

# A Tool for Workflow-based Product Line Software Development

Jin-Seok Yang<sup>†</sup> · Kyo C. Kang<sup>††</sup>

## ABSTRACT

A convergence software development methodology based on product line engineering provides an architecture model for application development and it also provides workflow as a behavior specification of control component development to develop transaction centric application. To effect a change on software development based on product line engineering it has to be supported by a tool. But almost workflow modeling tools dose not support product line engineering concept. So we need new workflow modeling tool to support the convergence software development methodology. In this paper, we introduce a toolset for workflow modeling that consists of eclipse plug-in applications and open source tool and describe the relationships of tools through example.

**Keywords :** Product Line Engineering, Convergence Software, Workflow, Architecture Model, CASE, VULCAN, Workbench

## 워크플로우 기반의 제품라인 소프트웨어 개발 지원 환경

양진석<sup>\*</sup> · 강교철<sup>\*\*</sup>

### 요약

제품라인공학 기반의 융합소프트웨어 개발 방법론에서는 어플리케이션 개발을 위해 아키텍처 모델을 제안하고 있다. 그리고 명세를 이용하여 제어컴포넌트를 개발하도록 제안하는데 워크플로우는 트랜잭션을 주로 처리하는 어플리케이션 개발을 위해 제안되었다. 제품라인 공학 기반의 소프트웨어 개발이 효과를 발휘하기 위해서는 도구의 지원이 반드시 필요하다. 하지만 기존의 워크플로우 모델링 도구들은 제품라인 공학 개념을 지원하지 않기 때문에 워크플로우 기반의 제품라인 소프트웨어 개발을 지원할 수 있는 도구의 개발이 필요했다. 본 논문에서는 워크플로우 기반의 제품라인 소프트웨어 개발을 지원하기 위해서 개발된 도구를 소개하고 개발된 도구의 활용 가능성을 확인하기 위해서 간단한 활용예제를 소개한다.

**키워드 :** 제품라인공학, 융합소프트웨어, 워크플로우, 아키텍처 모델, CASE, 벌컨, 워크벤치

### 1. 서론

최근 들어 융합소프트웨어가 강조되고 점차 많은 기업들이 세계시장으로 진출을 모색하고 있다. 그 결과 소프트웨어 개발에 반영해야 하는 휘처(Feature)는 증가하고 그에 따라 발생하는 다양성을 소프트웨어에 반영하기 위한 노력을 기울이고 있다.

소프트웨어 제품라인 공학(Software Product Line Engineering)은 어플리케이션을 각각 개별적으로 개발하는

것이 아니라, 유사한 기능을 제공하는 어플리케이션들을 함께 개발 하도록 하여 위와 같은 문제를 해결할 수 있는 개발 패러다임을 제시하고 있다. 개발자는 개발 대상 어플리케이션의 기능 및 비기능 측면을 고려하여 공통점과 차이점 분석을 통해서 핵심 자산(Core Asset)을 확보하고 자산들을 조합하여 원하는 어플리케이션을 개발할 수 있다[1].

포항공과대학교 융합소프트웨어개발센터(CoSDEC)는 이러한 기술적 사회적 변화에 대응 할 수 있도록 K. C. Kang 이 제안한 FORM(Feature Oriented Reuse Method)[2]를 기반으로 하는 제품라인공학 기반의 융합소프트웨어 개발 방법을 제안하고, 방법론을 지원하는 워크벤치(Workbench) 벌컨(VULCAN)의 전반적인 개념을 제안하였다[3].

융합소프트웨어 제품라인 방법론에서는 어플리케이션 개발을 위한 아키텍처 모델을 제안하고(Fig. 1 참조) 어플리케이션의 유형에 따라서 명세 기반으로 자동화된 제어컴포넌트 개발이 될 수 있도록 제어컴포넌트의 행위를 명세할 수 있는 모두 네 가지의 방법을 제안하고 있다[4]. 그 가운데

\* 이 논문은 한국정보처리학회 제38회 추계 학술발표 대회에서 '워크플로우 기반의 제품라인 소프트웨어 개발 지원환경'의 제목으로 발표된 논문을 확장한 것임.

\*\* 이 논문은 정보통신산업진흥원의 SW공학 요소기술 연구개발 사업에 의해 지원되었음.

