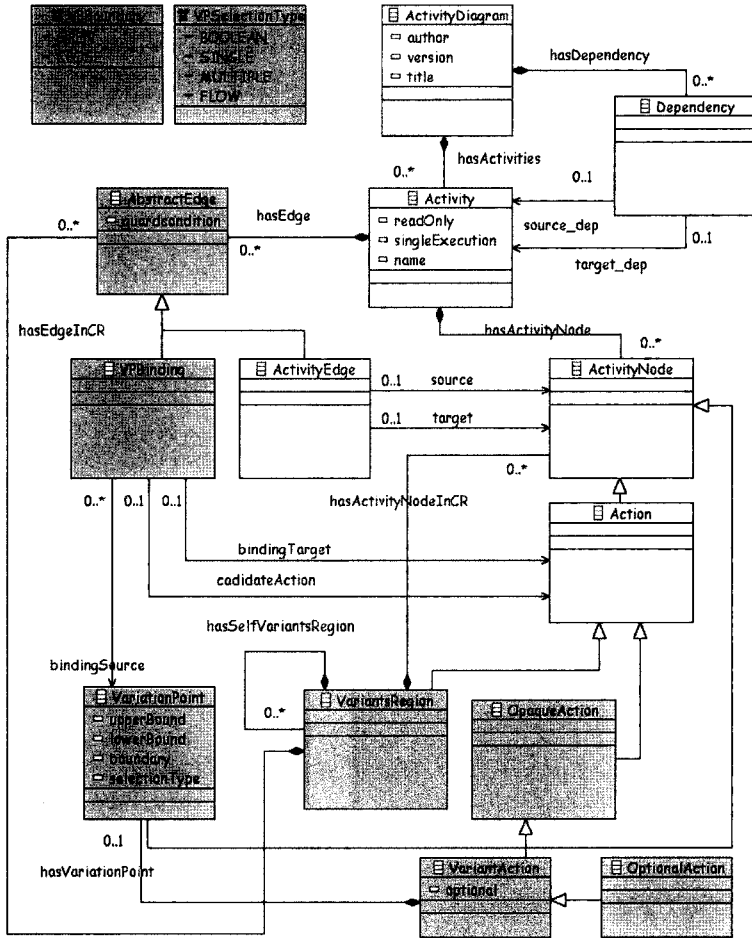


그림 5 UML2.0의 Activity Diagram의 메타 클래스들을 확장한 BPFM의 메타 클래스 다이어그램

수 있는 클래스들을 추가적으로 정의하여 그림 5와 같은 회색으로 표현된 메타 클래스들로 나타낸다.

BPFM에서 표현 가능한 가변성은 3절에서 언급한 것과 같이 크게 1st-Level 가변성과 2nd-Level 가변성으로 구분된다. 1st-Level 가변성은 단지 부 프로세스 또는 작업에 대한 존재유무에 대한 변화만을 나타내므로 존재유무의 변화를 표현할 수 있는 Optional Action 클래스를 그림 5와 같이 추가 정의한다. 2nd-Level 가변성은 VariationPoint 메타 클래스, Variants Region 메타클래스와 VPBinding 메타 클래스에 의해 실제화된다. 2nd-Level 가변성을 정의하기 위해서는 부 프로세스 또는 작업을 의미하는 Opaque Action 클래스가 변화 가질 수 있는 변화에 대한 표현이 가능해야 하고 BPFM에서는 이것을 가변점 사용을 통해 표현하므로 가변점에 대한 메타클래스를 VariationPoint로 정의하였다. 그리고 Opaque Action의 변화는 하나의 가변점만을 통

해 표현되어야 함으로 VariationPoint 클래스와 Opaque Action 클래스는 0..1 다양성(Cadinality)으로 정의된 합성관계 (Composite)를 가진다. 결국 가변점은 변화 가능성에 관련된 모든 정보를 정의할 수 있어야 하며 이것은 가변성 선택 및 결정시에 필요한 의사결정 정보를 제공하기 위한 기준 정보가 된다.

VariationPoint 메타 클래스는 표 1에 정의된 속성을 통해 변화에 대한 정보를 표현 가능하게 한다. VariationPoint의 속성 값의 정의는 BPFM을 통한 비즈니스 프로세스 모델의 자동 생성시에 가변성의 결정 정보로 사용되며 속성값의 구체적인 의미는 4.2절의 가변성 유형을 통한 결정방법에서 자세히 제시하도록 하겠다.

모든 BPFM의 모델 요소는 고유한 속성을 가지고 있으며 BPFMT는 각각의 모델 요소의 속성에 대한 값의 지정을 지원하여 BPFM의 가시적 모델 이외의 부가적인 모델 정보에 대한 기술을 가능하게 한다. VPBinding

표 1 VariationPoint 속성

명칭	설명	Data Type
Boundary	추가적인 가변치의 정의 가능 여부	{open, close}
Lower Bound	최소 선택 가변치 수	unsigned integer
Upper Bound	최대 선택 가변치 수	unsigned integer
Selection Type	가변성 선택 유형	{boolean, single, multiple, flow}

메타 클래스는 가변점을 포함한 Opaque Action이 변화될 수 있는 비즈니스 프로세스 흐름 또는 작업단위를 나타내는 가변치들간의 관계를 표현한다. BPFM에서 비즈니스 프로세스의 작업에 대한 가변치는 Action 메타 클래스의 자식 클래스들을 통해 표현이 가능하다. 하지

만 Action 메타클래스의 자식클래스만을 사용하여 특정 범위의 비즈니스 프로세스 흐름을 표현하는 것은 불가능하므로 특정 범위의 비즈니스 프로세스 흐름을 정의할 수 있는 VariantsRegion 메타 클래스를 정의한다. Variants Region은 Composite Pattern을 사용하여 Action 메타 클래스의 자식 클래스들을 내부에 표현할 수 있고 VariantsRegion 자체도 내부에 표현하게 위한 모델링 구성 요소를 나타낸다.

