

# 워크플로우 기반의 제품라인 소프트웨어 개발 지원 환경

양진석\*, Lin Qing\*, 강교철\*\*  
 \*포항공과대학교 정보통신연구소  
 \*\*포항공과대학교 컴퓨터공학과  
 e-mail : gildong@somewhere.sck.ac.kr

## A Tool for Workflow-based Product Line Software Development

Jin-Seok Yang\*, Lin Qing\*, Kyo C. Kang\*\*  
 \*POSTECH Information Research Lab., POSTECH  
 \*\*Dept. of Computer Engineering, POSTECH

### 요 약

제품라인공학 기반의 융합소프트웨어 개발 방법론에서는 어플리케이션 개발을 위해 아키텍처 모델을 제안하고 있다. 개발 모델에서는 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.

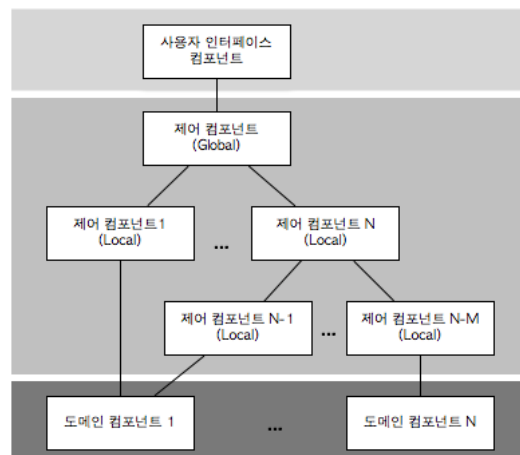
### 1. 서론

최근 들어 융합소프트웨어가 강조되고 점차 많은 기업들이 세계시장으로 진출을 모색하고 있다. 그 결과 소프트웨어 개발에 반영해야 하는 위치(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 (그림 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 워크플로우는 트랜잭션을 주로 처리하는 어플리케이션의 제어행위를 명세하기 위해서 제안되었다.



(그림 1) 융합소프트웨어 개발을 위한 아키텍처 모델

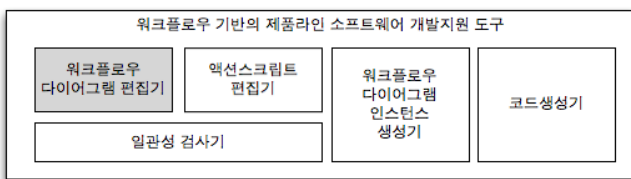
워크플로우 명세를 기반으로 제품라인 소프트웨어의 제어 컴포넌트 개발이 효과를 진행되기 위해서는 도구의 지원이 반드시 필요하지만, 기존의 워크플로우 모델링 도구들은 제품라인 공학 개념을 지원하지 않았다[5][6]. 그래서 우리는 융합소프트웨어 제품라인 방법론에서 제안하는 워크플로우 기반의 소프트웨어 개발을 지원하기 위해서 개발된 도구들을 제안한다. 개발된 도구들은 다른 도구들과 함께 워크벤치 별션을 구성하는 요소가 되었다.

본 논문은 다음과 같이 구성된다. 2 장에서는 워크플로우 기반의 제품라인 소프트웨어 개발도구에 대한 구성에 대해서 소개한다. 3 장에서는 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