† 정 회 원 : 포항공대 정보통신 연구소 소프트웨어공학 연구실 연구원

†† 중신회원 : 포항공과대학교 교수

논문접수 : 2012년 12월 24일

수정일 : 1차 2013년 3월 13일

심사완료 : 2013년 3월 13일

\* Corresponding Author : Jin-Seok Yang(edeward@gmail.com)

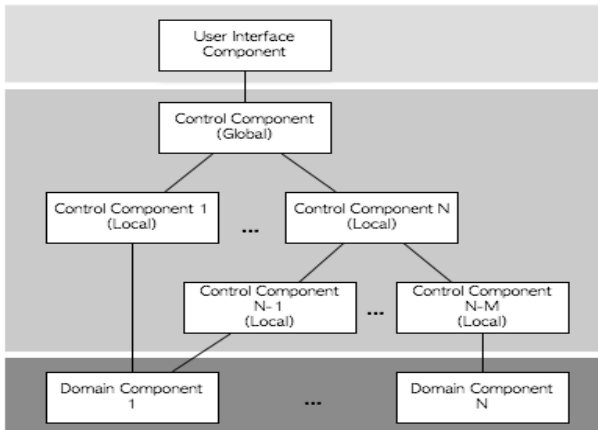


Fig. 1. Architecture Model for Convergence Software

워크플로우는 트랜잭션을 주로 처리하는 어플리케이션의 제어 행위를 명세하기 위해서 제안되었다.

워크플로우 명세를 기반으로 제품라인 소프트웨어의 제어 컴포넌트 개발이 효과를 진행되기 위해서는 도구의 지원이 반드시 필요하지만, 기존의 워크플로우 모델링 도구들은 제품라인 공학 개념을 지원하지 않았다[5][6]. 그래서 우리는 융합소프트웨어 제품라인 방법론에서 제안하는 워크플로우 기반의 소프트웨어 개발을 지원하기 위해서 개발된 도구들을 제안한다. 개발된 도구들은 다른 도구들과 함께 워크벤치 별전을 구성하는 요소가 되었다.

본 논문은 다음과 같이 구성된다. 2장에서는 워크플로 기반의 제품라인 소프트웨어 개발도구에 대한 구성에 대해서 소개한다. 3장에서는 간단한 사례연구를 통해서 도구의 활용가능성을 살펴본다. 마지막 장에서는 결론 및 향후연구에 대해서 소개한다.

## 2. 워크플로우 기반의 제품라인 소프트웨어 개발도구

우리는 앞장에서 소개된 아키텍처 모델을 지원하면서 휘저 중심의 제품라인 공학 개념이 반영된 최대한으로 단순한 워크플로우 기반 개발도구를 만들기를 원했다. 그 결과 도구 개발을 위해서 결정된 기능은 다음과 같다.

- UML2.0 표기법을 활용한 최소 워크플로우 행위명세 지원
- 제어컴포넌트와 도메인 컴포넌트와의 결합을 위한 방법 제공
- 휘저기반 워크플로우 행위 명세의 파라미터화 (Parameterization)
- 휘저모델-아키텍처-워크플로우 행위 명세 사이의 일관성 검사
- 워크플로우 행위 명세 기반의 코드 생성

그리고 위의 기능을 제공하기 위해서 개발된 개발 지원도구는 Fig. 2와 같이 다섯개로 구분될 수 있다. 개발지원도구는 ‘워크플로우 다이어그램 편집기’를 제외한 나머지 도구들

은 모두 이클립스 IDE 플랫폼을 기반의 이클립스 플러그인 어플리케이션으로 개발되었다. ‘워크플로우 다이어그램 편집기’의 경우 UML 2.0 표기법을 지원하는 다이어그램의 편집 기능만 제공하면 되므로 오픈소스 도구인 StarUML 5.0을 이용하였다[7].

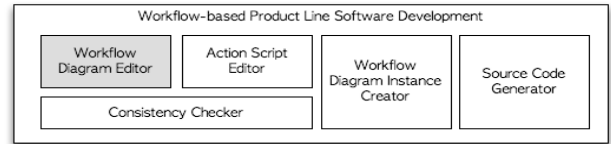


Fig. 2. Conceptual Architecture of tool

개발자들은 워크플로우 기반의 제품라인 소프트웨어의 제어 컴포넌트 개발을 위해 Fig. 3과 같은 흐름으로 지원도구들을 사용할 수 있다.