BPFMT의 개발은 BPFMT Ecore 모델을 기반으로 [16]에서 제시된 모델링 지원 도구 개발 절차에 따라 GMF를 사용하여 이클립스로 구동 가능한 플러그인 (Plug-In) 형태로 개발되었고 그림 6과 같은 구동 모습을 보여준다.

표 2 BPFM 모델 요소

모델 요소	설명	Notation
OptionalAction	1st -Level 가변성	
Variation point	2nd -Level 가변성의 가변점	
VPBinding	가변점과 가변치간의 관계	
VariantsRegion	비즈니스 프로세스의 흐름을 표현한 가변치	
UML2.0 Activity Diagram 모델 요소	비즈니스 프로세스의 공통성 표현. 17가지 UML 2.0 Activity Diagram 모델 요소로 구성 (Activity, Opaque Action, Activity Parameter, Dependency, InitialNode, ActivityFinalNode, DecisionNode, MergeNode, ForkNode, JoinNode, FlowFinalNode, InputPin, OutputPin, ObjectNode, ObjectFlow, ControlFlow, AcceptTimeEventAction, AcceptEventAction, SendSignal Action)	UML 2.0 Activity Diagram과 동일

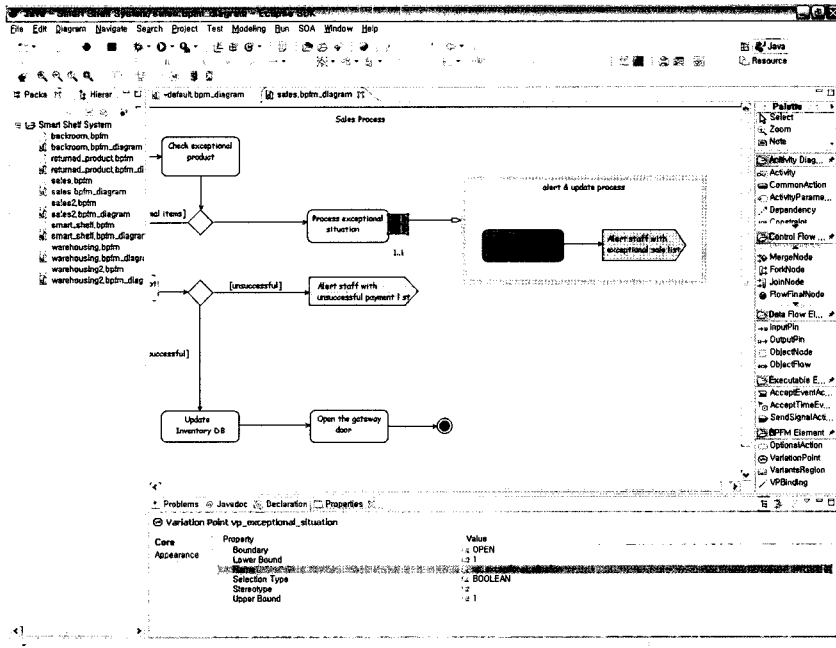


그림 6 BPFM 모델링 도구: BPFMT

#### 4.2 Decision-Making Tool of Business Process Family Model

DMT-BPFM은 개발된 BPFM의 공통성을 재사용하고 가변성을 특정 요구사항에 맞게 선택 및 결정하여 비즈니스 프로세스 모델의 자동 생성을 지원한다.

DMT-BPFM은 재사용될 핵심 자산인 BPFM의 저장 위치를 입력 받아 해당 BPFM의 가변성 정보를 추출한다. 추출된 가변성 정보에 대한 결과는 그림 7의 Tree View1에 표시된다. 그림 7은 추출된 가변성 정보들을 표시할 뿐만 아니라 추출된 가변성 정보에 대한 선택 및 결정을 지원한다. 의사 결정(Decision-Making)에 대한 결과는 Tree View2에 표시되며 Tree View1과 Tree View2에 존재하는 모든모델 정보에 대한 속성들의 값은 Table View에 표시 된다. Tree View1에 존재하는 모든 가변성 정보에 대한 의사결정이 이루어졌을 경우 비즈니스 프로세스 모델의 자동 생성이 가능해진다. DMT-BPFM은 지정된 BPFM의 가변성 정보에 대한 의사결정을 지원하여 비즈니스 프로세스 모델의 자동 생성을 지원한다.

3 절에서 제시된 BPFM의 1st-Level, 2nd-Level 가변성 유형, 2nd-Level 가변성의 가변점 정보 중 의사결정에 필요한 선택 유형(selection type) 값에 따라 DMT-BPFM에서는 비즈니스 프로세스 모델의 자동 생성을 위한 다른 의사 결정 정보를 사용자가 정의할 수 있도록 한다. 사용자의 의사결정 정보의 입력 후, DMT-BPFM은 입력된 의사 결정 정보에 맞는 특정 패턴으로 비즈니스 프로세스 모델을 자동 생성한다. 4.2.1 절에서는 1st-Level가변성, 4.2.2절에서는 2nd-Level가변성의 의사 결정에 따른 비즈니스 프로세스 모델의 생

성에 대해 설명한다.

##### 4.2.1 1st-Level 가변성의 의사 결정에 따른 비즈니스 프로세스 모델 생성

BPFM의 1st-Level 가변성은 3절에서 제시한 것과 같이 비즈니스 프로세스 모델을 구성하는 부 프로세스와 작업에 대한 존재 유무의 변화를 나타낸다. 그러므로 BPFM에서 1st-Level 가변성을 나타내는 Optional Action에 대한 의사 결정 정보는 해당 Optional Action을 비즈니스 프로세스 모델의 생성시 포함 할 것인지의 여부가 된다. 이러한 의사 결정 정보에 따라 1st-Level 가변성은 그림 8과 같은 비즈니스 프로세스 모델 생성 패턴을 가진다.