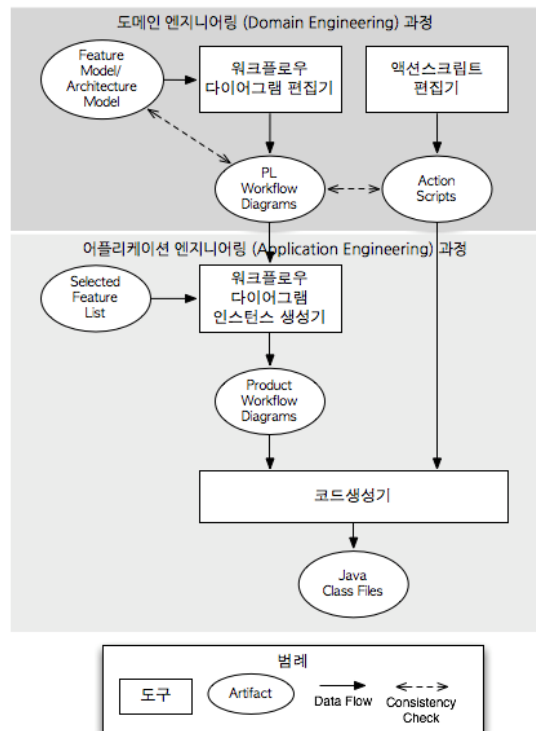


(그림 2) 워크플로우기반 제품라인 소프트웨어 개발도구의 구성

그리고 위의 기능을 제공하기 위해서 개발된 개발 지원도구는 그림 2 와 같이 다섯개로 구분될 수 있다. 개발지원도구는 ○ 워크플로우 다이어그램 편집기○ 를 제외한 나머지 도구들은 모두 이클립스 IDE 플랫폼을 기반의 이클립스 플러그-인 어플리케이션으로 개발되었다. ○ 워크플로우 다이어그램 편집기○ 의 경우 UML 2.0 표기법을 지원하는 다이어그램의 편집 기능만 제공하면 되므로 오픈소스 도구인 StarUML 5.0 을 이용하였다[7].

개발자들은 워크플로우 기반의 제품라인 소프트웨어의 제어컴포넌트 개발을 위해 그림 3 과 같은 흐름으로 지원도구들을 사용할 수 있다. 개발자는 자산을

만드는 제품라인 공학의 도메인 엔지니어링 (Domain Engineering) 과정에서 ○ 워크플로우 다이어그램 편집기○ 와 ○ 액션스크립트 편집기○ 를 이용하여 가변 요소가 포함된 제품라인 워크플로우 다이어그램(PL Workflow Diagram)과 액션 스크립트를 작성할 수 있다. ○ 일관성 검사기○ 를 통해 작성된 행위모델과 휘저모델, 아키텍처모델 사이의 일관성 검사 및 행위모델의 논리적인 오류를 검사할 수 있다. 검사가 완료된 행위명세는 자산으로써 관리된다.



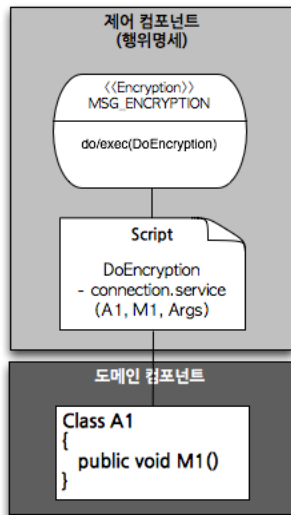
(그림 3) 제품라인기반의 소프트웨어 개발과정에서의 도구의 사용 및 도구들 사이의 관계

자산으로 부터 제품을 만드는 제품라인 공학의 어플리케이션 엔지니어링 과정에서 개발자는 ○ 워크플로우 다이어그램 인스턴스 생성기○ 를 이용해 휘저모델로 부터 선택된 휘저목록과 제품라인 워크플로우 다이어그램을 입력으로 제품 워크플로우 다이어그램을 생성할 수 있다. 생성된 제품 워크플로우 다이어그램은 휘저선택에 따른 가변요소에 대한 파라미터화가 적용된 제품라인 워크플로우 다이어그램의 인스턴스(Instance)이다. 인스턴스의 예는 3 장에서 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 UML2.0 의 요소들 가운데 State Diagram 과 State 의 액션(Action)을 활용하고, 도메인 컴포넌트와의 결합을 위한 액션 스크립트(Action Script)를 추가로 제공한다. 이들 사이의 관계는 그림 4 와 같다. 개발자는 State 의 do 액션속성에 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.



(그림 4) 워크플로우 행위명세와 아키텍처모델과의 관계

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 ⓪ 코드 생성기⓪ 를 통해서 특정 구문으로 치환이 된다. 그 예는 다음과 같다.

```

@@VAR
String orderID;
Boolean result;
@@BODY
Vector<String> args = new Vector<String>();
orderID=(String)@GETVALUE(⓪ orderID⓪ );
result = (Boolean)connector.service(⓪ UserManager⓪ ,⓪ Validate⓪ , args);
    
```