개발자는 제품라인 소프트웨어의 자산(Asset)을 만드는 제품라인 공학의 도메인 엔지니어링 (Domain Engineering) 과정에서 대상 소프트웨어의 제어 컴포넌트의 행위명세를 작성하기 위해서 ‘워크플로우 다이어그램 편집기’와 ‘액션 스크립트 편집기’를 이용할 수 있다. 이 도구들을 이용하여 개발자는 가변요소가 포함된 제품라인 워크플로우 다이어그램 (PL Workflow Diagram)과 특정 액티비티가 활성화 될때 실행되는 도메인 컴포넌트를 연결하는 액션 스크립트 (Action Script)를 작성할 수 있다. 워크플로우를 이용하여 행위명세를 작성할때 개발자는 워크플로우의 액티비티 (Activity)와 흐름(Flow)에 UML 스테레오타입(Stereotype)

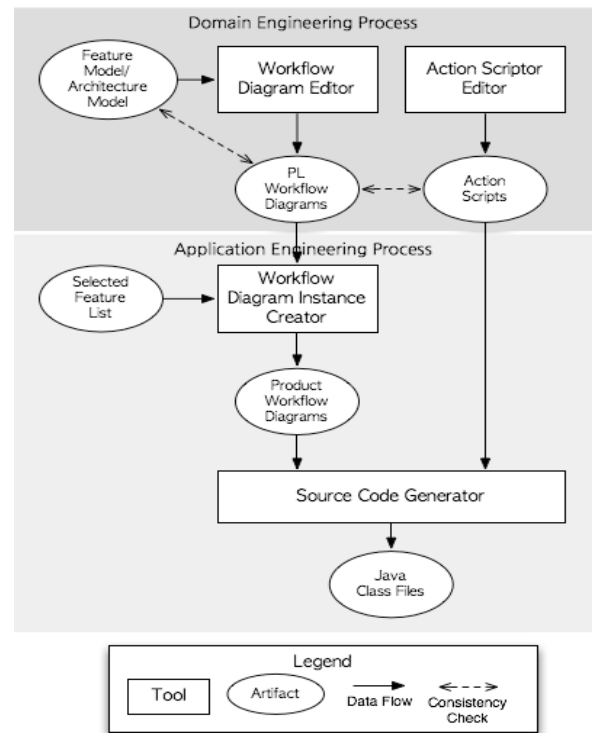


Fig. 3. Tool Supporting in product line engineering process

을 이용하여 휘처이름과 AND, OR 연산자를 이용한 간단한 논리식을 만듦으로써 가변요소로 만들 수 있다. 이렇게 정의된 논리식은 이후 어플리케이션 엔지니어링 과정에서 휘처선택의 결과에 따라 참/거짓이 결정되며, 참으로 판정된 요소들만 인스턴스 생성시 사용된다. 논리식 작성과 관련된 제한사항은 본 논문의 범주에서 벗어나기 때문에 여기에서 자세히 다루지 않는다.

‘일관성 검사기’를 통해 개발자는 작성된 워크플로우 다이어그램에 오류가 있는지에 대해서 확인 할 수 있다. 이 도구는 제품라인 워크플로우 다이어그램의 검증과 제품 워크플로우 다이어그램의 검증에 모두 사용된다.

전자의 경우, 도구는 워크플로우 행위명세에 포함된 가변성 정보에 대한 검증을 진행한다. 이 검증 과정에서 휘처모델, 아키텍처 모델, 그리고 워크플로우 다이어그램을 입력으로 받아, 앞서 설명한 워크플로우의 가변정보를 설정하기 위해서 사용된 휘처와의 일관성 검사를 진행한다. 휘처모델-아키텍처-워크플로우 행위 명세 사이에 일관성 검사는 현재 아래의 네 가지 기본적인 규칙을 통해서 이뤄진다. 첫째, 모든 휘처는 적어도 한번은 아키텍처 구성 요소 및 워크플로우 행위명세에 연결되어야 한다. 둘째, 아키텍처를 구성하는 모든 구성요소는 휘처에 바인딩 되어 있어야 한다. 셋째, 아키텍처와 워크플로우 행위명세에 연결된 휘처들로부터 도출되는 휘처들 사이의 계층적 관계는 휘처모델의 휘처들 사이의 계층적 관계와 동일해야 한다. 넷째, 대체적 관계의 휘처는 같은 요소에 연결될 수 없다. ‘일관성 검사기’는 위의 규칙들에 대한 검사를 수행한다.