그림 8의 Pattern A는 BPFM에 존재하는 Optional Action을 사용하는 의사 결정을 통해 해당 Optional Action이 Opaque Action으로 변환된 비즈니스 프로세스 모델 생성 결과를 나타낸다. 그리고 Pattern B는 Optional Action을 사용하지 않는다는 의사 결정을 통해 Optional Action이 제거된 비즈니스 프로세스 모델의 생성 결과를 보여준다.

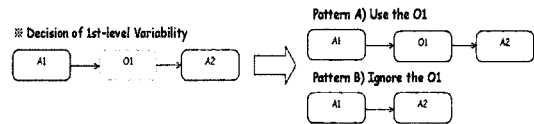


그림 8 1st-Level 가변성 결정에 의해 생성 가능한 비즈니스 프로세스 모델 생성 패턴

##### 4.2.2 2nd-Level 가변성의 의사 결정에 따른 비즈니스 프로세스 모델 생성

BPFM의 2nd-Level가변성은 가변점의 정보에 의해 가변성의 선택 및 결정시에 필요한 정보를 정의하고 이는 변화 가능한 비즈니스 프로세스 모델 생성에 사용된다. 2nd-Level가변성은 크게 4가지 형태의 선택 유형으로 분류되고 DMT-BPFM은 가변점에 정의된 선택 유형에 따른 의사 결정 정보를 사용자에게 요구한다. 표 3은 2nd-Level가변성의 선택 유형과 선택 유형에 따른 필요한 의사결정 정보를 나타낸다. 표 3과 같이 2nd-Level가변성의 선택 유형은 크게 BOOLEAN, SINGLE, MULTIPLE, FLOW로 분류되며 선택 유형이 MULTIPLE일 경우 추가 선택 유형은 sequence, parallel, condition으로 세분화 된다.

· *BOOLEAN selection type & 2nd-level Variability*  
 선택 유형이 BOOLEAN인 경우는 가변점에 연결된 가변치를 자동 생성할 비즈니스 프로세스 모델의 생성시 사용여부에 대한 의사 결정 정보가 필요하다.

단 BPFM이 BOOLEAN을 선택유형으로 가지는 가

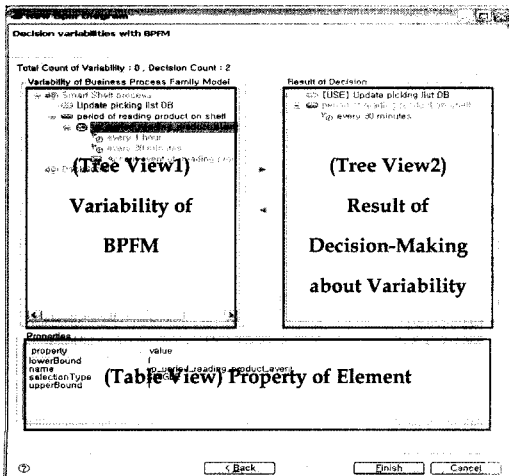


그림 7 DMT-BPFM의 BPFM의 가변성 선택 및 결정 UI(User Interface)

표 3 2nd-Level가변성의 선택 유형과 선택 유형에 따른 필요한 의사 결정 정보

가변성 유형 & 선택 유형	필요 의사 결정 정보
2nd-Level & <i>BOOLEAN</i>	가변치 사용여부 (use, ignore)
2nd-Level & <i>SINGLE</i>	사용할 하나의 가변치 선택
2nd-Level & <i>MULTIPLE</i>	<ul style="list-style-type: none"> <li>· 사용될 다수의 가변치 선택</li> <li>· 선택된 가변치들의 구성방법 선택                             <ul style="list-style-type: none"> <li>- sequence: 순서 정보</li> <li>- parallel: 필요 결정 정보 없음</li> <li>- condition: 조건 정보</li> </ul> </li> </ul>
2nd-Level & <i>FLOW</i>	<ul style="list-style-type: none"> <li>· VariantsRegion에 정의된 모델 요소를 사용한 비즈니스 프로세스 흐름 모델링 필요</li> <li>· 흐름의 시작과 끝 요소 지정.</li> </ul>

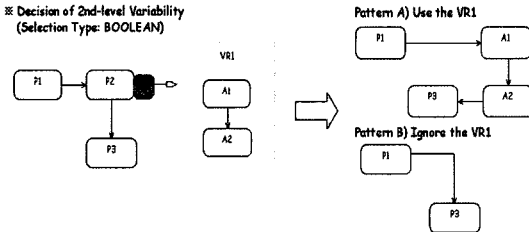


그림 9 BOOLEAN: 2nd-Level 가변성의 비즈니스 프로세스 모델 생성 패턴

변점에 연결될 수 있는 가변치의 수를 1개로 제한하기 때문에 BOOLEAN의 의사결정 정보는 특정 가변치의 사용여부만을 나타낸다.

그림 9는 BOOLEAN을 선택유형으로 가지는 가변성의 비즈니스 프로세스 모델 생성 패턴을 보여준다. 이전의 1st-Level 가변성과 같이 선택 유형이 BOOLEAN인 2nd-Level 가변성은 그림 9와 같이 크게 해당 가변치를 사용한 패턴(Pattern A)과 사용하지 않은 패턴(Pattern B)로 비즈니스 프로세스 모델의 자동 생성을 가능하게 한다. 하지만 1st-Level 가변성과의 차이점은 선택 유형이 BOOLEAN인 2nd-Level 가변성은 특정 작업 단위뿐만 아니라 특정 영역의 비즈니스 프로세스 흐름에 대한 사용여부를 VariantsRegion을 가변치로 표현하여 나타낼 수 있다. 그림 9는 VariantsRegion을 사용하여 A1의 수행 후 A2를 수행하는 비즈니스 프로세스 흐름의 사용여부를 BOOLEAN 유형을 통해 표현한 것이다.