위의 액션 스크립트 예에서 UserManager 클래스의 Validate 메소드를 호출하는 것을 확인할 수 있다. 워크플로우 행위모델의 인스턴스를 파라미터화 방법을 통해서 생성하기 위해서 다이어그램을 구성하는 State 에 휘처를 연결(Binding)할 수 있는 방법으로 스테레오 타입(Stereo Type)을 사용한다. 그림 4 에서 MSG\_ENCRYPTION 는 스테레오타입에 Encryption 휘처가 연결되어 있다. 어플리케이션 엔지니어링 과정에서 선택된 휘처집합에 Encryption 휘처가 없다면 해당 휘처가 연결된 모든 State 와

State 에 연결된 모든 Sequence Flow 는 인스턴스를 생성하는 과정에서 사라지고 그에 따라서 야기 될 수 있는 도달할 수 없는 State (Dead State)까지 사라지게 된다.

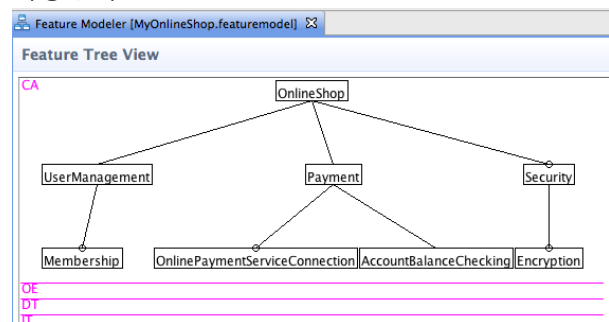
휘처모델-아키텍처-워크플로우 행위 명세 사이에 일관성 검사는 현재 아래의 네 가지 기본적인 규칙을 통해서 이뤄진다. 첫째, 모든 휘처는 적어도 한번은 아키텍처 구성 요소 및 워크플로우 행위명세에 연결되어야 한다. 둘째, 아키텍처를 구성하는 모든 구성요소는 휘처에 바인딩 되어 있어야 한다. 셋째, 아키텍처와 워크플로우 행위명세에 연결된 휘처들로 부터 도출되는 휘처들 사이의 계층적 관계는 휘처모델의 휘처들 사이의 계층적 관계와 동일해야 한다. 넷째, 대체적 관계의 휘처는 같은 요소에 연결될 수 없다. ⓪ 일관성 검사기⓪ 는 위의 규칙들에 대한 검사를 수행한다. 하지만 아직 워크플로우 행위모델에서 발생할 수 있는 논리적인 오류에 대한 검사는 지원하지 않으며 개발중에 있다.

마지막으로, ⓪ 코드생성기⓪ 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

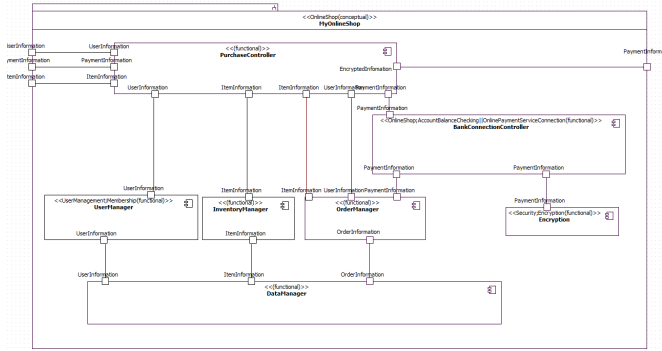
우리는 간단한 물품구매 소프트웨어를 활용 예제로 사용했다.



(그림 5) 물품구매소프트웨어의 휘처모델

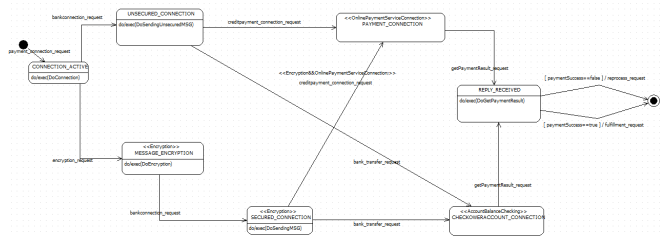
개발 대상이 된 소프트웨어의 가변요소는 그림 5 의 휘처모델에서 알 수 있듯이 Membership, Online Payment Service Connection, 그리고 Security 및 Encryption 휘처로 정의하였다.