후자의 경우, 도구는 비즈니스 프로세스 모델에서 검증 대상으로 삼는 문법적 오류(Syntax Error), 구조적 오류(Structural Error), 의미적 오류(Semantic Error)[8]에 대해서 검증을 지원한다. 문법적 오류는 정의된 워크플로우 모델의 표기법들에 대한 오류로 워크플로우 다이어그램 편집도구에서 제공하는 사용 제한으로 오류를 검출할 수 있는 반면에 구조적 오류와 의미적 오류는 워크플로우에 내재된 논리상의 오류로 반드시 확인이 필요하다. 하지만 본 도구는 일단 데드락(Deadlock), 결정적인 무한루프(Deterministic Infinite Loop), 동기화 손실(Lack of Synchronization), 종료 불가능 상태(Activity without Termination), 비활성 상태(Activity without Activation)과 같은 구조적 오류[9]에만 초점을 둔다. 이와 같은 구조적 오류를 검증하기 위해서 현재 [9][10]와 같이 다양한 방법이 연구되고 있지만 구조적 오류 검증에 가장 효율적인 방법인 페트리 넷(Petri Net)을 사용한다[11]. 이와 같이 행위모델과 휘처모델, 아키텍처모델 사이의 일관성 검사 및 행위모델 인스턴스의 논리적인 오류검사를 지원하며 검사가 완료된 행위명세는 자산으로써 관리된다.

자산으로부터 제품을 만드는 제품라인 공학의 어플리케이션 엔지니어링 과정에서 개발자는 ‘워크플로우 다이어그램 인스턴스 생성기’를 이용해 휘처모델로부터 선택된 휘처목록과 제품라인 워크플로우 다이어그램을 입력으로 제품 워크플로우 다이어그램을 생성할 수 있다. 생성된 제품 워크플로우 다이어그램은 휘처선택에 따른 가변요소에 대

한 파라미터화가 적용된 제품라인 워크플로우 다이어그램의 인스턴스(Instance)이다. 인스턴스의 예는 3장에서 확인할 수 있다.

지금까지 도구의 구성과 도구들 사이의 관계에 대해서 설명했다. 이어서 개발된 도구가 어떻게 앞서 제시한 기능들을 만족하고 있는지 좀 더 자세하게 살펴본다.

도메인 엔지니어링과정에 사용되는 도구는 제어컴포넌트의 행위모델을 작성하고, 명세된 행위모델에서 특정 태스크가 활성화 되었을때 도메인 컴포넌트(도메인 컴포넌트는 메뉴얼하게 개발된다고 가정한다.)를 사용 또는 적용 할 수 있도록 도와준다. 이를 구현하기 위해서 도구에서는 UML2.0의 요소들 가운데 State Diagram과 State의 액션(Action)을 활용하고, 도메인 컴포넌트와의 결합을 위한 액션 스크립트(Action Script)를 추가로 제공한다. 이들 사이의 관계는 Fig. 4와 같다. 개발자는 State의 do액션속성에 exec(script\_name) 구문을 이용하여 활용할 액션 스크립트를 설정할 수 있다. 그리고 액션 스크립트의 connection.service(...) 구문을 이용해 특정 클래스의 메소드를 실행할 수 있도록 할 수 있다.

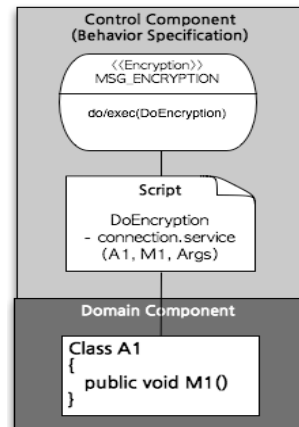


Fig. 4. A relationship between workflow behavior specification and architecture model

정의된 액션 스크립트는 크게 내부 변수 선언 부분과 로직을 작성할 수 있는 부분으로 구분되어 있으며 로직을 작성시 사용되는 타입과 구문은 자바(Java) 언어를 따르며 구문안에 @문자로 시작되는 매크로(Macro)를 사용할 수 있다. 정의된 매크로는 ‘코드생성기’를 통해서 특정 구문으로 치환이 된다. 매크로가 사용된 액션 스크립트의 예는 다음과 같다.

```
//@VAR
String orderID;
Boolean result;
//@BODY
Vector<String> args = new Vector<String>();
orderID=(String)@GETVALUE("orderID");
result = (Boolean)connector.service("userManager",
"Validate", args);
```

위의 액션 스크립트 예에서 매크로 @GETVALUE를 이용하여 orderID의 값을 반환받는 구문과 connection.service()를 이용하여 도메인 컴포넌트의 구현체인 UserManager 클래스의 Validate 메소드를 호출하는 것을 확인할 수 있다.