• *SINGLE selection type & 2nd-level Variability*

선택 유형이 SINGLE인 경우는 가변점을 가지는 Opaque Action이 특정 가변치로 변환 될 수 있는 변화를 표현한다. 즉 BPFM에서 SINGLE 선택 유형인 가변점에 연결된 여러 개의 가변치들은 가변점을 포함한

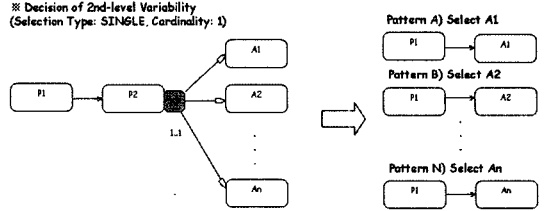


그림 10 SINGLE: 2nd-Level 가변성의 비즈니스 프로세스 모델 생성 패턴

Opaque Action의 변환 가능 후보들을 표현하며 하나의 Opaque Action은 정의된 가변치들 중 하나와 1:1로 변환 가능하다. 그러므로 선택 유형이 SINGLE인 2nd-Level 가변성은 가변점에 연결된 가변치들 중 비즈니스 프로세스 모델에 사용할 가변치에 대한 선택을 필요로 한다.

그림 10은 SINGLE 2nd-Level 가변성의 비즈니스 프로세스 모델 생성 패턴을 보여준다. 그림 10과 같이 가변점을 포함하는 P2는 A1에서 An 중 하나의 가변치로 변환 될 수 있으며 변환 결과들은 Pattern A에서 Pattern N까지 가능하다. 즉 선택 유형이 SINGLE인 가변성에 연결된 가변치의 수에 따라 생성 가능한 비즈니스 프로세스 모델의 수가 결정된다.

• *MULTIPLE selection type & 2nd-level Variability*

선택 유형이 MULTIPLE인 경우는 사용 가능한 여러 개의 가변치들을 정의하고 특정 수의 가변치들의 사용을 통해 일정한 의미를 가지는 비즈니스 프로세스 흐름을 표현한 모델의 생성이 필요할 시 사용된다. 그러므로 MULTIPLE을 선택 유형으로 가지는 2nd-Level 가변성의 의사결정 정보는 가변점에 연결된 가변치들 중 사용할 가변치에 대한 선택 정보와 선택된 가변치들의 일정한 형태에 맞게 구성되도록 하는 구성 방법에 대한 정보가 존재한다.

DMT-BPFM은 이러한 구성 방법을 sequence, parallel, condition 3가지로 정의하고 각 구성 방법에 따라 필요한 의사 결정 정보를 추가로 사용자에게 요구한다. sequence의 경우는 선택된 가변치들을 일정한 순서에 따라 일렬화 시킨 것을 의미하며 이를 위해 선택된 가변치들 간의 순서 정보가 추가로 필요하다. parallel의 경우는 선택된 가변치들을 동시에 실행 가능하도록 하는 것을 의미하며 추가적인 의사 결정 정보는 존재하지 않는다. 마지막으로 condition 구성 방법은 선택된 각각의 가변치들이 특정 조건을 만족할 때 수행 가능하도록 하는 구성방법이며 각각의 가변치의 수행 조건이 추가적인 의사 결정 정보가 된다.

그림 11은 선택 유형이 MULTIPLE인 2nd-Level 가변성의 비즈니스 프로세스 모델 생성 패턴을 보여준다.



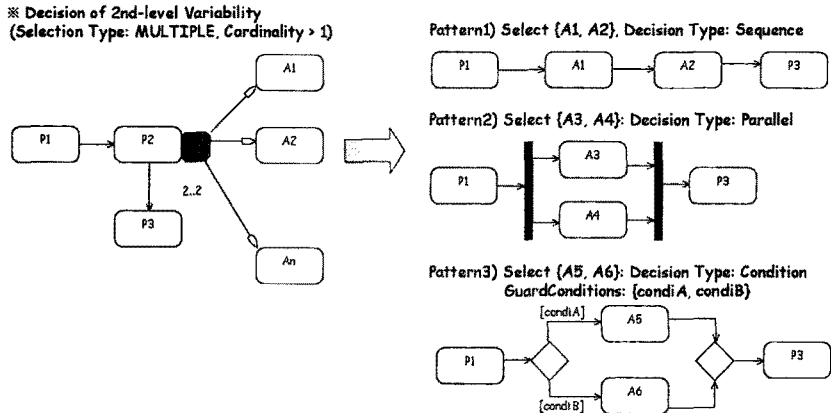


그림 11 MULTIPLE: 2nd-Level 가변성의 비즈니스 프로세스 모델 생성 패턴

그림 11과 같이 선택 유형이 MULTIPLE인 2nd-Level 가변성은 구성 방법에 따라 3가지 형태의 비즈니스 프로세스 모델의 생성이 가능하다. DMT-BPFM은 우선 가변점에 연결된 가변치들 중 사용할 가변치들의 선택을 사용자에게 요구한다. 단 선택 가능한 가변치들의 최소 선택 수는 가변점 정보의 Lower Bound의 값보다 커야 하며 최대 선택 수는 Upper Bound의 값보다 작아야만 한다. 사용 가변치들에 대한 선택 후, 선택된 가변치들의 구성 방법인 sequence, parallel, condition 중 하나의 선택이 필요하다.

그림 11과 같이 sequence가 선택된 경우는 선택된 가변치들의 순서 정보를 결정하고 선택된 가변치들을 ControlFlow를 통해 연결한 비즈니스 프로세스 모델이 생성된다. 즉 A1이 처음 순서이고 A2가 그 다음이라는 순서 정보의 정의를 통해 A1의 수행 후 A2가 수행되는 비즈니스 프로세스 모델이 생성 가능하다. parallel의 경우는 ForkNode와 JoinNode를 사용하여 선택된 가변치들의 동시 수행을 표현하는 하며 추가적인 결정 정보는 필요하지 않다. 단 가변점을 포함한 Opaque Action을 목표로 가진 흐름이 없는 경우(P1→P2의 흐름이 없는 경우) ForkNode가 없는 패턴으로 비즈니스 프

로세스 모델이 생성되며 반대로 가변점을 포함한 Action에서 출발하는 흐름이 없는 경우(P2→P3흐름이 없는 경우) JoinNode가 없는 비즈니스 프로세스 모델이 생성된다. 마지막으로 condition은 특정 조건에 의해 선택된 가변치들을 실행하는 구성 방법을 표현하므로 추가적인 결정 정보에는 각각의 선택 가변치들의 실행 조건이 존재한다. 즉 조건 분기를 DecisionNode로 표현하며 각각의 선택된 가변치의 실행 조건은 ControlFlow의 GuardCondition에 표현된다. 그리고 각각의 가변치들은 실행 후 MergeNode에 의해 결합된다.