융합소프트웨어 제품라인 방법론에 따라서 아키텍처 모델링을 수행하였다. (그림 6 참조. 아키텍처 모델링 방법은 본 논문의 범위에서 벗어나므로 자세히 설명하지 않는다).



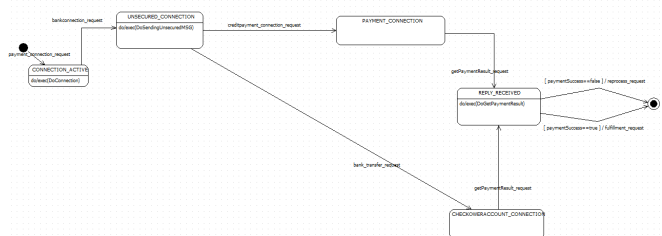
(그림 6) 제품구매 소프트웨어의 제품라인 아키텍처

도출된 컴포넌트들 가운데 두 개의 제어 컴포넌트 PurchaseController 와 BankConnectionController 에 대해 본 논문에서 제시한 워크플로우 모델을 이용하여 각 컴포넌트의 행위명세를 진행하였다. 아래의 그림 7 은 BankConnectionController 의 행위명세를 보여주며, 내부적으로 Encryption 과 Online Payment Service Connection 선택적 위치에 대한 가변요소가 포함되어 있다. 다른 컴포넌트들은 메뉴얼하게 개발이 이뤄졌다.



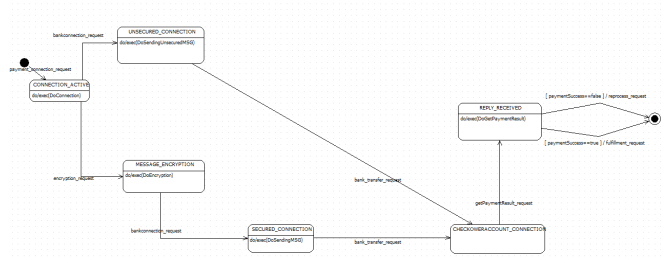
(그림 7) 제품라인 BankConnectionController 컴포넌트의 행위명세

일관성 검사가 완료된 자산으로 부터 우리는 위치 선택을 통해서 두 개의 제품을 생성할 수 있다. 아래의 그림 8 과 그림 9 는 위치 선택을 통해 자동으로 생성된 각 제품의 행위명세 인스턴스를 보여준다.



(그림 8) BankConnectionController 컴포넌트 행위명세의 인스턴스 생성 예제1 (Encryption 위치를 미선택)

이후 인스턴스와 액션스크립트를 이용해서 자바스코드를 생성하고 간단한 시뮬레이션 패널을 이용해 생성된 제품 소프트웨어가 행위명세와 같이 동작하는 시험해 볼 수 있었다.



(그림 9) BankConnectionController 컴포넌트 행위명세의 인스턴스 생성 예제 2 (Online Payment Service Connection 위치를 미선택)

#### 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, 앞서서도 언급하였듯이 행위 모델 자체를 검증할 수 있는 환경 및 검사 환경이 부족하여 보완해야 할 필요가 있다. 이런 부족한 부분에 대해서는 점차적으로 보완해 나가면서 좀 더 실질적인 소프트웨어 개발에 적용하여 기능을 개선할 예정이다.

#### Acknowledge

이 논문은 정보통신산업진흥원의 SW 공학 요소기술 연구개발사업에 의해 지원되었음을 밝힙니다.

#### 참고문헌

- [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, Ö FORM: a feature-oriented reuse method with domain-specific reference architecture,ö Annals of Software Engineering, Vol5,pp.143-168,May 1998
- [3] Hyesun Lee, Jin-Seok Yang, and Kyo C.Kang, Ö VULCAN: Architecture-Model-Based Software Development Workbench,ö 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, Ö VULCAN : Architecture-Model-Based Workbench for Product Line Engineering,ö The 16<sup>th</sup> International Software Product Line Conference, Sep 02-07, 2012
- [5] www.omg.org, Ö Business Process Model and Notationö , Jan. 2011
- [6] Visual Paradigm, Ö Business Process Visual ARCHITECT (BP-VA)ö
- [7] staruml.sourceforge.net/ko, Ö StarUML 5.0-OpenSource UML/MDA Platformö