파라미터화 방법을 적용하여 워크플로우 행위모델의 인스턴스를 생성하기 위해서 다이어그램을 구성하는 State에 회차를 연결(Binding)할 수 있는 방법으로 앞서 설명한 것과 같이 스테레오타입(Stereotype)을 사용한다. Fig. 4에서 MSG\_ENCRYPTION는 스테레오타입에 Encryption 회차가 연결되어 있다. 어플리케이션 엔지니어링 과정에서 선택된 회차집합에 Encryption 회차가 없다면 해당 회차가 연결된 모든 State와 State에 연결된 모든 Sequence Flow는 인스턴스를 생성하는 과정에서 사라지고 그에 따라서 야기 될 수 있는 도달할 수 없는 State (Dead State)까지 사라지게 된다.

마지막으로, ‘코드생성기’는 제품 워크플로우 다이어그램과 액션 스크립트를 이용해 자바 소스코드를 자동으로 생성한다. 다이어그램의 구성요소는 대응되는 클래스로 생성되며, 액션 스크립트 역시 전처리과정을 거친 후 하나의 클래스로 생성을 시킨다.

지금까지 워크플로우 기반의 제품라인 소프트웨어 개발을 지원하는 도구에 대해서 전반적으로 설명하였다. 다음장에서는 간단한 활용예제 통해서 개발된 도구가 어떻게 사용되는지 소개한다.

### 3. 활용예제

우리는 도구의 활용모습을 보여주기 위해서 간단한 물품 구매 소프트웨어를 활용 예제로 사용한다.

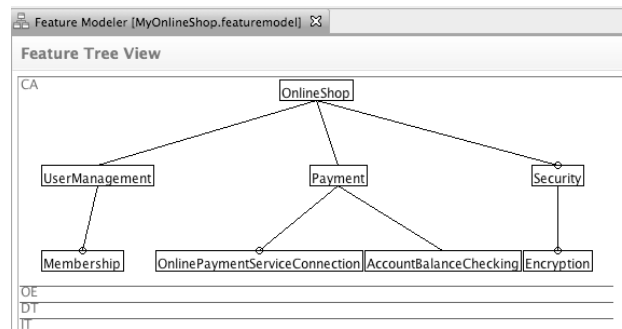


Fig. 5. A part of feature model for shopping mall

Fig. 5는 제품라인 기반으로 개발되는 물품구매 소프트웨어의 회차모델을 보여준다. Fig. 5의 회차모델에서 알 수 있듯이 가변요소는 Membership, Online Payment Service Connection, 그리고 Security 및 Encryption 회차로 정의되어 있다. 정의된 네 개의 가변 회차는 모두 선택적(Optional) 회차들이다.

융합소프트웨어 제품라인 방법론에 정의된 FORM-UML 아키텍처 모델링[13]을 수행하였다. (Fig. 6 참조. FORM-UML 아키텍처 모델링 방법은 FORM에서 정의된 네가지 아키텍처 모델을 UML의 표기법을 빌려서 모델링하는 방법으로 본 논문의 범위에서 벗어나므로 자세히 설명하지 않는다.)

도출된 컴포넌트들 가운데 두 개의 제어 컴포넌트 PurchaseController와 BankConnectionController에 대해 본 논문에서 제시한 워크플로우 모델을 이용하여 각 컴포넌트의 행위명세를 진행하였다. 아래의 Fig. 7은 Bank ConnectionController의 행위명세를 보여주며, 내부적으로 Encryption과 Online Payment Service Connection 선택적