• FLOW selection type & 2nd-level Variability

FLOW인 경우는 가변점에 대응하는 VariantsRegion 내부의 모델 요소를 재사용하여 의사 결정자가 원하는 모델을 그림 12와 같이 정의할 수 있다. 단 BPFM의 VariantsRegion내부에 정의된 요소들에 대한 변경은 불가능하며 내부에 존재하는 요소들을 재사용하여 특정 목적을 달성하는 비즈니스 프로세스의 흐름에 대한 모델링을 수행해야만 한다.

그림 12와 같이 VariantsRegion에 존재하는 A1과 A2를 재사용하고 Opaque Action의 추가를 통해 특정한 비즈니스 프로세스를 모델링 하거나 조건적인 행위

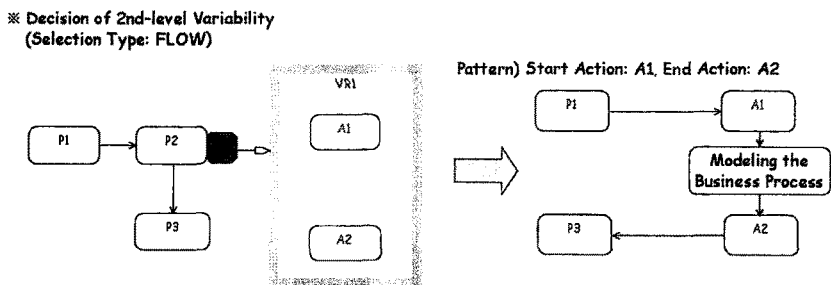


그림 12 FLOW: 2nd-Level 가변성의 비즈니스 프로세스 모델 생성 패턴

의 수행을 표현하는 DecisionNode를 사용한 모델링, 이벤트 흐름을 나타내기 위해 AcceptEvent Action, Send-SignalAction등을 사용한 모델링을 통해 FLOW 선택 유형을 가진 2nd-Level 가변성을 결정할 수 있다. 추가적인 비즈니스 프로세스 모델링 후, 해당 비즈니스 프로세스 모델의 시작과 끝에 해당하는 모델 요소에 대한 지정이 마지막으로 필요한 의사 결정 정보가 된다.

결국 DMT-BPFM에서 FLOW 선택 유형을 가지는 2nd-Level가변성은 가변점에 연결된 VariantsRegion 내부 요소를 재사용한 사용자 임의의 비즈니스 프로세스 모델링 후, 모델링 된 결과에 존재하는 비즈니스 프로세스 모델의 시작과 끝을 표현하는 모델요소의 선택을 통해 사용자가 원하는 비즈니스 프로세스 모델의 자동 생성을 지원한다.

DMT-BPFM은 위에서 언급한 4가지의 선택 유형의 의사 결정 입력 UI를 사용자에게 제공하여 선택 유형에 따라 필요한 의사 결정 정보를 사용자에게 요구한다. 그림 13은 선택 유형이 SINGLE인 2nd-Level 가변성의 의사 결정 정보 입력 UI에서 존재하는 가변치 중 사용할 가변치를 선택할 수 있는 리스트 화면을 나타낸다.

그리고 선택 유형이 MULTIPLE인 2nd-Level 가변성의 condition 구성 방법에 대한 의사 결정 정보 입력 UI는 그림 14와 같으며 그림 14는 현재 사용할 가변치에 대한 수행 조건 입력 화면을 나타낸다.

### 4.3 Business Process Modeling Tool

BPMT는 BPFMT의 공통성과 가변성 모델정보를 재사용하여 생성되거나 어플리케이션 개발자에 의해 생성된 비즈니스 프로세스 모델에 대한 편집 기능을 지원한다. BPMT는 그림 5에서 제시한 BPFMT의 메타 클래스 다이어그램에서 가변성 부분을 제외한 공통성 부분에 해당하는 메타 클래스들을 기반으로 개발된다. 그리고 표 2에서 제시된 UML2.0 Activity Diagram 모델 요소만 사용하여 비즈니스 프로세스 모델링 기능을 지원한다. BPMT는 BPFMT과 같이 이클립스에서 동작하는 플러그인 형태로 개발되었다.

## 5. 사례 연구

본 절에서는 지능형 선반 시스템(Smart Shelf System)의 BPFM을 개발하고 개발된 BPFM을 통해 비즈니스 프로세스 모델의 자동생성 사례를 제시한다.