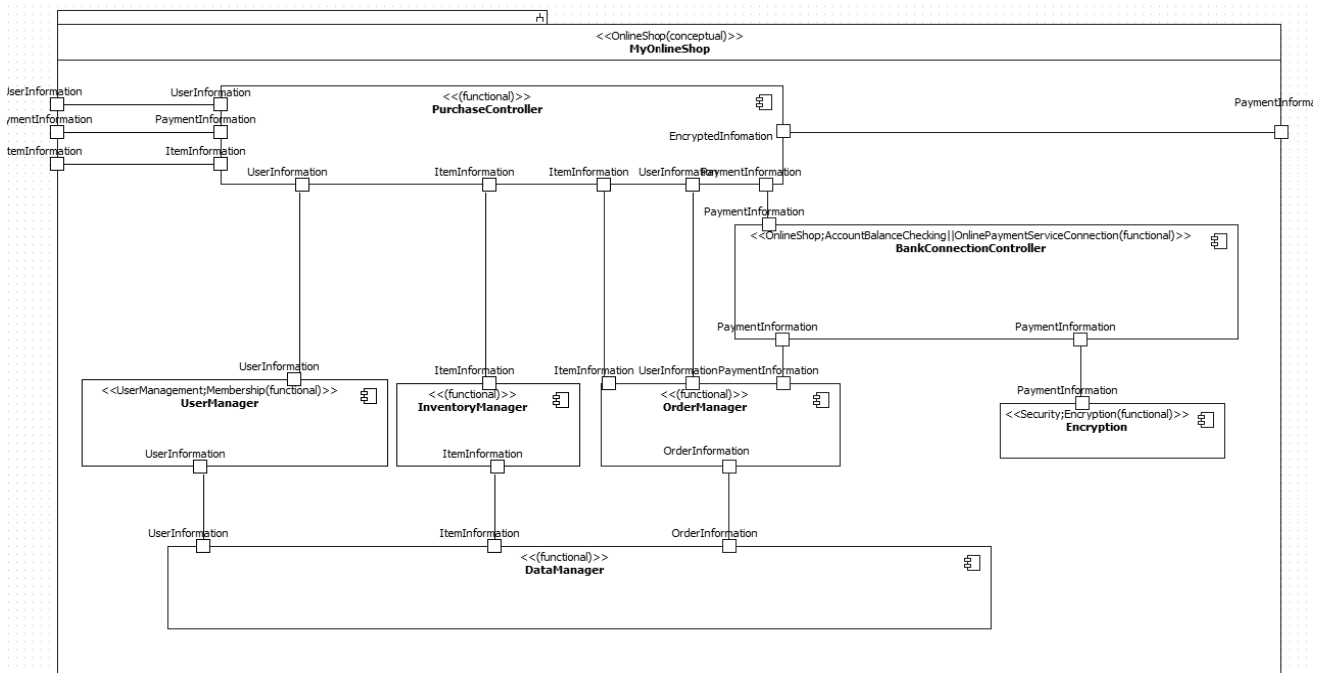


Fig. 6. Product Line Architecture for shopping mall

휘처에 대한 가변요소가 포함되어 있다. 두 개의 제어 컴포넌트들 제외한 다른 도메인 컴포넌트들은 개발자에 의해서 직접 개발이 이뤄졌다.

일관성 검사가 완료된 자산으로 부터 우리는 휘처 선택을

통해서 두 개의 제품을 생성할 수 있다. 아래의 Fig. 8과 Fig. 9는 휘처 선택을 통해 자동으로 생성된 각 제품의 행위명세 인스턴스를 보여준다.

이후 인스턴스와 액션스크립트를 이용해서 자바소스코드

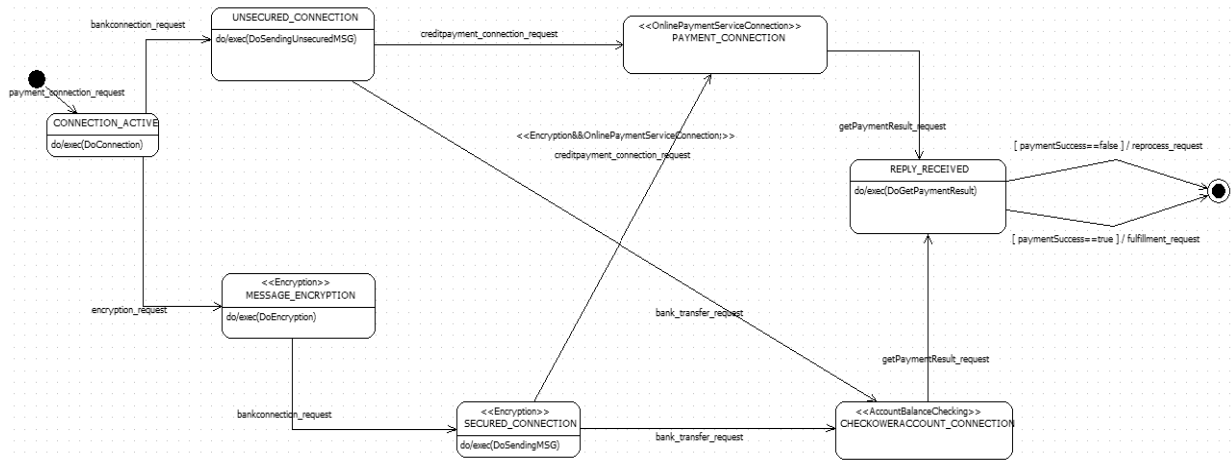


Fig. 7. Product Line BankConnectionController Component's Behavior Specification

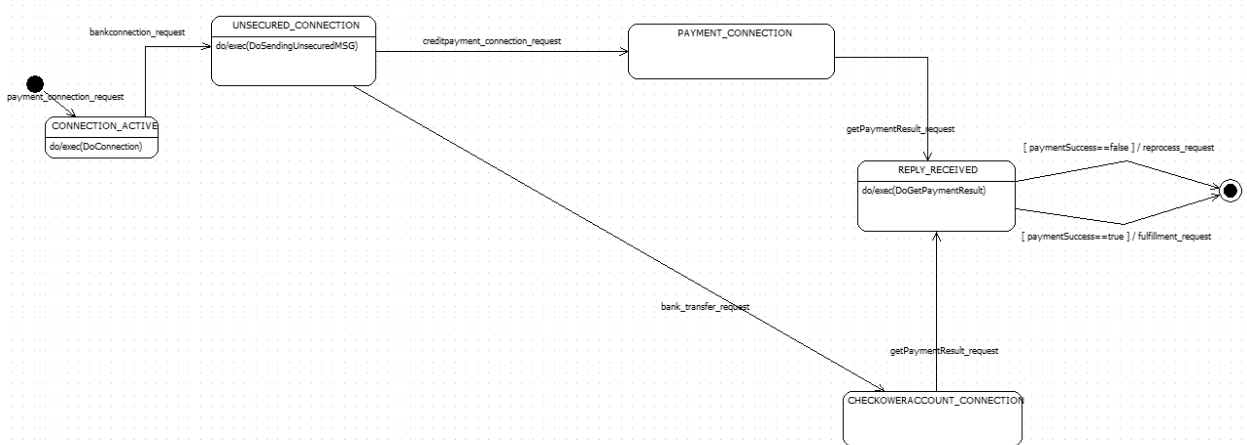


Fig. 8. A Instance example A of BankConnectionController Component's Behavior Specification (Encryption feature is not selected)

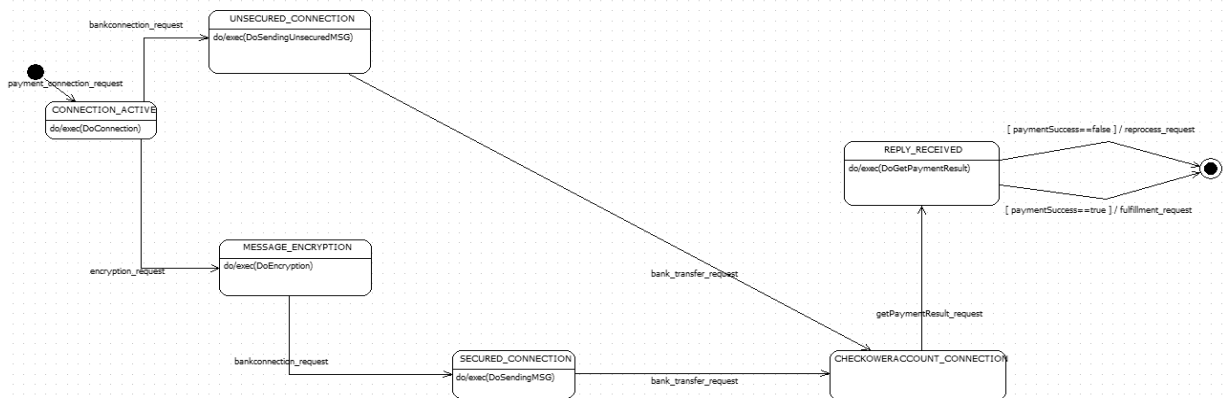


Fig. 9. A Instance example B of BankConnectionController Component's Behavior Specification (Online Payment Service Connection feature is not selected)



를 생성하고 간단한 시뮬레이션 패널을 이용해 생성된 제품 소프트웨어가 작성된 행위명세와 같이 동작하는지 생성된 소스코드에 포함된 디버깅용 기록을 이용하여 시험하고 생성된 두 제품 모두 행위명세에 따라 제어가 되는 것을 확인할 수 있었다.