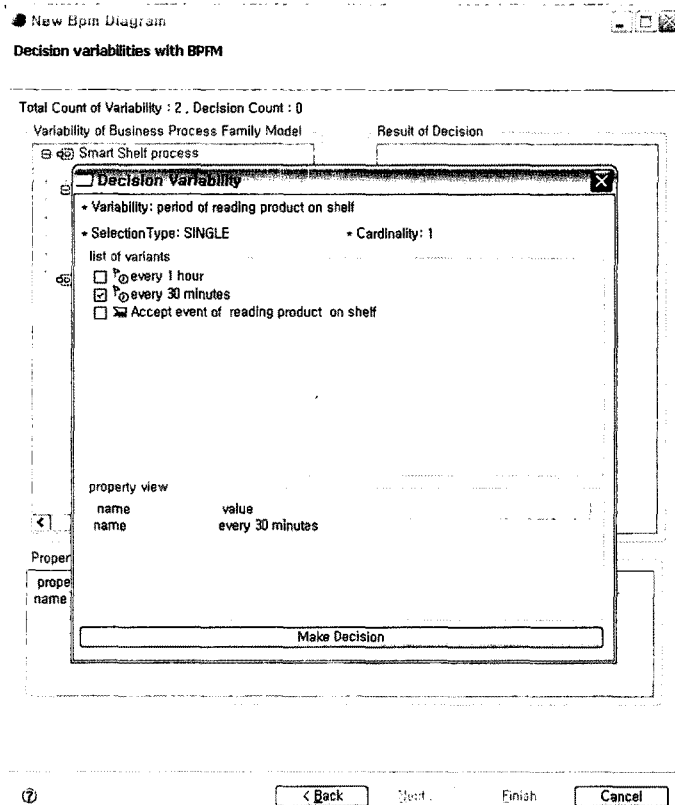


그림 13 SINGLE: 2nd-Level 가변성 유형의 의사 결정 정보 입력 UI

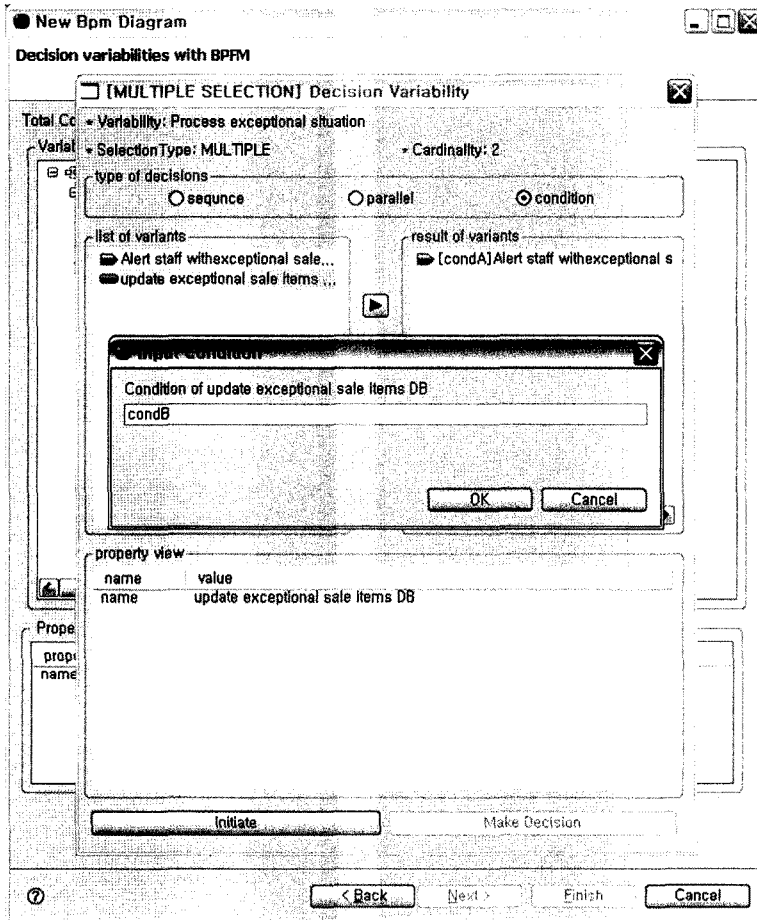


그림 14 MULTIPLE: 2nd-Level 가변성 유형의 condition 구성 방법에 대한 의사 결정 정보 입력 UI

지능형 선반 시스템은 크게 선반의 상태를 주기적으로 확인하는 smart shelf process, 창고에 존재하는 제품을 관리하는 warehousing process, 매장의 제품 보관소의 상태를 관리하는 backroom process, 반품 제품에 대한 처리를 담당하는 returned product process 그리고 제품 판매 흐름을 담당하는 sales process로 구성된다. 지능형 선반 시스템의 프로세스 중 smart shelf process의 공통성과 가변성을 표현한 BPFM은 그림 15와 같이 나타난다.

Smart shelf process는 주기적으로 선반에 존재하는 제품 정보를 획득하고 제품의 수량을 확인하여 선반에 기존 제품 수량보다 적은 제품 수량이 존재할 시 선반에 제품 부족 상태를 직원에게 알려 일정한 제품 수량이 선반에 존재하도록 하는 프로세스이다. 하지만 이러한 일반적인 흐름 이외에 smart shelf process는 선반에 존재하는 제품의 판매 예상 주기나 매장의 영업 방침에 따라 선반에 존재하는 제품 수량을 검사하는 주기는 변화

될 수 있다. 한 예로 특정 매장에서 식품의 판매가 저녁 시간에 가장 활발하게 이루어진다면 식품이 배치된 선반은 특정 저녁 시간 동안에 선반의 상태를 검사하는 주기를 30분 단위로 빠르게 하여 선반에 일정 수량의 식품이 계속적으로 배치될 수 있게 하고 그 이외의 시간대는 1시간의 선반 검사 주기를 줄 수 있는 것이다.

그림 15의 smart shelf process에 대한 BPFM에서는 두 가지의 가변성이 존재하는데 하나는 선반에 존재하는 제품의 상태를 검사하는 주기의 변화를 하지만 이러한 일반적인 흐름 이외에 smart shelf process는 선반에 존재하는 제품의 판매 예상 주기나 매장의 영업 방침에 따라 선반에 존재하는 제품 수량을 검사하는 주기는 변화될 수 있다.