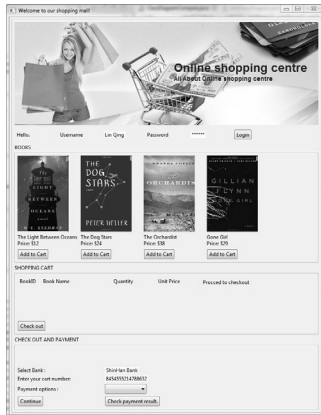


Fig. 10. A HMI panel for simulation

#### 4. 결론 및 향후연구

본 논문에서는 워크플로우 기반의 제품라인 소프트웨어 개발을 지원할 수 있는 도구에 대해서 소개했다. 본 도구를 이용해 기존 워크플로우 모델링 도구에서 부족한 제품라인 개념을 적용하여 소프트웨어를 개발할 수 있는 환경을 제공할 수 있었다.

하지만 아직 워크플로우 모델링을 위한 최소한의 요소만 제공하기 때문에 추가요소를 반영할 필요가 있으며, 추가요소 반영에 따른 검증기에 개선 및 검증기의 정확성을 위해 잘 알려진 오류예제에 대한 시험이 진행되어야 할 것이다. 이런 부족한 부분에 대해서는 점차적으로 보완해 나가면서 좀 더 실질적인 소프트웨어 개발에 적용하여 기능을 개선 보완할 예정이다.

#### 참 고 문 헌

[1] P.Clemens, L.Northrop, Software product lines: practices and patterns, Addison-Wesly Professional, Aug., 2001.  
 [2] K.Kang, S.Kim, J.Lee, K.Kim, E.Shin, and M.Huh, "FORM: a feature-oriented reuse method with domain-specific reference architecture," Annals of Software Engineering, Vol.5, pp.143-168, May, 1998.  
 [3] Hyesun Lee, Jin-Seok Yang, and Kyo C.Kang, "VULCAN: Architecture-Model-Based Software Development Work bench," The Joint 10<sup>th</sup> Working IEEE/IFIP Conference on Software Architecture and 6<sup>th</sup> European Conference on Software Architecture, Aug., 20-24, 2012.  
 [4] Hyesun Lee, Jin-Seok Yang, and Kyo C. Kang, "VULCAN

: Architecture-Model-Based Workbench for Product Line Engineering," The 16<sup>th</sup> International Software Product Line Conference, Sep., 02-07, 2012.

[5] www.omg.org, "Business Process Model and Notation", Jan., 2011.  
 [6] Visual Paradigm, "Business Process Visual ARCHITECT (BP-VA)"  
 [7] staruml.sourceforge.net/ko, "StarUML 5.0-OpenSource UML/MDA Platform"  
 [8] Gun-Woo Kim, Jeong-Wha Lee, Jin Hyun Son, "Design Anomalies in the Business Process Modeling", Journal of KIISE, Vol.14, No.9, pp.850-863, 2008. 12.  
 [9] H.Bi, J.Zhao, "Applying propositional logic to workflow verification," Information Technology and Management, 5(3-4), pp.293-318, 2004.  
 [10] Eshuis, H., Semantics and Verification of UML Activity Diagrams for Workflow Modeling, PhD thesis, Universty of Twente. CTIT Ph.D.-thesis series No.02-04.  
 [12] W.M.P. van der Aalst, "Workflow verification: Finding control-flow errors using Petri-net-based techniques", Business Process Management, pp.161-183, 2000.  
 [13] Kwan Woo Lee, "Managing and Modeling Variability of UML Based FORM Architectures Through Feature-Architecture Mapping," KIPS Transactions, Vol.19-D, No.1, pp.81-94, 2012. 2.



#### 양진석

e-mail : edeward@gmail.com  
 1999년 동국대학교 전산학과(학사)  
 2006년 고려대학교 컴퓨터학과(석사)  
 2006년~현 재 포항공대 정보통신 연구소  
 소프트웨어공학 연구실 연구원  
 관심분야: 소프트웨어공학, 정형기법, 동기화 언어, 제품라인공학, 제어 소프트웨어



#### 강교철

e-mail : kck@postech.ac.kr  
 1973년 고려대학교 통계학과(학사)  
 1982년~1984년 미시간대학교 Visiting Professor  
 1984년~1985년 (미국) 벨 통신 연구소 Technical Staff

1985년~1987년 AT&T 벨 연구소 Technical Staff  
 1987년~1992년 카네기멜론 대학교 Project Leader  
 1992년~현 재 포항공과대학교 교수  
 2000년~2001년 카네기멜론대학교 Visiting Scientist  
 관심분야: 소프트웨어 공학, 실시간 내장형 시스템, 소프트웨어 재사용, 제품라인 공학