한 예로 특정 매장에서 식품의 판매가 저녁시간에 가장 활발하게 이루어진다면 식품이 배치된 선반은 특정 저녁 나타내는 가변성과 제품의 부족상태를 데이터 베이스에 저장하는 가변성이다. 두 가지의 가변성을 제외

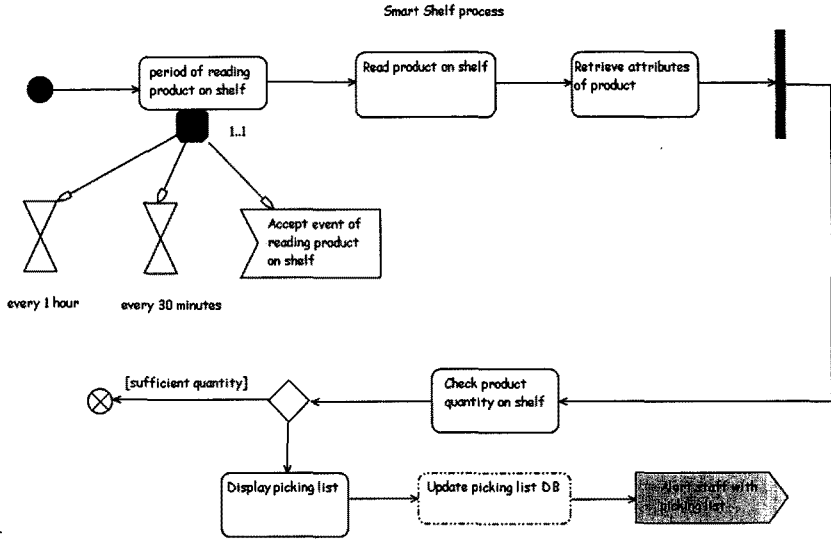


그림 15 smart shelf process의 BPFM

한 나머지 비즈니스 프로세스 모델 요소는 공통성으로 표현되고 있다. 그림 16은 DMT-BPFM를 통해 공통성을 재사용하고 가변성에 대한 선택 및 결정을 수행하여 선반의 정보를 30분 단위로 검사하고 제품 부족 상태를 데이터 베이스에 저장하지 않고 바로 직원에게 제품 부족 정보를 전달하는 자동 생성된 smart shelf process 모델을 나타낸다.

뿐만 아니라, 그림 15의 smart shelf process의 BPFM을 이용하여 그림 17과 같이 특정 이벤트에 의해 선반의 제품 상태를 확인하며 제품 수량 부족 정보를 저장하는 smart shelf process 모델을 자동 생성도 가능하다.

### 6. 결론

본 논문에서는 빠르고 다양하게 변화하는 요구사항을 기업의 자산인 비즈니스 프로세스에 대응하기 위하여 비즈니스 프로세스 패밀리 모델을 이용한 비즈니스 프로세스 모델 자동 생성 도구를 제시하였다.

비즈니스 프로세스 패밀리 모델을 이용한 비즈니스 프로세스 모델 자동 생성 도구는 기업의 자산인 비즈니스 프로세스에 대한 모델링뿐만 아니라 비즈니스 프로세스에서 발생할 수 있는 변화를 BPFM의 가변성으로 정의할 수 있게 하여 변화의 대응에 대한 준비를 사전에 가능하도록 한다. 그리고 개발된 BPFM의 공통성을

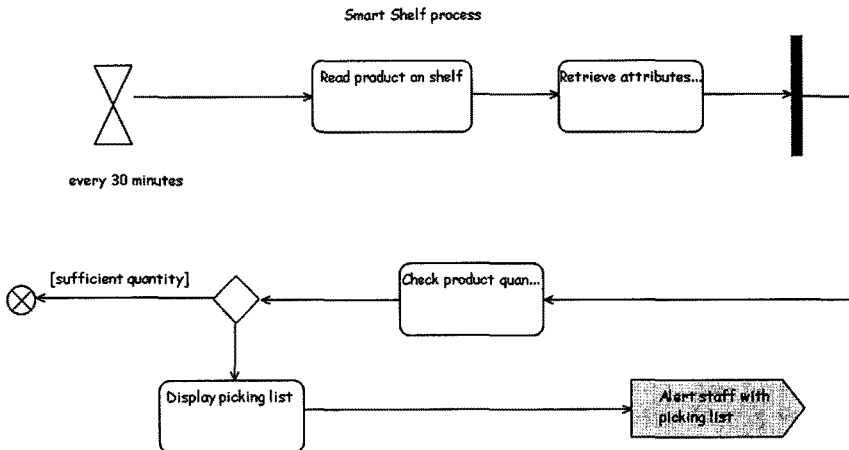


그림 16 smart shelf process BPFM을 이용한 자동 생성 비즈니스 프로세스 모델 유형1

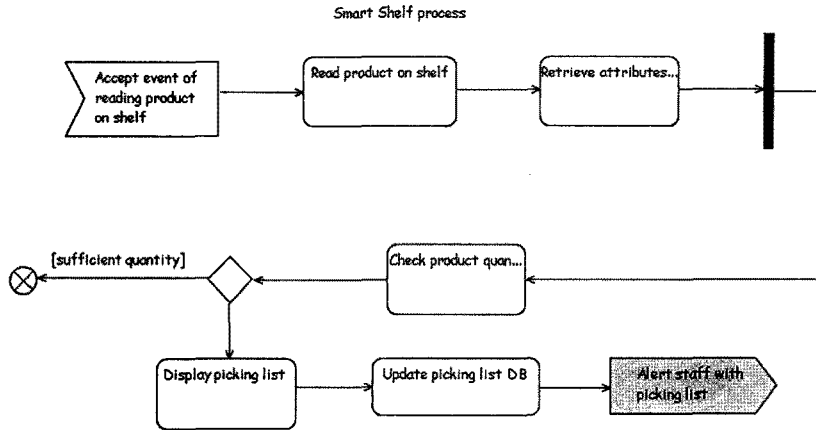


그림 17 smart shelf process BPFM을 이용한 자동 생성 비즈니스 프로세스 모델 유형2

재사용하고 가변성을 변화에 적합한 형태로 선택 및 결정하여 자동으로 비즈니스 프로세스 모델의 생성을 지원하여, 동일한 목적을 가지나 차별화된 특징을 가지는 다양한 비즈니스 프로세스 모델들을 적은 비용과 빠른 시간 안에 개발 할 수 있게 한다.

BPFMT와 BPMT는 UML2.0 Activity Diagram으로 비즈니스 프로세스에 대한 모델링을 지원하고 있다. 현재 IBM에서는 UML Activity Diagram을 사용하여 웹 서비스의 호출 흐름을 모델링하고 이러한 정보를 Business Process Execution Language(BPEL)로 변환하는 방법[18]을 제시하였다. 향후에는 이러한 접근 방법을 적용하여 비즈니스 프로세스 정보의 가변성과 공통성에 대한 재사용을 분석과 설계 단계뿐만 아니라 구현단계에도 적용할 수 있도록 개발된 비즈니스 프로세스 모델에 대한 BPEL 변환 기능을 BPMT에 추가 구현할 것이다. 또한 핵심자산의 효율적 관리를 지원하기 위해 가변성 정보, 가변성에 대한 선택 및 결정정보, 자동 생성된 비즈니스 프로세스 모델 정보에 대한 추적성 제공 기능을 지원할 것이다.

### 참 고 문 헌

[1] IBM Service-Oriented Modeling and Architecture, [www.ibm.com/developerworks/webservices/library/ws-soa-design1/](http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/)

[2] Service Oriented Architecture, [http://www.adobe.com/it/enterprise/pdfs/Services\\_Oriented\\_Architecture\\_from\\_Adobe.pdf](http://www.adobe.com/it/enterprise/pdfs/Services_Oriented_Architecture_from_Adobe.pdf)

[3] The OMG and Service Oriented Architecture, <http://www.omg.org/attachments/pdf/OMG-and-the-SOA.pdf>

[4] Service-Oriented Architecture Scenario, [http://www.gartner.com/DisplayDocument?doc\\_cd=114358](http://www.gartner.com/DisplayDocument?doc_cd=114358)

[5] Object Management Group, *Unified Modeling*

*Language specification ver 2.0*, 2003.

[6] Business Process Management Initiative Home, "Introduction to BPMN," <http://www.bpmn.org/>

[7] Vergidis K., Tiwari A., Majeed B., "Business Process Analysis and Optimization: Beyond Re-engineering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol.38, Issue.1, pp. 69 - 82, 2008.

[8] Gomaa H., Saleh M., "Software product line engineering for Web services and UML," *The 3rd ACS/IEEE International Conference on 2005*, pp. 110.

[9] N. Y. Topaloglu, R. Capilla, "Modeling the Variability of Web Services from a Pattern Point of View," *European Conference on Web Services (ECOWS2004)*, LNCS Springer-Verlag, 2004, pp. 128-138.

[10] Product Line Development, <http://www.ito.tu-darmstadt.de/publs/pdf/TR-PLD.pdf>

[11] Pohl, Klaus, Böckle, Günter, Linden, Frank J. van der, *Software Product Line Engineering*, Springer, 2005.

[12] 윤석진, 이승연, 신규상, 양영중, 박창순, "가변성을 고려한 핵심 자산의 결정 모델 기술 및 지원도구", *한국정보과학회 가을 학술 발표논문집*, Vol.31, No.2, pp. 466-468, 2004.

[13] Hassan Gomma, "Designing Software Product Lines with UML," Addison Wesley, 2004.

[14] Weiss, D.M., Lai, C.T.R., "Software Product-Line Engineering: A Family Based Software Development Process," Addison-Wesley, ISBN 0-201-694387, 1999.

[15] 홍민우, 문미경, 염근혁, "비즈니스 프로세스의 가변성 모델링 지원 도구 개발", *한국정보과학회 가을 학술 발표논문집*, Vol.34, No.2(A), pp. 52-53, 2007.

[16] 문미경, 채홍석, 염근혁, "도메인 핵심자산의 가변성 분석을 위한 2차원적 접근방법", *정보과학회 논문지 소프트웨어 및 응용 제33권 제6호* pp. 550-563, 2006.

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

[18] From UML to BPEL, <http://www.ibm.com/developerworks/webservices/library/ws-uml2bpel/>



홍 민 우

2007년 2월 부산대학교 정보컴퓨터공학부(학사). 2007년 3월~현재 부산대학교 컴퓨터공학과(석사). 관심분야는 소프트웨어 프로덕트 라인 공학, 서비스 기반 소프트웨어 개발, 모델 기반 소프트웨어

개발



문 미 경

이화여자대학교 전자계산학과(학사). 이화여자대학교 전자계산학과(이학석사). 부산대학교 컴퓨터공학과(공학박사). 2005년 3월~2005년 8월 부산대학교 차세대 물류IT기술사업단 박사후 연구원. 2005년 9월~2006년 8월 부산대학교 컴퓨터 및 정보통신 연구소 기금교수. 2006년 9월~2008년 2월 부산대학교 정보컴퓨터공학부 연구교수. 2008년 3월~현재 동서대학교 컴퓨터정보공학부 전임강사. 관심분야는 소프트웨어 프로덕트 라인 공학, 적응형 소프트웨어 개발, RFID 미들웨어 개발 및 RFID 기반 애플리케이션 개발 등



염 근 혁

1985년 2월 서울대학교 계산통계학과(학사). 1992년 8월 Univ. of Florida 컴퓨터공학과(석사). 1995년 8월 Univ. of Florida 컴퓨터공학과(박사). 1985년 1월~1988년 2월 금성반도체 컴퓨터연구실 연구원. 1988년 3월~1990년 6월 금성사 정보기기연구소 주임연구원. 1995년 9월~1996년 8월 삼성SDS 정보기술연구소 책임연구원. 1996년 8월~현재 부산대학교 정보컴퓨터공학부 교수. 관심분야는 소프트웨어 재사용, 프로덕트라인 공학, 서비스기반 소프트웨어 개발, 적응형 소프트웨어 개발, RFID기반 상황인식 미들웨어 등